GPS usage aboard the BlueSat microsatellite mission

Sabella, D.A.
Satellite Navigation and Positioning Group
The University of New South Wales
Sydney, Australia

Abstract

This paper describes a navigation system designed for the University of New South Wales BlueSat microsatellite. Through the use of a modified, off-the-shelf CMC AllStar GPS receiver, position, velocity and time data can be robustly provided in a highly dynamic environment.

Hardware and software interfaces have been carefully designed to allow integration within the BlueSat project. Techniques have been developed for processing the data and providing the system with navigation criteria. Risk mitigation is also examined.

Limited testing has been performed, with the high velocity receiver able to robustly track GPS Space Vehicles. Further testing will be carried out when a multi-channel GPS constellation simulator becomes available.

Nomenclature

CAN	Controller Area Network
LEO	Low Earth Orbit
NMEA	National Marine Electronics Association
PVT	Position, Velocity and Time
RAAN	Right Ascension
SV	Space Vehicle
PCB	Printed Circuit Board
TTFF	Time to first fix

1. Introduction

The navigation system described in this paper will be launched as a research payload on the BlueSat microsatellite.

BlueSat is a research-focused low earth orbit (LEO) microsatellite project at the University of New South Wales and is designed and built entirely by students. The BlueSat charter focuses on providing a platform for innovative experimental payloads and so is an excellent test-bed for new technologies. This paper describes the specification and design of a supplementary GPS-based navigation payload.

The current BlueSat mission is already equipped with a navigation computer so any system produced will initially only be used for testing and research, rather than mission-critical navigation. The GPS subsystem will initially assist the BlueSat flight computer to determine navigation solutions while in orbit. However, if the implementation proves to be reliable, the system may be used as the principal navigation device in later missions.

To allow a standard GPS receiver to function in space, many software and hardware modifications are necessary. The main software issue is algorithm robustness. Any algorithm implemented must be able to withstand the highly dynamic conditions - high velocity, altitude and acceleration. The environment also requires robust hardware to cope with the conditions experienced.

There are many risks inherent in such a space system built with off-the-shelf components. This paper attempts to address the major risks and describes the mitigation techniques chosen, where possible.

Testing modifications is unfortunately difficult, as it is not possible to realistically simulate space conditions on Earth. Methodologies are therefore developed using specialised testing equipment. The tests specified aim to fully evaluate performance prior to satellite launch, maximising the chances of success.

2. BlueSat

BlueSat (The Basic Low-earth-orbit University of New South Wales Experiment Satellite) is the microsatellite platform on which the receiver will be installed. The aim of BlueSat is to enable students to obtain hands-on experience in building a viable, research-focused microsatellite with amateur radio facilities.

The satellite has been designed entirely by students, who have been assisted by access to the design plans for the existing AMSAT microsatellite project. Relevant data relating to BlueSat is reproduced here; further details of BlueSat's goals and design are available at [1], while technical details are contained in [2].

BlueSat will include the navigation system described in this paper as an experimental payload. If tests are successful, future missions may use GPS technology as the primary navigation device. However, very little of the material described here is BlueSat-specific and the modified receiver could be used for other satellite missions.

The Keplerian orbital elements for the BlueSat mission have yet to be finalised, and are dependent on the ultimate launch provider. Expected values are [2]:

Table 1. BlueSat Keplerian Orbital elements

Element	Description	Value	Unit	Comment
A	Semi-major axis	6878-7378	km	Earth's (6378km) radius plus orbital altitude
E	Eccentricity	0	-	Circular orbit
I	Orbital inclination	80-100	degrees	Near polar sun-synchronous
□	RAAN	no data	degrees	
□	Argument of perigee	no data	degrees	
□	period	~ 100	minutes	

3. Requirements

In consultation with the BlueSat team several performance goals were determined for the GPS navigation project:

Position and velocity

Since the GPS subsystem is principally associated with navigation, it must provide reliable and accurate position and velocity data.

Time

As BlueSat is a research satellite, it will be making and recording a large number of measurements. These measurements must be stamped with the exact time they occurred to assist future analysis. As the GPS technology provides precise timing information, it can be used to calibrate the system clock.

Cheap and reliable

BlueSat has a very small budget, so the solution must be cheap to design and implement. Despite the requirement to be economical, reliability is essential, as navigation systems are mission-critical.

Withstand space conditions

Typically conditions in LEO are very different to those at sea level. There are many different issues that arise including solar radiation, high speeds/acceleration, temperature extremes and lack of atmosphere. The solution developed must adequately cope with these conditions.

Can be updated from the ground

As with all projects, requirements change over time. Therefore, there must be a mechanism for updating or modifying the GPS receiver's behaviour from the ground. If additional debugging data or a new feature is required, it must be possible to recompile a new binary on the ground and then upload it from the groundstation.

Fully integrated with BlueSat

It is important that the device interfaces correctly with BlueSat. That is, BlueSat must be able to control the device, understand its output and provide power to it.

4. Baseline Equipment

The firmware produced was based on the Mitel GPS Architect source code. This is a time-proven and complete C/Assembler implementation of the GPS algorithms for terrestrial (low velocity) usage. Access to the source code was required to allow the necessary algorithm modifications for LEO usage. The development of the firmware and algorithms is described in later sections.

A modified CMC AllStar device was used as the baseline receiver hardware. The AllStar is a twelve-channel GPS chipset, based around the GP2015 RF front end and the GP2021 correlator. A 40MHz ARM60 microprocessor provides overall control of the system. Three RS-232 ports are available on the device – two for general input/output and one for debugging.

The physical modifications made to the AllStar facilitated the use of the GPS Architect source code. As part of the modifications, more SRAM was also provided to make extra run-time memory available.

After these modifications were made, the device was named “ArcStar” (fusion of the words “Allstar” and “Architect”), to differentiate it from the unmodified AllStar receiver.

5. Environmental Risks

While it is easy to fix a bug in code by uploading a new code image from the groundstation, mechanical and electric failures cannot be so easily guarded against. The hostile environment in LEO is expected to be the most likely point of failure. However, once the microsatellite is launched it will be impossible to make physical changes to any of the subsystems. For this reason it is very important that interfacing and hardware be designed for full reliability.

5.1 Pitch and Roll

It is expected that BlueSat will pitch twice per orbit and roll with a period of approximately 2.5 minutes. This means that each GPS Space Vehicle (SV) will stay in view for a maximum of approximately one minute. With the tools available, it was not possible to fully simulate this rotational motion. For the purposes of testing, it was assumed that ephemeris data can be acquired within the one minute interval of uninterrupted spacecraft view. From the results of the BirdSat team [3], this does not appear to be an unreasonable assumption. If this proves not to be the case, it may present a challenge to achieving navigation solutions. However, it is not a critical point of failure. If the code proves unable to lock onto the SVs while in view, further optimisations may be made to the algorithms, and a new code image may be uploaded from the groundstation.

There is no theoretical reason why sufficiently optimised code cannot perform a signal lockon within this timeframe - ephemeris data should be downloadable with a lower bound of around twenty seconds.

Unlike the GPS subsystem, the radio uplink is expected to be largely unaffected by roll due to its antenna geometry. Therefore the upload of new code will be possible despite any roll experienced.

5.2 Software Failure

It is very uncommon for computer programs to be without errors of some kind. As a result it is critical that software failures do not in any way endanger the operability of the system. Section 7.3 describes the procedure for updating code from the ground. Through the use of this update mechanism, it is expected that software bugs will not result in unrecoverable failures.

5.3 Temperature and Vacuum

The BlueSat team expect an operating temperature range of -2 to +27C. This falls well within the ArcStar’s environmental specifications of -30 to +75C.

However, these specifications do not guarantee operational success. The manufacturer data is provided for operation in the Earth's atmosphere. However, in the near-vacuum conditions of LEO there is no air. As a result, convection cannot take place and thus cooling must rely only on radiation effects.

Vacuum testing has not yet been undertaken, so it is not possible to be sure that the heat dissipation will be adequate. Heat generation appeared to be fairly minimal however, and no extra cooling technology is required for on-Earth conditions.

5.4 Soldering/Component Requirements

The demands placed on components and solder joints are far greater in a space environment than on Earth. One failure point experienced in LEO is the jerk stress - joints can work themselves free, particularly when solder quality is poor. Secondly, in LEO there are much greater changes in temperature, leading to far greater thermal stresses and contractive forces. When directly exposed to the Sun's radiated heat energy, the microsatellite will be relatively hot. However, when the Earth obscures BlueSat's view of the sun, the temperature will drop sharply.

This temperature cycling will cause thermal expansion effects on the printed circuit board (PCB) and the various components, causing them to shrink and grow. The extent and direction of the expansion is dependent on the thermal coefficient of the material. For example, the PCB may shrink more than the surface-mount components that are attached to it, leading to forces at the solder points where the component is attached to the PCB.

Some soldering risk points were located and all can be easily mitigated through reflowing or resoldering of the suspect joints. The placement of components and the design of the PCB was also inspected and is considered to be quite acceptable for LEO usage.

5.5 Acceleration/Jerk Stress

The orbital and launch forces due to accelerations and jerk stresses will place the most strain upon the physical connections to the device. Therefore it is important that the soldering guidelines above are adhered to so that components are not sheared off the board.

It is also crucial that all connectors are designed so that they sit securely and do not fall out due to jerks. An inspection has been undertaken and the connectors were found to be adequate. The BlueSat team will also undertake a full pre-launch compliance test. This will include jerk and acceleration testing. This will ensure that there are no obvious failure points.

5.6 Solar Radiation

Solar radiation is arguably the most crucial environmental issue threatening BlueSat's continued operation. The main concern in this area is cosmic rays and other high-frequency energy causing damage to silicon - both memory and processors. This damage can be either temporary (e.g. bit-flipping, where a bit may suddenly change value after the energy passes through the silicon) or it can be permanent, and render the memory or device useless in the future.

In the case of occasional bit-flipping, the problem is not considered severe. Restarting and reloading the code will restore the receiver back to full performance, provided these events do not occur too regularly. However, permanent damage is much more of a problem.

Options to combat permanent or temporary damage include:

- Sourcing "rad-hard" components, that are fabricated to resist radiation damage.
- Full redundancy and shutdown of affected subsystems (e.g. provide backup GPS receivers in case of radiation damage to one device).
- Shielding to enclose the device in a shell of material such as tantalum, to protect from radiation.

The BlueSat mission expects to concentrate on shielding techniques for radiation damage mitigation.

6. *Orbit Analysis*

To model the orbital conditions experienced by BlueSat on launch, the Matlab Constellation toolkit [4} was used to perform orbital simulations. The simulations were run for the period corresponding to GPS week 141.

Simulations were performed to study BlueSat visibility from a groundstation located at UNSW. The results and some analyses are presented in this section. Note that none of these simulations included satellite roll effects due to equipment limitations (Section 5.1.)

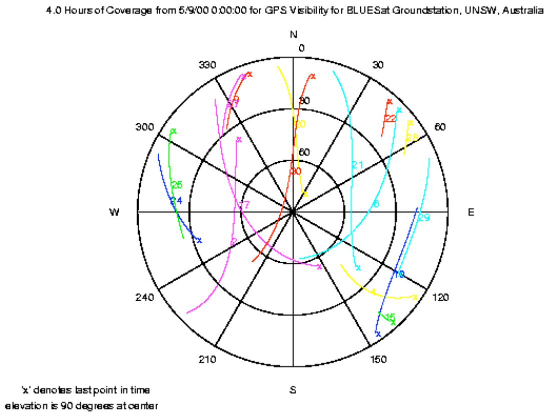

Figure 1. GPS SV visibility from groundstation, UNSW

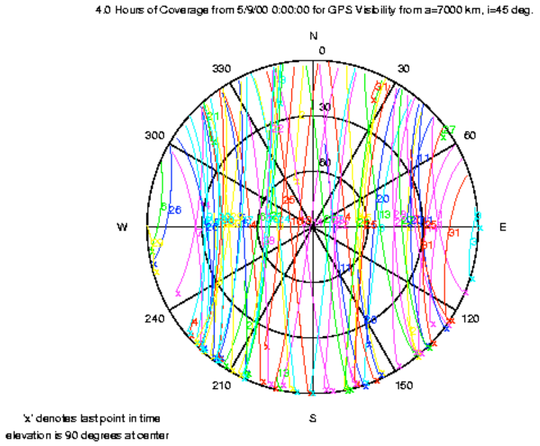

Figure 2. GPS SV visibility from BlueSat

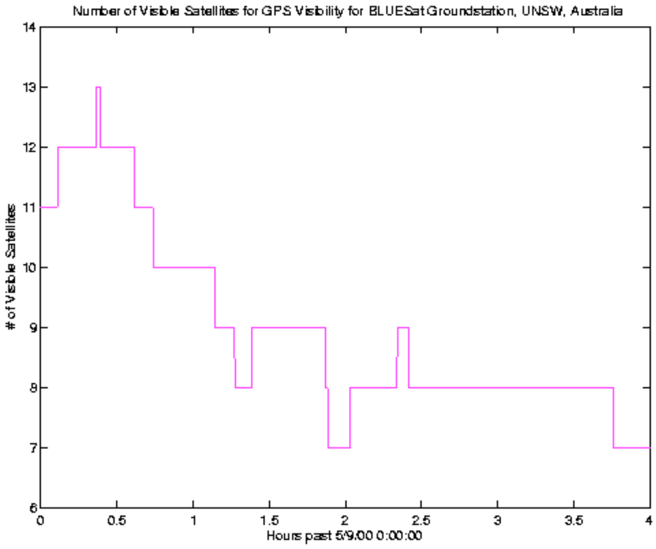

Figure 3. GPS SV satellite availability from the BlueSat groundstation

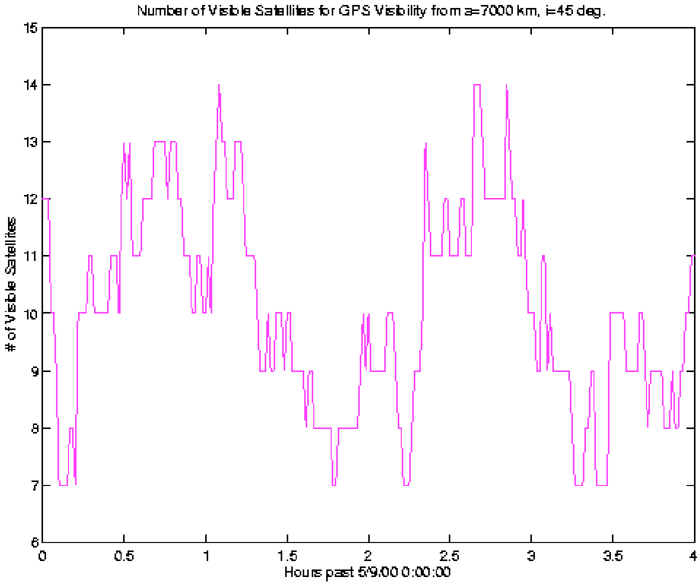

Figure 4. GPS SV satellite availability from the BlueSat microsatellite

Figure 1 shows the sky view from the ground, while Figure 2 shows the SVs that can be seen by BlueSat in LEO over the same time interval. There are clearly many more GPS SVs that are visible in LEO than at the groundstation. This is predominantly due to the high altitude and velocity of the microsatellite.

It can also be seen from Figure 3 and Figure 4 that while the sky view appears far more dynamic than on the ground, the SVs will stay in view long enough for a fix to be maintained. In fact, at a stated ~2.5mins per revolution [2], the roll of BlueSat will clearly be the limiting factor in visibility windows, rather than the visibility time of SVs.

The results of the simulations therefore indicate that navigation solutions should be achievable so long as robust tracking and navigation algorithms are used.

7. Interfacing

A critical part of the project is to consider the ways in which the ArcStar GPS receiver will interface with BlueSat and other systems. The GPS payload will directly interface with BlueSat's flight computer. The data connection will be via a standard RS-232 (serial) connection. Some research was undertaken into the CAN bus, but was later abandoned due to power considerations. However, CAN may be used on future missions as an interface bus.

7.1 Physical Interfacing

Table 2 shows the physical connectors used to interface the ArcStar with other devices.

Table 2. Interface specifications

Interface	Connector	Comments
Power	0.100 x 0.100, 20 pin (2x10) Straight header	5V DC. See [5] for details
Serial	0.100 x 0.100, 20 pin (2x10) Straight header, as above	3 ports provided, as described below
Antenna	Female TNC	

The GPS receiver has three RS-232 serial ports. They are assigned as follows:

- Debug - This port allows a separate host to communicate with the monitor software on the ArcStar. Typically this port will be used for uploading executable code and starting, stopping or debugging code execution. The protocol used on this port is fixed by the monitor software and cannot be dynamically varied by navigation code.
- General Purpose Port A - This is a generic serial port attached to the GP2021 correlator ASIC. By default, this is used to receive DGPS correction messages. Since DGPS is not being used in this project, the port remains unused.
- General Purpose Port B - Currently used for providing NMEA output, as defined below. This port is also controlled by the GP2021.

7.2 Software Interfacing

The position data is provided in industry-standard National Marine Electronics Association (NMEA) format. Supported NMEA message strings are GGA, RMC, GSV, GSA and PRD. A compact, proprietary BlueSat format has also been defined. The proprietary format contains the data:

- SVs tracked
- Time
- Date
- Latitude
- Longitude
- Altitude
- Velocity

This proprietary protocol encodes the information in 16 bytes per sample and contains the bare minimum information necessary for navigation. As a result, more output data may be stored in limited memory space.

7.3 Remote Updates

It was realised from the outset that the code update mechanism must be implemented very robustly. Uploading new images from the ground mid-mission must be possible.

The method for installing new firmware is as follows:

7.3.1 New code from the ground

In the scenario where a new code image must be uploaded from the ground to BlueSat, the update needs to be performed as follows:

1. On the ground, changes should be made to the code as required. The code should then be compiled.
2. During the next available viewing window, the new code is uploaded from the groundstation to BlueSat using the radio modem link.
3. BlueSat stores the image at a known address in the main system memory, in the segment allocated to the GPS subsystem.
4. The groundstation sends the flight computer a command to load the code from the memory into the ArcStar.

5. The flight computer initiates a session with the ArcStar and uploads the code to the GPS from the known address and commences code execution.

7.3.2 New code stored in BlueSat's main memory

If the code to be loaded is already stored in BlueSat's system memory, the procedure above is the same, but obviously starts at step 4. In this case there is no need to transfer a new image from the ground.

7.3.3 Multiple images in BlueSat's main memory

Since an executable image is around 120kB in size, there is room to store multiple images in BlueSat's 16 megabytes of allocated storage. Therefore the flight computer can implement a file system to allow the storing of multiple executable images. With this system, the flight computer is able to pick a particular image by name from its memory. and load the image into the ArcStar as required.

This method does not require a new image to be uploaded from the ground every time the code image is swapped, providing the code is already stored on BlueSat. That is, at Step 4 above, the flight computer could be sent a filename from the ground. In Step 5, the flight computer would look up that filename, and send the appropriate executable to the ArcStar. This would be particularly useful when attempting to compare the performance of two different images.

8. High Precision Timekeeping

A design goal provided by the BlueSat project was a high precision time source. Since BlueSat is principally a research microsatellite, it will regularly need to take measurements and store or parse data. When analysing error logs or other data generated by BlueSat, it is very convenient to know exactly when a particular event took place. Since GPS can provide high precision synchronised timing, it provides an excellent source of high reliability time. The ArcStar provides time information accurate to ±100nanoseconds.

Using the GPS receiver as the definitive time source on the microsatellite requires that the ArcStar always be on. This is not practical on BlueSat due to power budgets, so an alternative solution is needed.

When the GPS receiver is off the flight computer can maintain time information itself. Each time the GPS receiver is turned on the flight computer would retrieve a timestamp from the ArcStar. The flight computer is then able to update its system time, only occasionally referring to the GPS in order to recalibrate its clock.

9. Algorithms

The changes made to the ArcStar software principally addressed two issues. Firstly, algorithms were written to support LEO operation. Secondly, code was written to facilitate efficient communications with BlueSat. The specific issues that have been addressed are:

- Removing the artificial height and velocity limitations placed on navigation solutions
- Correcting Doppler prediction algorithms
- Removing constraints imposed by navigation solution filtering
- Ensuring coordinate system implementation is sufficiently robust
- Providing initial velocity as well as position/time data to allow faster acquisition
- Providing initial condition debugging information

10. Size Optimisation and Memory

If at any point after launch new firmware is required, the code must be uploaded to the microsatellite via the main up-link. This radio modem link operates at 9600bps through the atmosphere and is only available when BlueSat passes overhead of the groundstation. As a result, the window of opportunity for radio communications will not exceed approximately 9 minutes. Also, sufficient storage space in RAM is needed for storing results.

For these two reasons it is important to keep the software images as trim as possible. There are several areas where it has been possible to cut significant bulk off the code size:

- Support for the Mitel proprietary data output. Since either NMEA or a far more simple BlueSat proprietary protocol are being used, the Mitel protocol may be removed.
- DGPS support is unnecessary as this mission will not use DGPS.
- Various other smaller modifications.

The GPS subsystem has 16 megabytes of BlueSat main memory exclusively available to it. As a result, multiple execution images may be stored in the main memory, and the appropriate image may be loaded directly. That is, a new image may be loaded without necessarily transferring the data across the ground link each time the switch is made. This process is described in Section 7.3.

The navigation code image is approximately 120 kilobytes at present. As a result it is anticipated that the size of BlueSat's main memory will present no difficulties for image storage.

11. Testing

In general it is trivial to test the operation of a GPS tracking algorithm for both performance and reliability. A standard test would be to acquire a Position, Velocity and Time (PVT) fix at a known location, and compare the known value with the GPS-derived value. Time to First Fix could also be measured by simply timing the acquisition process.

This procedure is suitable for testing ground-based receivers, however is not useful in the context of this work. The ArcStar already works well on the ground; it is in the dynamics of LEO that functionality must be tested. This is quite hard to do, velocities of kilometres per second and altitudes of hundreds of kilometres can't easily be simulated while on the Earth's surface. A mechanism is described below for performing this testing with the use of RF signal simulators.

It must be remembered, however, that it is not critical that the code functions perfectly on launch. Since a remote update mechanism is available, any error in the code can be fixed remotely. Only the interfacing is required to work with high reliability.

11.1 Testing Software and Algorithms

As the dynamic conditions of space cannot be easily simulated in the lab, another method must be used to evaluate results. The most practical solution is the use of a GPS Signal Generator. These are sophisticated electronic devices that simulate GPS SV signals. When provided with appropriate operational parameters, the signal generator outputs an RF signal that approximates that which should be received from the SVs in question.

This determinism is a very useful property. In contrast to the real GPS constellation, the operational parameters are exactly known and can be exactly compared with the values observed by the ArcStar. Furthermore, the tests can be repeated over and over again, using exactly the same scenario. This means identical tests may be performed multiple times to truly evaluate performance.

11.2 Hardware and Interfacing

Interface testing could not be fully completed, as the BlueSat portion of the interfaces had not been fully constructed at the time this project was undertaken.

It is critical that connectors do not wriggle out of place during orbit. When BlueSat is tested for jerk stresses on the vibration simulator, attention will be paid to ensure that the connectors are held firmly in place and do not slip.

Consideration has also been given to the environmental effects on connectors. For example, certain plastic connectors and insulation exposed to the increased ultraviolet radiation in LEO may become brittle and break. As a result it is critical that all connectors and cables are rated for LEO conditions.

Vacuum testing is also planned. The ArcStar will be tested in a near-vacuum environment on Earth and monitored to ensure heat dissipation is adequate. If overheating appears to be a problem, appropriate cooling strategies will have to be put into place.

11.3 Testing Methodologies

A detailed procedure for evaluating the performance of algorithms is provided by [6]. The authors used an STR4760 multi-channel simulator to test their algorithms. The outlined steps can be equally applied to the ArcStar.

When testing algorithm modifications, there are three main parameters that must be considered. These are detailed below.

- **Ephemeris download time.** Since the roll of the satellite limits the view of any given region of the sky to just over one minute, it is critical that ephemeris data is able to be downloaded in this time. If ephemeris download is interrupted, it must be restarted from the beginning.

- **Time to first fix (TTFF).** TTFF is important, because it is “dead time” - the ArcStar is consuming power, but not producing any useful results. Since microsatellites have a tight power budget, it is important that TTFF be minimised to conserve power. If the TTFF is too long, then it will not be feasible to turn on the GPS receiver too often as the high power and time overheads experienced in the acquisition phase will reduce the availability of power to other systems.

- **Ability to maintain fixes.** Once the first PVT fix is obtained, it is important that the ArcStar continues to successfully track the satellites. The testing process must ensure that continuous tracking can be maintained with the velocities and Doppler shifts encountered in LEO. Also, the tracking should be mildly resistant to jerks - small jerks should not affect the tracking performance.

12. Algorithm Results

A Spirent GSS4100 single channel SV simulator was available for a short time. This did at least allow for testing individual SV tracking, however it did not permit full evaluation of all the algorithms. A multi-channel simulator is required for full simulation of a PVT fix.

The GSS4100 was initialised and tracking tests were run. The experiments showed that with the modified firmware, the ArcStar was able to successfully lock onto SVs with Doppler shifts of at least 5km/s. The simulated SV could be tracked indefinitely, indicating that the algorithms are robust enough to maintain tracking at these high dynamics.

By comparison, the original firmware was limited to a maximum velocity of 83.3m/s. It was impossible to assess TTFF since it was not possible to achieve a PVT fix with only one simulated SV.

Although these results are promising, successfully tracking of one GPS satellite does not ensure operational success. Until the code is tested on a multi-channel simulator it will be impossible to fully assess functionality.

13. Conclusions and Future

A straightforward implementation has been specified for immediate deployment of a GPS receiver on BlueSat. The resulting design meets all BlueSat design goals and is built entirely from cheap, off-the-shelf components designed for terrestrial use. This allows leveraging of the significant engineering experience employed in producing GPS ground-based receivers.

All Matlab orbital simulations and the experiences of related projects show that the GPS SVs should be visible from BlueSat. Full navigation should be possible, providing the current version of the firmware can track with the specified microsatellite roll characteristics. Full testing on a multi-channel simulator will confirm whether this is indeed the case.

If any software bugs are apparent, or any required features are missing from the firmware, then it is possible to update new code from the ground. The effect of some hardware failure points has also been considered.

Environmental conditions and other risks were analysed. Mitigation strategies were suggested, where possible, for all risks. The basic testing undertaken has indicated promising results, with the receiver maintaining a good signal lock on a SV at all times, at Doppler shifts of several kilometres per second.

Further development and refinement is currently being undertaken on the project, including porting this modified firmware to the new Sigtec MG5001 receiver [7]. This device has a much smaller physical footprint and consumes less power than the ArcStar, making it better suited for space usage.

References

[1] BlueSat (2002) The BlueSat project website [Online], Available: http://www.bluesat.unsw.edu.au/ [Accessed: 2003, February 3]

[2] BlueSat (2001) BlueSat Annual Report 2001, The University of New South Wales, Sydney, Australia.

[3] Gill E., Montenbruck O., & Briess K. (2000) GPS-Based Autonomous Navigation for the BIRD Satellite, 15th International Symposium on Spaceflight Dynamics, Biarritz, Germany.

[4] Constellation Matlab Toolkit, Available http://www.constell.org/ [Accessed: 2003, February 28]

[5] Bae Systems (1998) AllStar Users Manual, Part Number: 220-600944-0XX.

[6] Montenbruck O., & Holt G. (2002) Spaceborne GPS Receiver Performance Testing, DLR, Cologne, Germany.

[7] Signav web site [Online], Available: http://www.signav.com.au/ [Accessed: 2003, February 28]