Real Time Hardware in The Loop Simulation Testbed of Spacecraft Formation Flying

Submitted by
Riaz Ahmed
Supervisor: Dr. Adrian Tatnall

PhD Thesis
Jun, 2008

Astronautics Group, School of Engineering
University of Southampton

Abstract

The great potential benefits associated with SFF (satellite formation flying) have led to considerable research in this concept around the world. As it is a new field of research and the implementation of formation flying in practice brings some inherent challenges and risks It is therefore particularly useful to develop a real time platform, where formation flying theories, technologies, algorithms and their coordination in open and close loop can be tested in scenarios close to those actually expected.

In this research a low cost, real time hardware in the loop (RTHIL) test bed for low earth orbit autonomous SFF consisting of nano satellites has been developed which can integrate original hardware in the close or open loop. The proposed real time close loop test bed consists of a real time SFF model, a real time relative navigation system, guidance and control algorithm, and GPS simulator.

Different SFF models already developed by researchers have been studied for their potential use in a real time test bed. Due to the limitations associated with them a novel real time SFF model and algorithm has been developed for real time hardware in the loop test bed. This real time SFF model was downloaded to a setup of two single board real time computers connected to each other through a serial port.

The hardware of the relative navigation system has been designed and developed in this research. Single frequency GPS has been used as the relative navigation sensor. A model has been developed to estimate the dynamic characteristics of the GPS receiver for LEO orbits. This model simulates the relative dynamics of the GPS constellation and the LEO satellite. S3C2410 has been selected as the navigation processor. Radio transceivers are used to exchange data between the satellites. A uC/OS-II real time operating system is used in the navigation software code. The Hardware and software structure has been developed to simulate both centralized and decentralized approaches. A novel relative navigation algorithm is discussed.

A Graphical Interface Software (GIS) has also been developed to initialize hardware and software prior any simulation and test run. It synchronizes the start of simulation on all real time computers, relative navigation systems and other components/simulator. It captures real time data from the deputy satellite. It also provides different debug and test options.

Offline simulations were run for three models Hill's, COEPOKE and RTSFF models by considering gravitational and atmospheric perturbation. The comparison of the results showed that the RTSFF model is able to simulate formations both in circular and elliptical orbits satisfactorily. The analysis of these models on the basis of their mathematical derivations showed that the RTSFF model gives better results than other models. The worth of this model for the real time test bed has been shown by running real time simulations onto a network of two synchronized single board real time computers and when the results were compared with offline simulation results they were found to be quite satisfactory.

The hardware of the relative navigation system has been checked and tested. A basic infrastructure for navigation software has been developed. Different open loop tests have been run to verify the working of the hardware and software and these have verified the correct functioning of the system. The navigation algorithm could not be implemented in the embedded environment, Due to the unavailability of GPS simulator, no close loop simulation or test has been performed. The functionality of all the software, models, hardware and their operational control under GIS software has been checked in real time environment and found to be working correctly and are ready to be run in any open loop or closed loop simulation test.

Table of Contents

Acknowledgement

I am on the verge of a big milestone and I would like to say that this would have not been possible without the assistance and support of different people. First of all I would like to thanks my Supervisor Dr. Adrian Tatnall, whose patience, expertise, and generosity made this work possible. I could not forget his support encouragement and guidance at difficult stages during this work. I am also grateful for his dedication to the task of reading through many drafts that I have produced.

I would like to gratitude my astronautics colleagues, especially Francois for his valued help during schematic and PCB design. I thank the staff for their support. I am thankful to Dr. Anna Barney for helping me with references

At this moment of happiness I would like to pay special tribute to my wife and children, Naveeda Riaz, Muhammad Jaazal Riaz, Snya Riaz, Khadira Riaz for their great patience, support and understanding while pursuing my research.

Acronyms

SFF	Spacecraft Formation Flying
LEO	Low Earth Orbit
RTSFF	Real Time Satellite Formation Flying
RTHIL	Real Time Hardware in the Loop
TOW	Time of Week
RTOS	Real Time Operating System
FIFO	FIRST IN FIRST OUT
PCB	Printed Circuit Board
RF	Radio Frequency
GIS	Graphical Interface Software
SDRAM	Synchronous Dynamic Random Access Memory
GPS	Global Positioning System
EEPROM	Electrical Erasable Programmable ROM

Definitions:

Geocentric Frame of Reference (IJK-frame): It has its origin at the earth's centre. The fundamental plane is the equator and the positive X-axis points in the vernal equinox direction. The Z-axis points in the direction on North Pole and Y-axis at the right angle to the plane containing X and Z axes. It is non-rotating frame with respect to the stars.

Formation Flying Frame of Reference (RSW-frame): It is a reference frame in a satellite formation flying and is attached to the master satellite. R-axis called radial axis is along the radius vector, S-axis is at right angle in the direction of satellite velocity, W is at right angle to the plane containing R and S axes to complete a right angle coordinate system.

Real Time Computers: Real Time Computers are capable of simulating and responding to an event at the same speed as it would occur in real life

Embedded Real Time Operating System (RTOS) (Labrose [1]) It manages the time of the microprocessor. It responds to a time critical event in a deterministic manner and within a prescribed time before the failure occurs.

Message Queue(Labrose [1]): A RTOS facility that is used to exchange data between the tasks.

Semaphore(Labrose [1]): A mechanism used to control the access to shared resources and allow two tasks to synchronize their activities.

FIFO buffer: First In First Out memory, usually available on the communication ports to temporarily store the data.

COM1, COM2: The serial ports on the computers used to exchange data with another device or computer

OSStartHighRdy( ): A operating system routine that depends on processor architecture. It is required to write this routine

OSCtxSw( ): Context switch routine and again depend on processor architecture.

OSIntCtxSw( ): Interrupt context switch routine

OSTickISR( ): Operating System timer tick routine. It programs a microcontroller timer for timer tick. This timer tick generates an interrupt to the operating system

Symbols

a – Semi-major axis
e - Eccentricity
ω - Argument of perigee
Ω - Right ascension of ascending node
i - Inclination with equatorial plane
θ - True anomaly
M- Mean Anomaly
n- Mean motion
di - Inclination difference between master and deputy satellite
da - Semi major axis difference between master and deputy satellite
de - Eccentricity difference between master and deputy satellite
$d\omega$ - Argument of perigee difference between master and deputy satellite
$d\Omega$ - Right ascension difference between master and deputy satellite
dM - Mean anomaly difference between master and deputy satellite
$d\nu$ - True anomaly difference between master and deputy satellite
h - Angular momentum
μ – Gravitational earth constant, $\mu = 3.986 \times 10^{14} m^3 / \sec^2$
r_e – Radius of earth, $r_e = 6371 Km$
J_2 - Geo potential coefficient representing Earth's oblateness
c- Speed of light in vacuum
p- Semilatus rectum
C_d - Drag Coefficient
R- Position vector of satellite
A- Effective area of satellite normal to velocity
u- argument of latitude, $u = \omega + \theta$
Φ_j^i - Integrated carrier phase range between i^{th} satellite and j^{th} receiver (in meters)
δt - Clock error
d_{ion} -Ionospheric delay
N- Integer ambiguity
Lz - Temperature Gradient at Geometric height z
g_0 - Sea level Gravity Constant, $g_0 = 9.81 \ m/\sec^2$
Tm -Molecular Temperature (K)
λ - Wavelength
Subscripts
d – Refers to the deputy satellite
m – Refers to the master satellite

Chapter 1

Introduction

Satellites missions that are flown today usually require a single satellite. They operate independently and fulfill all the mission requirements. To build a large satellite and then to launch it, is very expensive and sometimes it becomes impractical to develop a large enough satellite to fulfill a specific mission requirements. Instead of developing and designing a single large independent satellite, a new type of spacecraft system architecture is being proposed for some missions. In this new spacecraft system architecture a cluster of small spacecraft is used to achieve the mission. In this cluster of satellites, several satellites fly very close to each other, with a high level of inter satellite communications and synchronization. The payload and processing load are shared among all the satellites in the cluster. All satellites maintain their relative position with respect to each other and from the Master Satellite in order to achieve the desired mission performance. The act of maintaining and reconfiguring the shape and size of satellite constellation is collectively called Formation Flying. This formation is kept in a predefined controlled window through suitable controllers which counteract the affect of differential perturbations. The benefits associated with formation flying are

- The distributed architecture can enable missions to be designed that would otherwise be impossible, hence enable improved science missions.
- The distributed architecture makes it less vulnerable to failure and hence increases reliability. As long as the cluster is designed to avoid single-point failures, by distributing critical functions across the cluster, the mission can still survive if one or more satellites cease functioning.
- Adaptability and expandability is also improved by allowing for the possibility of launching additional satellites.
- Mission and manufacturing cost can be reduced by mass producing several similar satellites.

1.1- Motivation and Objectives:

The great benefits associated with SFF (satellite formation flying) have led to considerable research around the world. As it is a new field of research and most of the

theories, algorithms, technologies associated with them are still unproven their implementation in practice brings some inherent challenges and risks with them. Their performance in critical situations is still open to question. It is therefore particularly useful to develop a real time platform, where these theories, technologies, algorithms and their coordination in open and close loop can be tested in scenarios close to those actually expected.

The main objective of this thesis is to develop a low cost, real time hardware in the loop (RTHIL) test bed for low earth orbit autonomous spacecraft formation flying consisting of nano satellites(less than 10Kg), which can integrate original hardware in the loop. This test bed will provide laboratory testing grounds for new concepts in software and hardware for coordinating the position and trajectory of multiple satellites. Following requirements were set for this test bed.

- ✓ It can simulate a LEO satellite formation flying mission with relative position accuracies from few centimeters to many meters. It can simulate a formation consisting of 2 satellites and can be expanded easily by adding more satellites in the formation.
- ✓ It can simulate both centralized and decentralized formation control concept.
- ✓ Can simulate both circular and elliptical formation flying orbits (e=0 and e<1)
- ✓ It can accommodate any perturbation (gravitational, atmospheric, thrust or any noise designed by the user)
- ✓ It can integrate hardware in the loop, like navigation sensor, inter satellite communication device, thrusters, GPS simulator etc.
- ✓ Design the test bed in modular way so that any component in the loop can easily be replaced by its hardware or software versions, without disturbing the architecture of the test bed, like in the absence of hardware GPS simulator a software GPS simulator could be used. In the absence of thrusters its transfer function can be used.
- ✓ Can provide laboratory testing grounds for new concepts in software and hardware for coordinating the position and trajectory of multiple satellites. It should provide access to embedded software code so that new navigation concept and algorithm could be tested (like least square estimation, extended Kalman filter, Adaptive filter, single difference or double differenced measurement model and any combination of these). Different other options for inter satellite communication hardware and software algorithm can be tested. Radio communication between the satellites can be in broadcast mode, addressed mode

and can use frequency hoping and spread spectrum technique. Data transfer protocol, data check sum algorithm, and affect of FIFO buffer in radio transceiver can be optimized. For a specific mission. Test bed should be able to repeat any particular simulation for any number of times with same input parameters.

- ✓ Can provide a real time environment to validate components level technology and also to validate guidance control and relative navigation algorithm. The performance of different hardware (GPS based navigation sensor,)
- ✓ Can help to estimate the necessary hardware requirement for a specific missions, like processing power, number of I/Os, power consumptions etc.
- ✓ The real time test bed should give stable and reasonably accurate results for at least one orbit because during one time period the over all affect of all perturbing forces on secular and periodic variations in relative states becomes clear.

1.2- Concept:

In this project a Real Time Hardware in the Loop (RTHIL) Spacecraft Formation Flying Test Bed has been developed. A closed form loop for this test bed is presented in Fig 1.1. In a hardware in the loop approach , orignal components as much as possible, have been connected in the same order as they will be used on the orignal system. Any hardware components, which are not available have been replaced with its transfer function in the loop. Similarly dynamics of the system have been modeled and programed by using some suitable language in a real time computers.These software models are then interfaced with hardware components through I/O interfacing cards installed in real time computers. By using hardware in the loop real time simulation, it is possible to develop and verify new algorithms, to verify new hardware technologies and can simulate scenarios that lead to failures.

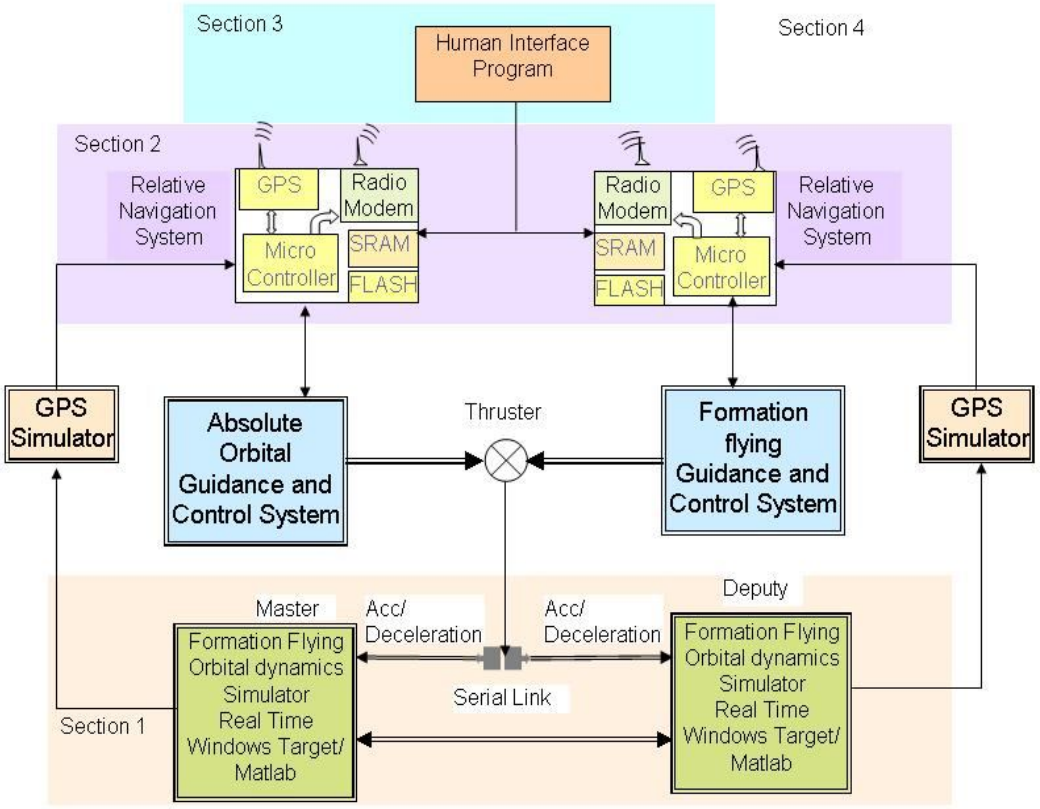


Figure 1.1: Hardware in the Loop real time test bed

The real time hardware in the loop simulation (RTHIL) software model developed in this research is used to simulate the satellite's absolute dynamic and relative states in real time. Absolute states can be used by a GPS Simulator to give the true dynamical impression to the GPS receivers, where as relative states are used as reference to verify the relative accuracies of the relative navigation system and algorithm. A GPS simulator generates GPS signals according to the input dynamics and after data acquisition GPS receivers generate raw data and timing data. It should have at least two output channels to simulate two vehicles formation flying. After data acquisition, sorting and differencing, navigation processor calculates absolute state vectors and relative state vector. A guidance algorithm generates the error difference of the current state vector from reference state vector and control algorithm generates control signals to the thrusters. Outputs from the thrusters in the form of torque are fed to a spacecraft formation flying model. This model adjusts its dynamics on the basis of the feedback thrust and generates new state vectors for next simulation step. As this RTHIL has a

modular design, any component in the loop can have a software or hardware representation by maintaining interfaces with other components in the loop.

During the real time simulation, absolute states and relative states are saved along with their time tags. A navigation algorithm calculates absolute and relative states by using raw observations. These states are then compared with the states given by the real time SFF model for the same time tag. This comparison will give the accuracy estimation of the navigation algorithm. This RTHIL test bed can be run in both close and open loop. In close loop guidance, control and thrusters (software model or actual hardware with feedback system) will be in the loop with the RT SFF model, GPS Simulator and relative navigation system. Close loop scheme is normally used to validate the control algorithm, fuel balance estimation, collision avoidance analysis and testing of new hardware technologies. Fig 1.1 shows a closed loop schematic of the RTHIL test bed.

An open loop scheme does not include guidance and control algorithms or any feedback loop. Relative states are uncontrolled and will move apart under differential perturbation forces. The real time SFF model gives absolute and relative states. This scheme is used to verify navigation algorithms, relative navigation system and its performance. Figure 1.2 shows an open loop schematic of the RTHIL test bed. For quick initial navigation algorithm estimation, an open loop scheme can be simplified further and this is discussed in chapter 6.

This test bed has different simulation options, it can run a real time simulation with RT SFF model, GPS Simulator, and relative navigation system or it can run a real time simulation by using prerecorded GPS raw observations (open loop setup). In this way it is possible to repeat any specific formation flying scenario during algorithm development.

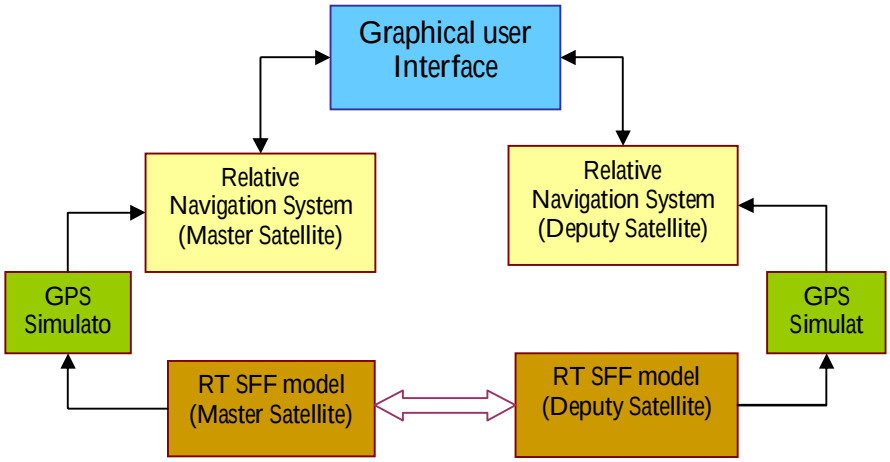


Figure (1. 2) : Open Loop RTHIL Test Bed

The graphical user interface can be used to initialize different parameters of the navigation system including GPS receivers, radio modems and navigation software. It can be used to do the following

- configure the relative navigation software for centralized or decentralized approach.
- capture and display different data in real time from relative navigation systems.
- Provide the option for different simulation, like real time simulation with GPS simulator in loop, real time simulation with pre-recorded GPS data, run simulation with software GPS simulator in loop.
- Run different test programs to verify the working of different hardware and software tasks.

It was originally planned to use a GPS simulator to simulate and generate high dynamic GPS signals. The relative navigation system is designed in such a way so that it can be connected directly to hardware GPS Simulator or software based GPS simulator. In the 2nd option the GPS receivers will be bypassed and the output of the GPS Simulator will be fed directly to the navigation processor to be used for navigation algorithms. A software based GPS simulator is a very cost effective alternative to a hardware based Simulator. It is less

expensive, easy to use, has a fast development time, but it is less accurate. In this research a GPS simulator was not available and so this interface has not been tested.

1.3- Thesis Overview:

The rest of the thesis is divided into four main sections as shown in figure 1.1

1. Real Time Formation Flying Modeling, Simulations and Setup
    a. Chapter 2---SFF Modeling History
    b. Chapter 3---Formation Flying Modeling and Simulation
2. Relative Navigation System
    a. Chapter 4----GPS as Relative Navigation Sensor
    b. Chapter 5----Relative Navigation System
3. Graphical Interface Program
    a. Chapter 6----Graphical Interface Software
4. RTHILS setups
    a. Chapter 7----RTHIL Setups, Recommendations and conclusions

The first section consists of two chapters. In the first chapter (chapter 2) some existing SFF models are discussed. Accuracies of these models are analyzed by considering their limitations and the assumptions used in their derivations. Their potential use in this RTHIL test bed is also discussed. It is concluded that these existing SFF software models are not adequate for this RTHIL test bed. In chapter 3, an efficient and distinctive real time SFF model and algorithm is developed by using a geometrical approach. Offline simulations in Matlab/SIMULINK were run for this real time SFF model and a comparison of results with existing SFF models is presented. A basic real time formation flying setup is then developed by using two single board computers. Real time simulations were run and a comparison of results with offline simulations are presented in this chapter.

Section 2 consists of two chapters. Chapter 4 discusses Global Positioning System (GPS), and its possible use for space use and its potential as a relative and absolute navigation sensor. It also discusses hardware and software GPS simulators and their role in RTHIL. Chapter 5 deals with relative navigation hardware design and development. Navigation

software development is also presented in this chapter. The real time operating system for the navigation system and its porting to the navigation platform is discussed here.
Section 3 (Chapter 6) describes the importance of graphical n interface software and its development. Different simulation parameters and options are discussed in this chapter. Its overall role and capability in the loop is also discussed.

Section 4 (Chapter 7) summarizes the research. In this chapter the overall final RTHIL setup and its capabilities are discussed. The different configurations of RTHIL setup and their importance towards the hardware/software components validation are discussed here step by step. The conclusions of the research are presented. Future recommendations and improvements are suggested.

1.4 Spacecraft Formation Flying--Literature Review:

The main advantage of satellite formation flying is replacing future large satellite with a number of smaller, less massive satellites. The ability to spread functionality across a coordinated satellite cluster, can lead to a significant improvement in the functionality that can be achieved with a large single spacecraft. Satellites in a formation can be reconfigured on-orbit to change the architecture and perform a different mission. Additional satellites can be added to the formation at any time to enhance the overall performance and mission capability of the formation. The distributive architecture of the formation makes them less vulnerable to single point or single satellite failure and hence increases overall reliability. The production costs of a number of similar satellites and their subsystems can be considerably reduced.

Satellite formation flying does, however, also carries some disadvantages. In some cases the overall mass of the formation can be more than single large satellite, which can increase the overall cost of the launch. Initialization and maintenance of formation are challenging tasks, especially when satellites are launched separately. Misalignment of sensors, and electrical and thermal noise may degrade the measurements, if they are mounted on different satellites. Formation flying satellites normally require more fuel to maintain relative separation, and this can reduce the over all life of the component satellites and hence of the formation. Shaw et al. show that distributed systems generally are more expensive than single satellite due to the requirements of additional guidance and control systems (hardware and software).

The benefits of satellite formation flying, however, are significant for many missions e.g. a single aperture telescope would not be able to achieve the same resolution that can be achieved by spreading apertures on different satellites hundred of meters apart. Similarly earth imagery can be enhanced by merging images from different sensors mounted on different satellites in a formation.

Different formation flying missions have been planned by different organizations and institutes. Some of them are for science purposes and some of them are for military purposes. Currently the two formation flying missions are flying into the space. GRACE is a leader-follower formation consists of two satellites in LEO and other formation is Earth-Orbiter-1/Landsat7. TechSat21 was proposed to be launched in 2003, but this mission was not carried forward. Earth-Orbiter-1 (EO-1) was launched by the NASA to demonstrate autonomous formation flying in leader-follower formation with LandSat7. These satellites do not maintain precise relative positions, but maintain temporal separation along the ground track. EO-1 is equipped with autonomous formation flying technology [62]. It initially flies 50m above the Landsat7, but due to air drag its altitude decreases and it moves closer to Landsat7. When the separation reaches the specified limits, the formation flying controller comes into action and EO-1 maneuvers back to the higher altitude. The GRACE [34] formation consists of two satellites flying in leader-follower formation and separated by 220Km. The satellites in the formation measure inter satellite separation very precisely. The differential earth gravity between these two satellites is detected by variations in relative separation.

There are number of other formation flying missions that have been proposed for future science and military missions. Major project in the planning and construction phase are the NASA terrestrial planet finder (TPF), DARWIN extra solar planet search, laser antenna LISA for the detection of gravity waves, earth observation SAR-Lupe, Cartwheel and many more. A list of formation flying missions is presented in Table 1. MUSTANG [65] formation flying project was a joint effort of university and UK Space industry. Its purpose was to address the issues related to distributed systems technologies and formation flying. Cranfield University's Space Research Center and the University of Southampton jointly led the project with industrial support provided by Astrium UK. It was to consist of two nano satellites (each with mass less than 10Kg). A comprehensive literature review on formation flying can be found in Robert's PhD thesis [66].

Mission Program	Launch	Objective
XEUS- x-ray evolving Universe Spectroscopy Mission	2010	Detect and Study black holes and dark matter
SMART	2010	Formation Flying demonstration/ experimental setup for DARWIN mission
POES	1988-2015	Storm tracking, long range weather forecasting,
The Afternoon or 'A-Train' Constellation	2006-2008	Measuring aerosols, clouds, temperature, relative humidity, and radiative fluxes simultaneously all over of the globe during all the seasons
SCOPE		Magnetosphere observation mission
TanDEM-X[63]	2008-2009	Generation of world-wide consistent, timely, high-precision, Digital Elevation Models corresponding to the DTED-3 standards.
TPF[64]	2006-2014	To find and characterize earth like planet orbiting other stars

The concept of spacecraft formation flying came from the spacecraft rendezvous and docking maneuvers. In both concepts orbital corrections are applied to control the relative orbit between the two spacecraft. Spacecraft rendezvous and docking is slow and is a short time phenomenon and so any simplifications or small modeling errors will have a minimal impact on the control performance. Spacecraft formation flying, on the other hand, is supposed to be maintained over the entire life time span of the spacecraft and is a long time phenomena. Any simplifications or small errors in a formation flying orbital model will cost more fuel consumption or even lead to the formation becoming unstable. A precise relative orbital dynamic model is therefore, very important for spacecraft formation flying design and analysis.

1.4.1- SFF Models:

The analysis of key issues like initial array formation, array reconfiguration, array control, collision avoidance, fuel balance and stability, require an accurate formation flying model. Initially Hill's (Clohessy [2]) equations were used to study formation flying dynamics. They are a set of linearized differential equations, which can be solved analytically. Hill's equations are also used in the design of control laws. However, Hill's equations assume that the reference orbit is circular, the Earth is spherically symmetric, and the target orbit is very close to the reference orbit such that the relative external perturbing force and the nonlinear terms in the relative motion can be neglected.

Euler and Shulman [3] developed quadratic relative motion equations with respect to elliptical orbit in term of true anomaly. They did not give complete analytical solution of their equations. Several plots were given for out of plane motion, which shows improvement over a linear solution, although it is not clear which linear solution this is. Their solution were found to be valid for approximately half of the period of the reference orbit.

Werlwas[4] developed a new solution of the relative equation of the motion. His approach differed to others by the fact that he placed the reference frame on the deputy satellite rather than a point on a reference orbit. This formulation leads to a complicated but accurate solution. A limitation of the solution was that there can be no initial out of plane displacement, although an initial out of plane velocity is allowable. The form of the solution depends on the initial conditions, which is a disadvantage as well.

Melton[5, 6] developed a linear set of relative motion equations for an elliptical reference orbit including second order powers of eccentricity in the approximation. The solution was given explicitly in terms of time for both rectangular and cylindrical coordinates, and can include perturbations through a convolution integral. The cylindrical coordinate representation was found to be more accurate than the rectangular coordinate solution. Secular terms are present which limits the time interval over which the solution will be valid.

S.A Schweighart [7, 8] has proposed a method to include the J2 effect in the Hill's equations and derived another set of linearized constant coefficient differential equations with the J2 effect included. These equations are very similar to Hill's equations in form. The radial and along track separation are still coupled. These equations can also be used to design optimal control law that take in to account the affect of J2 perturbation. Due to the linearization, these

equations do not exactly predict the correct initial velocity. This in result creates a drift in the in-track direction of a few meters per orbit. Due to the limitations associated with Hill's equations, many other authors have proposed their own formation flying models.

An analytical method by Alfriend, and Gim[9], derived the state transition matrix for both osculating and mean elements for the relative motion of two neighboring satellites when the reference satellite (Chief) is in an elliptic orbit and both satellites are subjected to the $J2$ perturbation. In this method the relationship between the state vector and relative variables is obtained in matrix form instead of directly solving the differential equations of relative motions. Due to the neglected nonlinear terms in this method possess some limitations. The relative positional errors have been found to increase with the increase in the magnitude of the spacecraft separation and orbit eccentricity.

Sabol et al[10] presented COWPOKE equations based on Keplerian orbital elements to study the formation flying models for elliptical orbits. The COWPOKE equations use the difference in the orbital elements and do not account for any perturbations. According to Sabol, the equations should provide sufficient accuracy for low-fidelity simulation applications, although they are inadequate for high precision analysis. It does not give absolute states of each satellite in the formation. Due to assumptions and simplifications, it shows an error in the range of few cm to few Kilometers [10] with respect to true data. It gives relative positions in spherical coordinate system and transforming them into RSW frame introduces some errors. This error is more prominent in radial separation and depends upon along track separation.

These formation flying models use simplifications in their derivation which introduces inaccuracies. They do not consider the affect of differential perturbations. Also they do not give absolute state vectors. They use mean orbital elements, hence they cannot show the affect of periodic variations on formation flying. These SFF models do not fulfill the structural requirements necessary for this real time HIL test bed (described in section 2.1), hence another real time SFF model is developed in this thesis.

1.4.2- Formation Flying Configuration Strategy:

There are several formation flying strategies from a guidance and control point of view. Some of them give better control accuracies but at the cost of extra complexities and more computational burden. Similarly some strategies are more robust, while some are subject to single point failure. These strategies are

i- Independent Control:

All the satellites in the cluster are independent and there is no exchange of information between them. Each satellite in a multi satellite cluster determines its control actions based solely on information about its present state and its objectives. There is no need of any relative navigation system. The cluster of GPS satellites is an example of this kind.

ii- Master/Slave Control (Centralized approach):

In a master/slave control scheme the "master" satellite receives the state information from all satellites and decides the action for each of them to take. The "slaves" simply send state data to the master and receive commands in return, performing no control law computation themselves. This results in a computational load on one processor. Furthermore this approach is not robust to single point failure and it is not reconfigurable or flexible. In addition if the size of formation increases, then this computation could become extremely difficult to perform in real time.

iii- Leader-follower control (Decentralized Approach):

This involves the leader sending its state to the follower satellites. The followers attempt to track the leader's state, with some offset to avoid collisions. Each follower estimates its state in the formation and solutions are sent back to the Leader to form an entire fleet estimate. Here the computational load is spread across the satellites in the formation. If the Leader fails, any other satellite can take its place. This makes Leader-Follower configuration more robust. The formation can also be scaled to any number of vehicles. In this thesis leader-follower approach is used.

iv- Distributed control:

Distributed control is more complex, with each satellite having knowledge of the state of all the others, and determining its own motion based on this information.

1.4.3- Relative Navigation System:

All the key issues in spacecraft formation flying are the function of relative state vectors. Hence a relative navigation system should be accurate and reliable and have real time response to events. A general navigation system for SFF not only is able to give accurate

relative state information in real time but also absolute position of each satellite, because some formation flying control strategies and relative navigation algorithm needs absolute states[11]. The SFF considered in this research consists of LEO nano satellites. The mass of nanosatellites are a few Kilograms (usually less than 10Kg), like SPUTNIK-40 was only 3Kg, TUBSAT-N and TUBSAT-N1 were 8 and 3Kg respectively [12]. Because of their small mass, a relative navigation system should be compact in size (few cm), low mass (few grams), low acquisition cost, consume small amount of power (few Watts) and still enable high navigation accuracies (relative position accuracy in cm). Sensors that could be used for relative navigation systems are laser-based distance sensors, inertial instrument unit and GPS. Laser based distance sensors required highly stable laser source. Inertial measurement unit require calibration and also not accurate enough for relative states estimations. These sensors are a few Kilograms for nano satellites and consume large power.

Based on the required relative position accuracy values, formation flying mission can categorized as coarse formation flying and precise formation flying. Coarse formation flying missions are those that require relative positional accuracy from few centimeters to few meters. One such formation flying mission is the formation of synthetic aperture radar such as the Astrium's TerraSAR-L[13]. These types of formations are typically considered in LEO remote sensing missions where each satellite is an individual element of a large virtual antenna formed by the formation. By sharing the individual measurements, the resolution of the spacecraft cluster is potentially much higher than the resolution of any individual spacecraft. For such missions it might be unnecessary to maintain extremely precise relative positions. Another example of coarse formation flying is a stereo imaging mission like the ThreeCornerSat[14]. Stereo imaging only requires a nominal spacing of tens of Kilometers. The accuracy requirements for such missions are in the order of few meters. The precise formation flying missions are those that require relative position accuracy from millimeter to few centimeters. Space borne optical stellar interferometry involves the usage of mirrors to reflect the collected light from a distant body to one of the spacecraft that will also serve as the combiner. In order to meet the missions goals in such a missions, the control system must maintain the distances between spacecraft to an accuracy of at least 1-2 cm and in many cases the accuracy required is a few microns. Another example of precise formation flying mission is the proposed X-Ray spectroscopy mission XEUS[15], that aims to study black holes and intergalactic medium. The

mission proposes to use two LEO satellites with a separation distance of 50meters. The relative position accuracy has to be maintained with an accuracy of less than 1 millimeter.

For satellites in low Earth orbit (LEO) the use of the Global Positioning System (GPS) poses an attractive alternative to other relative navigation sensors in terms of accuracy, robustness, flexibility, and acquisition cost. GPS provides not only accurate relative states but also provide absolute states of the satellite. GPS provides highly accurate timing information for onboard time synchronization and enables simultaneous measurements from spacecraft within the formations. If multiple antennas are included on one vehicle, GPS is capable of attitude determination as well. The use of GPS as relative navigation sensor has earlier been assessed and demonstrated by different missions, like LandSat-7, EO-1 formations. Orion is a satellite formation flying mission, consist of three micro satellites in LEO orbit. The relative position accuracy required is 5 meters. GPS has been used on this formation for relative separation and velocity calculation[16]. The GRACE mission has validate the attainable relative navigation accuracy by using GPS is little better than 1mm. (Ref: Relative positioning in satellite formation flying using GPS by P.W.L van Barneld).Ebinuma (2002) demonstrated the use of single frequency GPS receiver for relative navigation in satellite formation flying. By applying kinematic Kalman filtering algorithm on single difference measurements, he demonstrated accuracies of 0.5m and 1mm/s (3D rms) respectively for relative position and velocity over a 10Km baseline using double difference carrier phase measurements. Busse (2003) presented a relative navigation algorithm for carrier phase differential GPS (CDGPS). He used a precise extended Kalman filter that was based on simple measurement model and a linear Keplerian propagation model. For a 1 Km base line separation, CDGPS algorithm demonstrate 1-2 cm relative position accuracy and <0.5mm/s (3D rms) relative velocity accuracy for low earth orbit formations. PRISMA[17] comprises of two micro satellite formations into Low Earth Orbit and its launch is scheduled in 2009. The absolute and relative position accuracy requirement is 2m and 0.1m respectively. GPS has been selected as relative navigation sensor for this formation

With these range of capabilities, demonstrated accuracies and experimental results make the GPS an ideal candidate as relative navigation sensor for both coarse and fine formation flying missions. Therefore GPS has been selected as relative navigation sensor for this test bed.

A relative navigation system uses either a centralized approach or a decentralized approach to calculate relative states between the satellites in formation. In the decentralized approach the computational load is distributed across the satellites in the formation and it is more robust. Formation can be expanded to any number of satellites. In this research the decentralized approach is adopted. In decentralize approach, master satellite transmit its absolute states, raw measurements, and timing information to follower satellites. The raw measurements from local receivers and the information send by the master satellite are used to estimate relative states. There are two approaches; one is to estimate relative states by using iterative least square approach. Least square approach is simple to implement. It does not require any measurement or user dynamic model, but on the other hand it is less accurate. Some authors have suggested, smoothing techniques to improve the results. Another approach is to use an extended Kalman filtering (EKF) technique. Kalman filtering algorithm requires a measurement model and a process dynamic model and in the case of SFF, both measurement and relative orbital dynamic models are known. Single difference equation for carrier phase and Doppler measurement in earth centered earth fixed (ECEF) coordinate frame are used as measurement model. Relative orbital dynamic model in ECEF frame is developed by Busse in his PhD thesis.[18] Relative state vectors calculated in ECEF frame are then transformed into the master frame of reference. In this thesis the Kalman filtering technique is used for relative state estimation. A relative orbital dynamic and measurement models in master frame of reference is developed as it will simplify the relative navigation algorithm. A radio modem is used to broadcast information between the satellites in a formation. There are two kind of radio modems exist on the basis of their functioning.

1. Frequency hopping
2. Direct sequence radio modem

Frequency hoping radio modems send packets of data on different frequency channels and the radio modem walks through all the channels to transmit or receive information. In this it also minimize the affect of interference at any particular frequency channel. This type of radio modems has the ability for point to point, point to multipoint, multipoint to multipoint transmission. They use unique MAC address to talk to specific transceiver. They have simple design and are less expensive than direct sequence modems. Frequency hopping imposed some computational overhead. Direct sequence radio modem spread the signal on a larger band by multiplexing it with a code, thus minimizing the affect of interference at a frequency.

It has less overhead and dead time, but it is more complex and more expensive. Due to its low cost, simple design and additional flexibility in transmission, frequency hopping radio modems are used in this research.

1.4.4- RTHIL Simulation test bed:

To validate the navigation algorithm, hardware components, guidance and control algorithm, real time hardware in the loop simulations are performed. Software models are used to simulate zero gravity environment and dynamics. It is recommended to connect as much as possible, the true hardware components in the same sequence as it will be in original system. For navigation algorithm open loop RTHIL is sufficient for its validation. For the validation of guidance and control algorithm, RTHIL simulation should be performed in closed loop. Many efforts have been done by different researcher all over the world to validate different aspects of SFF in real time.

Dr Jonathan H. at Stanford University developed a free floating test bed with an indoor pseudolite GPS environment. This test bed was used to study the relative position accuracy and control architecture for formation flying. At MIT University he developed a test bed consisting of three moving trucks with a pseudolite GPS system on them [19]**.** These were 2-D test bed. Another effort at Stanford University in this regard was demonstrated by using autonomous flying blimps carrying GPS as relative navigation system by Jonathan H [20]. It was a three dimensional test bed. In these efforts it was demonstrated successfully that GPS can be used as robust formation sensing system to measure relative distances inside the formation with cm-level accuracies. Although these efforts did not include high orbital dynamics, but they show GPS as a strong candidate for relative navigation and also helped to develop and test navigation algorithm and initial control architecture for formation keeping.

At University of Southern California a SFF test bed is being developed [21], where a helicopter is used to represent a space vehicle. Laser imaging and vision base relative navigation system is used. This formation flying test bed is in the build process and yet no practical demonstration on any aspect of formation flying has been presented.

SPHERES is SFF test bed being developed by Space system Laboratory and MIT University by D. Miller [22]. It consists of three nano satellites, which has been installed in International Space Station in December 2006, where these satellite can perform controlled coordinated maneuvers in zero gravity environment. SPHERES is designed to validate relative attitude control and relative station keeping between the satellites, collision avoidance and fuel

balancing algorithms. Ultra sound waves and beacon are used to calculate position of the satellite inside ISS module. Researchers can perform only four test sessions in a year within each test session 20 minutes of time available. It did not provide the facility to use GPS as navigation sensor.

Formation control test bed[23] is developed in Jet Propulsion Laboratory by a group of scientist and students. The objectives of this test bed was to demonstrate formation flying as viable mission architecture for Terrestrial Planet Finder Mission and validate formation algorithm and simulation test bed for higher confidence performance prediction. FCT consist of three robots navigating on an air bearing floor, propelled by cold gas thrusters. Each robot contains an attitude platform supported on an spherical air bearing which provides three rotational degree of freedom. The sixth degree of freedom is provided by powered vertical stage.

NASA's Goddard Space Flight Center in collaboration with JPL and DoD has developed a based test bed for formation flying by Leitner; Alfriend [9, 24]. It can simulate not only relative control but also gives attitude control simulation. Actual hardware is not used in this test bed. For inter satellite communication TCP/IP Ethernet network is used.

These formation flying test beds generally either simulate only a small aspect of the SFF or they are very expensive with limited access and most of their software information is not accessible.

Chapter 2

SFF Modeling History

2.1 Introduction:

Formation flying problems have long been under consideration and many mathematical models have been developed to analyze the different aspects of relative motion of spacecrafts in formation flying. A formation flying model should be robust in the sense that it should simulate different possible formation flying scenarios and configurations and should not diverge for any set of orbital parameter. Any modeling errors and estimation can lead to a false fuel balance and can cause instability in the formation. Most of the models were designed to see the mean variations in relative states over the time in offline simulation. This project was looking for a real time SFF which can fulfill the following requirements.

Requirements:

a. It can be used for both circular and elliptical orbits (e=0 and e<1).
b. It should be able to simulate any number of satellites in a cluster (more than 2).
c. The real time formation flying model for hardware in the loop simulation should model the actual formation flying scenario as closely as possible and should be able to accommodate different hardware in real time simulation.
d. It should give not only relative separations, but also absolute states of each satellite in the cluster in order to close the hardware simulation loop through the GPS Simulator.
e. The model should be not only accurate but also modeled in such a way that it takes very little CPU time (<< 1sec) for execution in order to reduce any truncation errors.
f. Periodic as well as secular variations in orbital elements due to perturbations should be taken into account.
g. It should be able to simulate different formation flying configuration, e.g. Master-Slave control, Leader Follower control, Independent control, distributed control
h. It should be able to accommodate perturbation affects easily.

Many already developed SFF models were studied and analyzed in the light of above requirements. A brief description of these models and their suitability for RTHIL project is presented in this chapter.

2.2 Hill's Equations:

An early model was presented by Clohessy-Wiltshire (CW) in 1960 and is known as CW equations and was based on the Hill's Equations [2]. These equations are derived directly from the two body equation in a Cartesian coordinate system and were first used in rendezvous maneuvers, but later on researchers started using them for spacecraft formation flying. Hill's equations are linearized set of equations with constant coefficients that describe

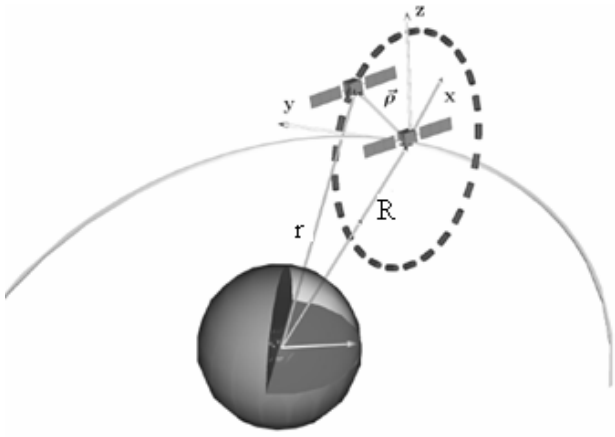


Figure (2.1): Formation flying geometry for Hill's Equation

the relative motions of two satellites under Keplerian central force.

$$\begin{aligned} \ddot{x} - 2\omega\dot{y} - 3\omega^2 x &= f_x \\ \ddot{y} + 2\omega\dot{x} &= f_y \\ \ddot{z} + \omega^2 z &= f_z \end{aligned} \qquad \text{..... (2. 1)}$$

Where *x, y, z* are radial along track and cross track distances (RSW frame) in master's frame of reference. Here the *x* and *y* motions are coupled while *z* motion is uncoupled from other motions. As these are linear equations and are in a simple form, they can be solved analytically, given the initial relative position (x_0, y_0, z_0) and initial relative velocity $(\dot{x}_0, \dot{y}_0, \dot{z}_0)$. The solution for the Hill's equation for the no thrust case is given by Vallado [25] as

$$x(t) = \frac{\dot{x}_0}{n}\sin(n \cdot t) - (3x_0 + \frac{2\dot{y}_0}{n})\cos(n \cdot t) + (4x_0 + \frac{2\dot{y}_0}{n})$$

$$y(t) = (6x_0 + \frac{4\dot{y}_0}{n})\sin(n \cdot t) + \frac{2\dot{x}_0}{n}\cos(n \cdot t) - (6nx_0 + 3\dot{y}_0)t + (y_0 - \frac{2\dot{x}_0}{n})$$

$$z(t) = z_0 \cos(n \cdot t) + \frac{\dot{z}_0}{n}\sin(n \cdot t) \qquad \text{..... (2. 2)}$$

The corresponding relative velocities by Vallado [25] are

$$\dot{x}(t) = \dot{x}_0 \cos(n \cdot t) + (3nx_0 + 2\dot{y}_0)\sin(n \cdot t)$$

$$\dot{y}(t) = (6nx_0 + 4\dot{y}_0)\cos(n \cdot t) - 2\dot{x}_0 \sin(n \cdot t) - (6nx_0 + 3\dot{y}_0) \qquad \text{..... (2. 3)}$$

$$\dot{z}(t) = -z_0 n \sin(n \cdot t) + \dot{z}_0 \cos(n \cdot t)$$

Since the orbits are circular the angular rate n of the local RSW frame is equal to mean motion of target satellite.

$$n = \sqrt{\frac{\mu}{r_{ref}^{\ 3}}}$$

The 3rd term in the along track separation equation for $\dot{y}(t)$ is a secular term and grows with time. To achieve a stable orbit this secular term should be zero. It is possible if

$$\dot{y}_0 = -2nx_0$$

Hill's equations are mostly used to design control laws since many control schemes require constant coefficient linear differential equations.

Hill's equations have some significant limitations [2, 7, 10]. These equations were derived for two closely placed satellites in near circular orbits. These are linearized equations and linearization introduces some modeling errors. The most significant limitation is that Hill's equations were derived under the assumptions that the earth is spherical and perturbation forces are negligible. This assumption is acceptable for the rendezvous maneuver problems, in which satellites are closely placed and they are undertaking short time maneuvers. In SFF

satellites are placed from meters to Kilometers apart and it is larger time scale mission. These equations do not give information about absolute positions of each satellite in the formation.

Schweighart and Sedwick[7] have proposed modifications to Hill's equation to include the J2 perturbation. They linearized J2 perturbation forces and included them in Hill's equations. In their model, Schweighart and Sedwick incorporated the gradient of J2 (delta J2) disturbance in the relative motion equations by taking the time average. This assumption causes the J2 disturbances to appear symmetrical about the reference orbital plane. For the in plane motion the assumption of the time averaged J2 disturbances does not introduce significant error, however for cross track motion, the error is significant. The equations are adjusted for the change in orbital time period and drift in the longitude of the ascending node to match the perturbed satellite. This adjustment is done by using an approximate solution and modeling errors introduced. The modified linear model proposed by Schweighart and Sedwick is [8]

$$\ddot{x} - 2(nc)\dot{y} - (5c^2 - 2)n^2 x = -3n^2 J_2 \frac{R_e^{\ 2}}{r_{ref}}\left(\frac{1}{2} - \frac{3Sin^2 i_{ref} Sin^2(kt)}{2} - \frac{(1+3Cos2i_{ref})}{8}\right)$$

$$\ddot{y} + 2(nc)\dot{x} = -3n^2 J_2 \frac{R_e^2}{r_{ref}} Sin^2 i_{ref} Sin(kt)Cos(kt) \qquad \text{..... (2. 4)}$$

$$\ddot{z} + (3c^2 - 2)n^2 z = 0$$

Where $k = n\sqrt{1+s} + \frac{3nJ_2R_e^2}{2r_{ref}^2}Cos^2 i$

and $s = \frac{3J_2R_e^2}{8r_{ref}^2}(1+3\cos 2i)$

Where *x, y, z* are radial along track and cross track distances and *c* is the mean motion.

2.3 The COWPOKE Model

Due to the limitations associated with Hill's equations, many other authors have produced their own SFF models. Sabol et. al. in Feb, 2003[10] proposed a SFF model to study formation flying for highly elliptical orbits. Sabol uses a geometrical approach for formation flying modeling. The resulting model called the COWPOKE model (Cluster orbits with Perturbations of Keplerian Elements) is based on orbital elements.

Sabol described an earth centered spherical frame and used differences in orbital elements of two satellites to determine their relative motion. The relative position difference is described by an altitude difference from a sphere having the radius of the reference satellite and angular components perpendicular to and along the reference satellite's direction of motion. The resulting spherical separation between two satellites in terms of cross track angular separation β, along track angular separation α and radial separation δr (Sabol, [10]) are as follows

$$\delta r = \frac{(a+\delta a)-(e+\delta e)^2)}{1+(e+\delta e)\cos(M+2e\sin M+\delta\theta)} - \frac{a(1-e^2)}{1+e\cos\theta}$$

$$\beta = \frac{\delta xt}{r} = -2\sin(\frac{\delta\Omega}{2})\sin(i)\cos\left(\omega+\frac{\delta\omega}{2}+M+2e\sin M+\frac{\delta\theta}{2}\right) \quad \text{..... (2. 5)}$$

$$+\delta i\sin(\omega+\delta\omega+M+2\sin M+\delta\theta)$$

$$\alpha = \frac{\delta at}{r} = (\delta\omega+\delta\theta)\cos(\delta i)+\delta\Omega\cos(i)$$

As it is derived in a spherical reference frame it is suited well to circular or near circular reference orbits, however when the orbital eccentricity becomes large, the reference frame is no longer spherical and the relative separation calculated by using the above set of equations will not be accurate. In addition the truncation of the Fourier-Bessel expansion for true anomaly at higher eccentricities can introduce significant errors. Perturbations are not considered in the model. It is good for closely placed satellites. Only linearized components are considered in the derivation of radial component. A method is suggested by one of Sabol's students (Catlin K. A [26, 27]) to use a variation of parameters method to include the affect of J2 and atmospheric drag perturbation.

2.4 Balaji's Geometrical Model:

In this model the relative coordinates of the deputy satellite is calculated from the master satellite's frame of reference in terms of the master and deputy satellite mean orbital elements [28-31]. Balaji defined a local frame of reference associated with master frame of reference and defined relative separations in this Cartesian coordinate frame.

The radial separation $x_r(t)$ is along the master's radius vector, $y_r(t)$ is in the direction of motion 90 deg to the radial vector, $z_r(t)$ is perpendicular to the plane containing $x_r(t)$ and $y_r(t)$ to complete a rectangular coordinate system. The relative position of the Deputy Satellite in the master frame of reference is given by Balaji S. K. [28-31] as.

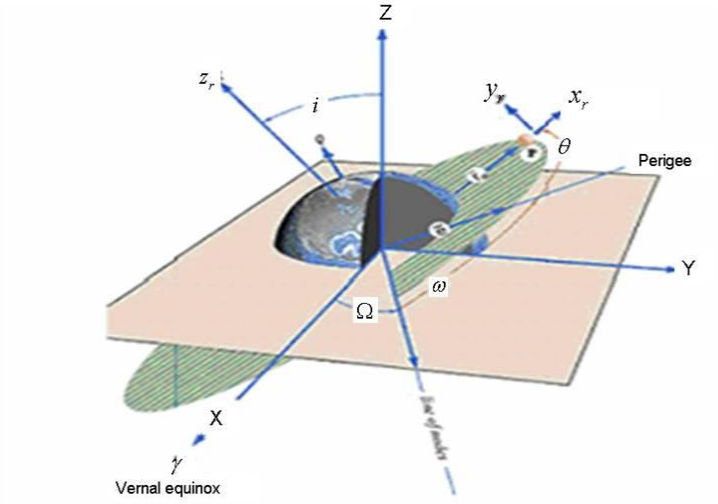


Figure 2.2: Orbital Frame and Relative Geometry

$$
\begin{aligned}
x(t) = {} & (\cos(u_m)\cos(\Omega_m)\cos(\Omega_d)\cos(u_d) - \cos(u_m)\cos(\Omega_m)\sin(\Omega_d)\sin(u_d) \\
& - \sin(u_m)\cos(I_m)\sin(\Omega_m)\sin(\Omega_d)\cos(I_d)\sin(u_d) + \cos(u_m)\sin(\Omega_m)\sin(\Omega_d)\cos(u_d) \\
& + \cos(u_m)\sin(\Omega_m)\cos(\Omega_d)\cos(I_d)\sin(u_d) + \sin(u_m)\cos(I_m)\cos(\Omega_m)\sin(\Omega_d)\cos(u_d) \\
& + \sin(u_m)\cos(I_m)\cos(\Omega_m)\cos(\Omega_d)\cos(I_d)\sin(u_d) \\
& + \sin(u_m)\sin(I_m)\sin(I_d)\sin(u_d)) \times R_d(t) - R_m(t)
\end{aligned}
$$

$$
\begin{aligned}
y(t) = [&-\sin(u_m)\cos(\Omega_m)\cos(\Omega_d)\cos(u_d) + \sin(u_m)\cos(\Omega_m)\sin(\Omega_d)\cos(I_d)\sin(u_d) \\
&-\cos(u_m)\cos(I_m)\sin(\Omega_m)\cos(\Omega_d)\cos(u_d) + \cos(u_m)\cos(I_m)\sin(\Omega_m)\sin(\Omega_d)\cos(I_d)\sin(u_d) \\
&-\sin(u_m)\sin(\Omega_m)\sin(\Omega_d)\cos(u_d) - \sin(u_m)\sin(\Omega_m)\cos(\Omega_d)\cos(I_d)\sin(u_d) \\
&+\cos(u_m)\cos(I_m)\cos(\Omega_m)\sin(\Omega_d)\cos(u_d) + \cos(u_m)\cos(I_m)\cos(\Omega_m)\cos(\Omega_d)\cos(I_d)\sin(u_d) \\
&+\cos(u_m)\sin(I_m)\sin(I_d)\sin(u_d)] \times R_d
\end{aligned}
$$

$$
\begin{aligned}
z(t) = [&\sin(I_m)\sin(\Omega_m)\cos(\Omega_d)\cos(u_d) - \sin(I_m)\sin(\Omega_m)\sin(\Omega_d)\cos(I_d)\sin(u_d) \\
&-\sin(I_m)\cos(\Omega_m)\sin(\Omega_d)\cos(u_d) - \sin(I_m)\cos(\Omega_m)\cos(\Omega_d)\cos(I_d)\sin(u_d) \\
&+\sin(I_d)\sin(u_d)\cos(I_m)] \times R_d
\end{aligned}
$$

…….. (2. 6)

It uses variation of parameters method to incorporate perturbation forces affect. It can be used both for circular and for elliptical orbits. It gives singularities for equatorial orbits (I = 0 or 180 deg). Balaji's model does not discuss periodic variations during the time period and only gives mean variations in orbital parameters.

2.5 A Comparison of Models

Hill's equations are linearized equations and only give good results for closely placed circular or near circular orbits over short time period of simulation. The Earth was assumed to be spherical in its derivation. Although some authors have suggested some methods to include the affect of earth oblateness in Hill's equations, they are quite complicated to include. Hill's equations do not show the air drag effect on formation flying drift. The close form solution is valid only for frozen orbits (called Hill's orbit), which satisfy the condition $\dot{y}_0 = -2nx_0$.

The COWPOKE model is again valid only for closely placed formation flying. It is defined in terms of difference in orbital elements in an earth centered spherical frame of reference. For higher eccentric orbits, the reference orbit will deviate from the sphere and hence the results will not be accurate. Also for higher eccentric orbits, the truncation error in Bessel Fourier series for true anomaly will be increased. It does not give absolute state vectors for master or deputy satellites. It uses mean orbital elements for its simulation and shows only secular drift, hence unable to show the periodic variations. The variation of parameter method is suggested to include perturbation affects, which cannot handle high thrust problem. Also

the $\frac{dM}{dt}$ equation in variation of parameters method was derived for elliptical orbits and will give errors for formation in circular orbits.

The geometric approach method developed by Balaji in his PhD thesis gives non linear analysis for both circular and elliptical orbits, but it requires high computations and large programming. It has large expressions and is more prone to human programming error. It uses mean orbital elements in its simulation. The $\frac{dM}{dt}$ equation in variation of parameters method was derived for elliptical orbits and will give errors for formation in circular orbits. It breaks down at $e = 1$ or $e = 0$.

Chapter 3

Formation Flying Modeling and Simulation

A- Satellite Formation Flying Modeling

3.1 Introduction:

The formation flying models discussed in the last chapter have their own limitations. Some are defined by using orbital elements, some are defined in Cartesian coordinates, some uses differences between orbital elements. The assumptions in their derivation affect their accuracy. These models cannot handle high thrust problems. Some are valid only for circular orbits. Some of them give curvilinear separation. It is difficult to separate deputy and master dynamics in these models, which make their use in real time hardware in the loop simulation difficult. So another real time formation flying model and algorithm is developed here.

3.2 Real Time Formation Flying Model:

Consider at time t a master satellite that has position $(X_m(t), Y_m(t), Z_m(t))$ and velocity $(V_{xm}(t), V_{ym}(t), V_{zm}(t))$, and a deputy satellite that has position $(X_d(t), Y_d(t), Z_d(t))$ and velocity $(V_{xd}(t), V_{yd}(t), V_{zd}(t))$ in an inertial geocentric frame (IJK frame) with the origin at the centre of the earth. At this time the master has orbital elements $e_m(t)$, $a_m(t)$, $\Omega_m(t)$, $\omega_m(t)$ $\theta_m(t)$, $i_m(t)$ and the deputy satellite orbital elements are $\Omega_d(t)$, $\omega_d(t)$, $\theta_d(t)$, $i_d(t)$, $a_d(t)$ $e_d(t)$

A representative non inertial coordinate system RSW is defined for formation geometry which is moving with the master frame of reference, where radial separation is along radial vector (R-axis), along track separation is in the direction of motion (S-axis) 90 deg to the radial vector, cross track separation is perpendicular to the plane (W-axis) containing R and S axis to complete a rectangular coordinate system as shown in figure 3.1. $x_r(t)$, $y_r(t)$ and $z_r(t)$ are the radial, along track and cross track separations. This coordinate system, and relative distances in this coordinate system are shown in figure 3.1 and 3.2. Another sets of

coordinate systems $OPQR_m$, and $OPQR_d$ fixed with master and deputy satellites with origin at centre of earth are defined as. P-axis is in the direction of radius vector from centre of earth to centre of mass of satellites, Q axis in the direction of velocity and R axis is at right angle to plane containing P and Q axis. In order to calculate the deputy's relative position in formation geometry with respect to master satellite, transform deputy coordinate system $OPQR_d$ to RSW frame of reference by using Euler transformation matrices and a translation. For simplicity we are not mentioning here the time dependency of the parameters, although they are time dependent.

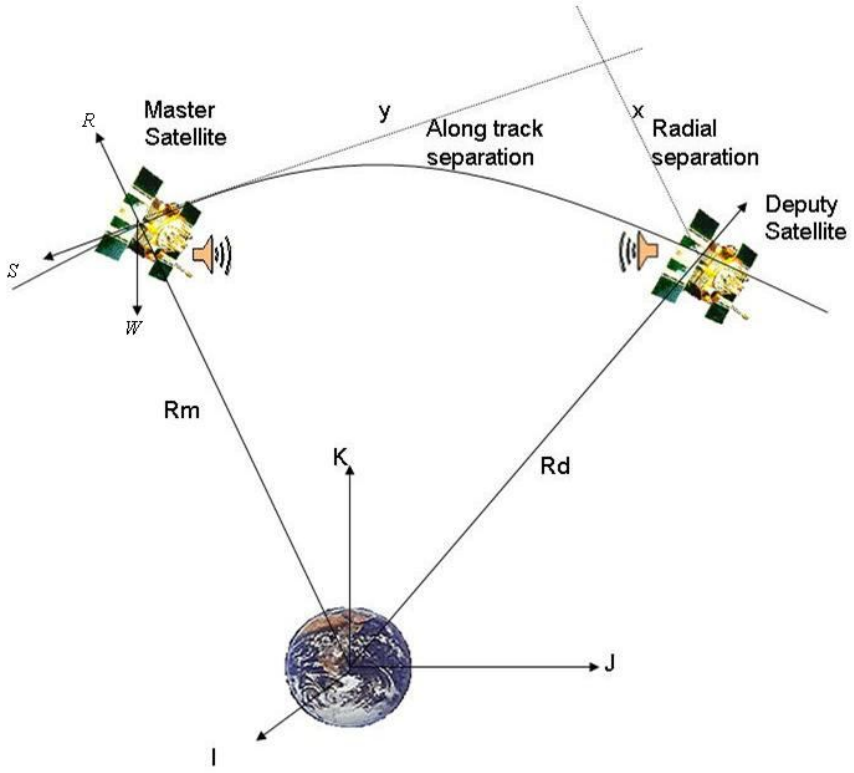


Figure 3.1: Relative geometry

XYZ axes can be transformed to *RSW frame* by 3 successive rotations as follows:

1. Rotation about the *Z* axis by $+\Omega$
2. Rotation about the *X* axis by $+ i$
3. Rotation about the Z axis by $+ u$ ($u = \omega + \theta$)

Where θ representing the true anomaly, Ω representing right ascension of ascending node (RAAN), ω representing argument of perigee, *u* representing the argument of latitude and *i* the inclination

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} \cos u_m & \sin u_m & 0 \\ -\sin u_m & \cos u_m & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i_m & \sin i_m \\ 0 & -\sin i_m & \cos i_m \end{bmatrix} \begin{bmatrix} \cos \Omega_m & \sin \Omega_m & 0 \\ -\sin \Omega_m & \cos \Omega_m & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_d \\ Y_d \\ Z_d \end{bmatrix}_g - \begin{bmatrix} R_m \\ 0 \\ 0 \end{bmatrix}$$

…… (3. 1)

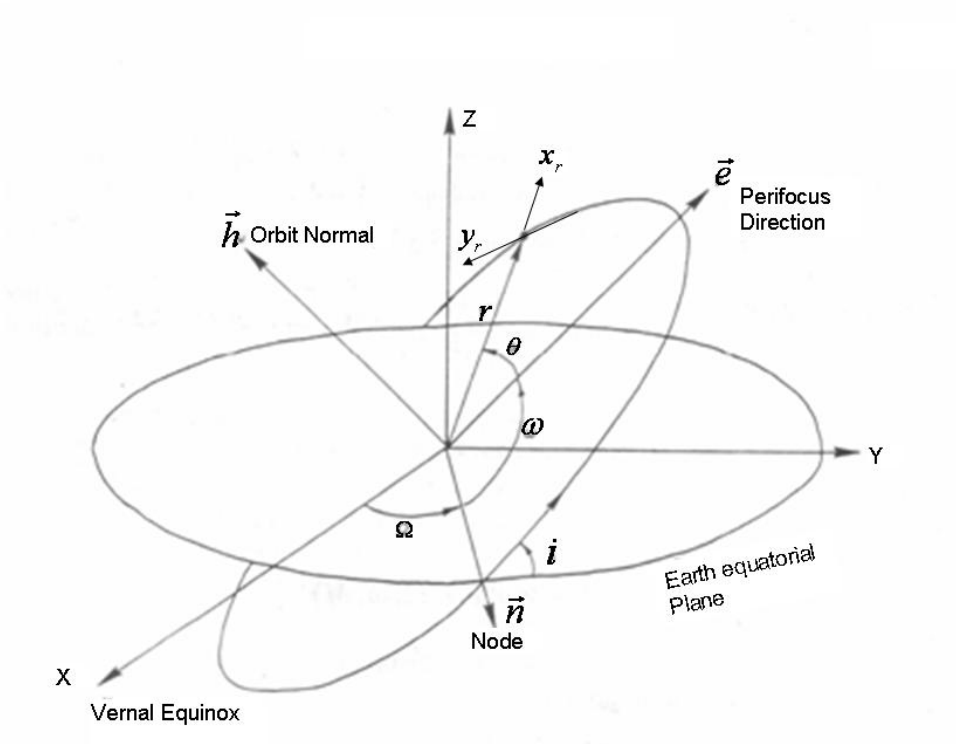


Figure 3.2: Orbital Frame and relative Geometry

Where

R_m = Radial distance from centre of earth to the master satellite

X_d, Y_d, Z_d are deputy satellite's coordinates in geocentric frame of reference and is given by the relation

$$\begin{bmatrix} X_d \\ Y_d \\ Z_d \end{bmatrix}_g = \begin{bmatrix} C\Omega_d & -S\Omega_d & 0 \\ S\Omega_d & C\Omega_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & Ci_d & -Si_d \\ 0 & Si_d & Ci_d \end{bmatrix} \begin{bmatrix} Cu_d & -Su_d & 0 \\ Su_d & Cu_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_d \\ 0 \\ 0 \end{bmatrix}_{OPQRd}$$

..... (3. 2)

Here C denotes Cosine and S denotes sine functions. Multiplication of the matrices in Equation 3.1 gives non linear relative separation expressions in terms of sine and cosine of Euler angles.

$$x_r(t) = [(\cos u_m \cos\Omega_m - \sin u_m \cos i_m \sin\Omega_m) X_d(t)$$
$$+ (\cos u_m \sin\Omega_m + \sin u_m \cos i_m \cos\Omega_m) Y_d(t) + \sin u_m \sin i_m . Z_d(t)] - R_m(t) \ldots\ldots\ldots\ldots \quad (3.3)$$

$$y_r(t) = (-\sin u_m \cos\Omega_m - \cos u_m \cos i_m \sin\Omega_m) . X_d(t)$$
$$+ (-\sin u_m \sin\Omega_m + \cos u_m \cos i_m \cos\Omega_m) . Y_d(t) + \cos u_m \sin i_m . Z_d(t) \ldots\ldots\ldots\ldots \quad (3.4)$$

$$z_r(t) = \sin i_m \sin\Omega_m . X_d(t) - \sin i_m \cos\Omega_m . Y_d(t) + \cos i_m . Z_d(t) \ldots\ldots\ldots\ldots \quad (3.5)$$

To simplify further, three fundamental vectors $\vec{h}, \vec{n}$ and $\vec{e}$ are used. As shown by the geometry of the orbit in geocentric frame in Figure (3.3) these vectors are defined in the following sections.

3.3 Specific Angular Momentum:

The vector $\vec{h}$ is a specific angular momentum vector. It is the cross product of the position and velocity vector.

$$\vec{h} = \vec{r} \times \vec{v}$$

It is always perpendicular to the plane containing $\vec{r}$ and $\vec{v}$ vectors i.e perpendicular to orbital plane.

Let

$$\vec{r} = [x.\vec{i} + y.\vec{j} + z.\vec{k}]$$
$$\vec{v} = [v_x.\vec{i} + v_y.\vec{j} + v_z.\vec{k}]$$

are position and velocity vectors in Geocentric frame, then the components of angular momentum vector in Geocentric frame are

$$h_x = (y.v_z - z.v_y)$$

$$h_y = (z.v_x - x.v_z) \quad \text{..... (3. 6)}$$

$$h_z = (x.v_y - y.v_x)$$

For an unperturbed orbit, the angular momentum vectors remain constant. Due to the oblate earth perturbation the angular momentum vector precesses around the spin axis of the earth.

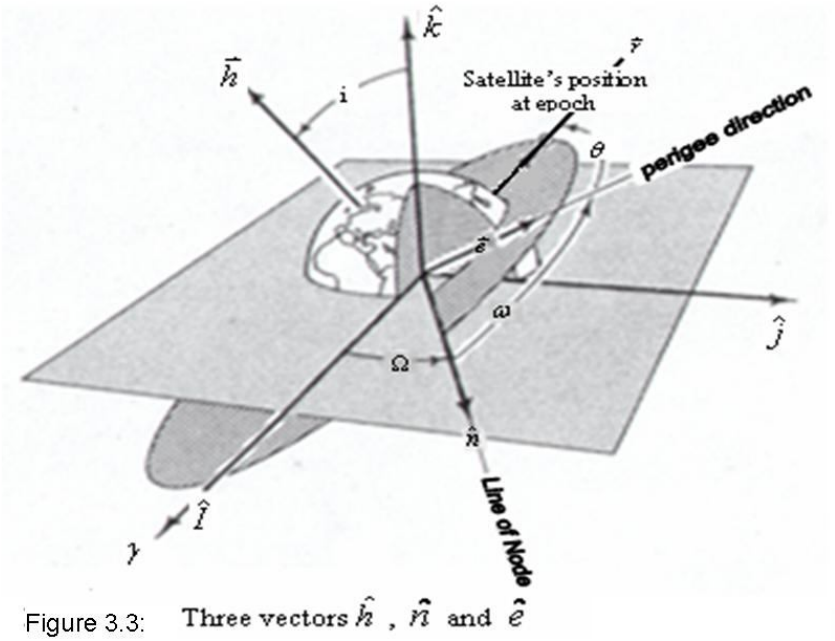


Figure 3.3: Three vectors $\hat{h}$, $\hat{n}$ and $\hat{e}$

3.4 The Eccentricity Vector:

The eccentricity vector points towards perigee from the centre of the earth with a magnitude exactly equal to the eccentricity of orbital plane. It is a constant vector. In case of the oblate earth perturbation the average rate of change in eccentric vector is zero. The eccentricity vector is given by Chobotove [32] as

$$\vec{e} = \frac{1}{\mu}(v^2 - \frac{\mu}{r})\vec{r} - (\vec{r}.\vec{v})\vec{v} \quad \text{..... (3. 7)}$$

3.5 The Node Vector:

The node vector points from centre of earth to the ascending node. This vector lies in equatorial plane. This vector remains constant in case of unperturbed orbit. Due to oblate

earth perturbation this vector rotates around earth axis at a rate which depends upon altitude and inclination.

The node vector $\vec{n}$ is defined as

$$\vec{n} = \vec{k} \times \vec{h} \quad \text{..... (3. 8)}$$

Thus

$$= \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 0 & 0 & 1 \\ h_x & h_y & h_z \end{vmatrix}$$

$$\vec{n} = -h_y.\vec{i} + h_x.\vec{j} \quad \text{.....(3. 9)}$$

From the definition, the $\vec{n}$ must be perpendicular to both $\vec{k}$ and $\vec{h}$. To be perpendicular to $\vec{k}$, the $\vec{n}$ must lie in equatorial plane and to be perpendicular to $\vec{h}$, the $\vec{n}$ would have to lie in orbital plane. Therefore $\vec{n}$ must lie both in equatorial and orbital plane, and hence will lie at the intersection of both planes. Now consider the figure 3.4 to find the sin and cosine of orbital element with the help of three fundamental elements already defined. The inclination angle $\vec{i}$ is the angle between unit vector $\vec{k}$ and angular momentum vector $\vec{h}$. The dot product of two vectors given by Chobotove and Bate [32, 33].

$$\cos i(t) = \frac{\vec{h}(t).\vec{k}}{\left|\vec{h}(t)\right|}$$

$$\cos(i(t)) = \frac{h_z(t)}{\left|h(t)\right|} \quad \text{.....(3. 10)}$$

The angle between $\vec{h}$ and $h_x\vec{i} + h_y\vec{j}$ is $(90^0 - i)$. The dot product of two vectors will give sine of inclination angle i

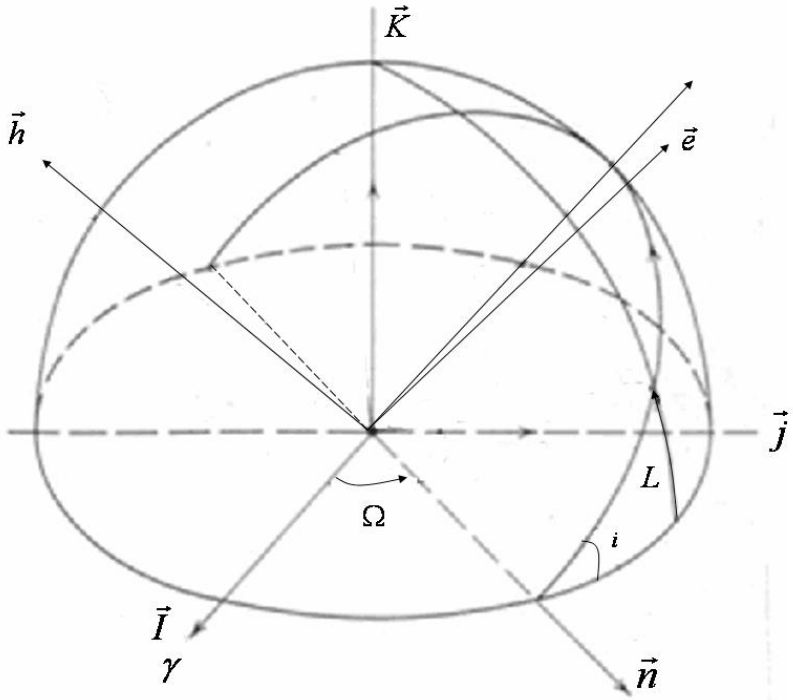


Figure 3.4: Orbital Geometry

$$\cos(90 - i(t)) = \frac{(h_x\vec{i} + h_y\vec{j}).(h_x\vec{i} + h_y\vec{j} + h_z\vec{k})}{(\sqrt{h^2{}_x + h^2{}_y})(\sqrt{h^2{}_x + h^2{}_y + h^2{}_z})}$$

$$\sin i(t) = \frac{\sqrt{h^2{}_x + h^2{}_y}}{|\vec{h}|} \quad \text{.....(3. 11)}$$

Right ascension of ascending node is a angle between $\vec{i}$ vector and nodal vector $\vec{n}$. Dot product of these two vectors will give

$$\cos\Omega(t) = \frac{\vec{i}.\vec{n}}{|n|}$$

$$\cos(\Omega(t)) = -\frac{h_y(t)}{\sqrt{h_x{}^2 + h_y{}^2}} \quad \text{.....(3. 12)}$$

Through vectors multiplication (Chobotov, [32]),

$$\sin\Omega(t) = (\vec{i} \times \frac{\vec{n}}{|n|}).\vec{k}$$

After simplification, it comes

$$\sin(\Omega(t)) = \frac{h_x(t)}{\sqrt{h_{x^x}{}^2 + h_y{}^2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3. 13)}$$

The angle between $\vec{r}$ and $\vec{n}$ vectors is $(\omega + \theta)$ and dot product of these vectors will give cosine of the angles between them.

$$\cos(\omega + \theta) = \cos u_m = \frac{\vec{n}.\vec{r}}{|\vec{n}||\vec{r}|}$$

Substitute the expressions for $\vec{n}$, $\vec{r}$, h_x and h_y give following expression

$$\cos(\omega + \theta) = \cos u_m = \frac{x\cos\Omega(t) + y\sin\Omega(t)}{r(t)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \text{(3. 14)}$$

The $\sin(\omega + \theta)$ can be found by vector multiplication of $\vec{n}$, $\vec{r}$ and $\vec{h}$ as

$$\sin(\omega + \theta) = \sin u_m = \frac{\vec{n} \times \vec{r}}{|\vec{n}||\vec{r}|} . \frac{\vec{h}}{|\vec{h}|}$$

After simplification,

$$\sin(\omega + \theta) = \sin u_m = \frac{Z(t)}{r(t)\sin i(t)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3. 15)}$$

and

$$\cos\omega = \frac{\cos\Omega.e_x + \sin\Omega.e_y}{|e|}$$

Where e_x and e_y are components of eccentricity vector. To simplify expressions in the derivation, the following terms are defined

$$\left.\begin{aligned} \vec{r}_{xy} &= X.\vec{i} + Y.\vec{j}\ , \\ \vec{r}_{yz} &= Y.\vec{j} + Z.\vec{k} \\ \vec{r}_{xz} &= X.\vec{i} + Z.\vec{k} \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3. 16)}$$

$$\left.\begin{aligned} \vec{V}_{xy} &= V_x\vec{i} + V_y.\vec{j} \\ \vec{V}_{yz} &= V_y\vec{j} + V_z.\vec{k} \\ \vec{V}_{xz} &= V_x\vec{i} + V_z.\vec{k} \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3. 17)}$$

Radial Separation derivation:

Substitute sine and cosine of Euler angles from equation 3.10 to 3.15 in equation 3.3. The 1st term of equation 3.3 comes out as

$$x1(t) = [(\cos u_m \cos \Omega_m - \sin u_m \cos i_m \sin \Omega_m) X_d(t)$$

$$= \left(\frac{-xh_y + yh_x}{r\sqrt{h^2{}_x + h^2{}_y}}\right)\left(\frac{-h_y}{\sqrt{h^2{}_x + h^2{}_y}}\right) - \left(\frac{Z.h}{r\sqrt{h^2{}_x + h^2{}_y}}\right).\frac{h_z}{h}.\frac{h_x}{\sqrt{h^2{}_x + h^2{}_y}}$$

$$= \frac{xh^2{}_y - y.h_x.h_y - z.h_x.h_y}{r.(h^2{}_x + h^2{}_y)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.18)$$

Consider the numerator term of the above equation and substitute the values of h_x, h_y, and h_z from equation 3.6.

$$= X.Z^2.V^2{}_x + X^3.V^2{}_Z - 2X^2.Z.V_x.V_Z - Y^2.Z.V_x.V_Z + X.Y^2V^2{}_Z + Y.Z^2.V_x.V_Y - X.Y.Z.V_Y.V_Z$$
$$- X.Y.Z.V_Y.V_Z + Z.Y^2.V_x.V_Z + X.Z^2.V^2{}_Y - Y.Z^2.V_x.V$$

Rearrange the terms

$$= X.Z^2(V^2{}_x + V^2{}_Y) + X.V^2{}_Z(X^2 + Y^2) - 2X.Z.V_Z(X.V_xY.V_Y) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.19)$$

Now $\vec{V}_{xy} = V_x\vec{i} + V_y.\vec{j}$ and $r_{xy} = \vec{X}i + \vec{Y}j$, then

$$\vec{V}_{xy} \cdot \vec{V}_{xy} = V^2{}_{xy} = V^2{}_x + V^2{}_y \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.20)$$

$$r^2{}_{xy} = \vec{r}_{xy} \cdot \vec{r}_{xy} = X^2 + Y^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.21)$$

$$\vec{r}_{xy} \cdot \vec{V}_{xy} = (X.V_x + Y.V_y) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.22)$$

Expression 3.19 can be written as

$$= X.Z^2(\vec{V}_{xy} \cdot \vec{V}_{xy}) + X.V^2{}_Z(\vec{r}_{xy} \cdot \vec{r}_{xy}) - 2X.Z.V_Z(\vec{r}_{xy} \cdot \vec{V}_{xy})$$

$$= X.(Z.\vec{V}_{XY} - V_Z.\vec{r}_{xy})^2.X_d(t)$$

Hence 1st term of equation 3.3 is

$$= \frac{X.(Z.\vec{V}_{XY} - V_Z.\vec{r}_{xy})^2}{r.(h^2{}_x + h^2{}_y)}.[X_d]_g \quad \text{..... (3.23)}$$

Similarly 2nd and 3rd terms of equation 3.3 comes out as

$$= \frac{Y.(V_z.\vec{r}_{xy} - Z.V_{xy})^2}{r.(h^2{}_x + h^2{}_y)}.[Y_d]_g \quad \text{..... (3. 24)}$$

$$= \frac{Z(V_z\vec{r}_{xy} - Z\vec{V}_{xy})}{r.(h_x{}^2 + h_y{}^2)}.[Z_d]_g \quad \text{..... (3.25)}$$

Combine together the three expressions (3.23), (3.24), and (3.25) for $x_r(t)$

$$x_r = \frac{(V_z\vec{r}_{xy} - Z\vec{V}_{xy})^2}{r.(h^2{}_x + h^2{}_y)}(X.[X_d]_g + Y.[Y_d]_g + Z.[Z_d]_g) - \vec{r} \quad \text{..... (3. 26)}$$

Consider the term $h^2{}_x + h^2{}_y$ in above equation and substitute the values of h_x and h_y from equation 3.6

$$(h_x{}^2 + h_y{}^2) = (Y.V_z - Z.V_y)^2 + (Z.V_x - X.V_z)^2$$

Rearranging the terms, we get

$$= V^2{}_z(X^2 + Y^2) + Z^2.(V^2{}_x + V^2{}_y) - 2.Z.V_z(Y.V_y + X.V_x)$$

Use the vector definitions given in equations 3.20, 3.21, 3.22

$$= (\mathrm{V}_z{}^2.(r_{xy} \cdot r_{xy}) + Z^2.(V_{xy} \cdot V_{xy}) - 2.Z.V_z.\vec{r}_{xy}.\vec{V}_{xy})$$

$$(h_x{}^2 + h_y{}^2) = (V_z\vec{r}_{xy} - Z\vec{V}_{xy}) \quad \text{..... (3. 27)}$$

Equation 3.26, then becomes

$$x_r = (\frac{X.[X_d]_g + Y.[Y_d]_g + Z.[Z_d]_g}{|r|}) - \vec{r}$$

$$x_r = \frac{\vec{r}}{r} \cdot \vec{r}_d - r \quad \text{.....} \quad (3.28)$$

Along Track Separation derivation:

Now substitute the sine and cosine of Euler angles, h_x, h_y and h_z in along track expression in equation3.4 . The 1st term comes out as

$$= \left(\frac{Z.h^2.h_y + X.h_x.h_y.h_z - Y.h^2{}_x.h_z}{r.h.(h^2{}_x + h^2{}_y)}\right).X_d \quad \text{.....} \quad (3.29)$$

Consider nominator term and substitute the expressions for h_x, h_y and h_z,

$$= [Z(Y.V_z - z.V_Y)^2 + (Z.V_x - X.V_z)^2 + (X.V_y - Y.V_x)^2].(Z.V_x - X.V_z)$$

$$+ X.(Z.V_x - X.V_z)(X.V_y - Y.V_x)(Y.V_z - ZV_y) - Y.(Y.V_z - Z.V_y)^2(X.V_y - Y.V_x)$$

Expand squares and rearrange the terms

$$= [Z^2.V_x - XZ.V_z - XY.V_y + Y^2.V_x].[Z^2(V^2{}_x + V^2{}_y) + V_z{}^2(X^2 + Y^2)$$

$$- 2Z.V_z(X.V_x + Y.V_y)]$$

Now use the vector definitions given equations 3.20, 3.21, 3.22 and rearrange the terms.

$$= [Z^2.V_x - XZ.V_z - XY.V_y + Y^2.V_x].[Z^2(\vec{V}_{xy} \cdot \vec{V}_{xy}) + V_z{}^2(\vec{r}_{xy} \cdot \vec{r}_{xy}) - 2ZV_z(.\vec{r}_{xy}.\vec{V}_{xy})]$$

$$= \vec{r}_{yz}(V_x.\vec{r}_{yz} - XV_{yz})(Z\vec{V}_{xy} - V_z\vec{r}_{xy})^2$$

Put this back into expression 3.29, will give 1st term as

$$= \left(\frac{\vec{r}_{yz}(V_x.\vec{r}_{yz} - XV_{yz})(Z\vec{V}_{xy} - V_z\vec{r}_{xy})^2}{r.h.(h^2{}_x + h^2{}_y)}\right).X_d \quad \text{.....} \quad (3.30)$$

Similarly 2nd term of equation 3.4 become

$$= -ZY^3V^3{}_z - YZ^3V^2{}_y.V_z + 2Y^2Z^2V_yV^2{}_z - YZ^3V^2{}_xV_z - X^2YZVV^3{}_z + 2XYZ^2V_xV^2{}_z$$
$$- X^2YZV^2{}_yV_z - Y^3ZV^2{}_xV_z + 2XY^2ZV_xV_yV_z + Y^2Z^2V_yV^2{}_z + Z^4V^3{}_y - 2YZ^3V^2{}_yV_z$$
$$+ Z^4V^2{}_xV_y + X^2Z^2V_yV^2{}_z - 2XZ^3V_xV_yV_z + X^2Z^2V^3{}_y + Y^2Z^2V^2{}_xV_y - 2XYZ^2V_xV^2{}_y$$
$$+ Y^3ZV^2{}_xV_zX^2Y^2V_yV^2{}_z - XY^3V_xV^2{}_z + XYZ^2V_xV^2{}_y - Y^2Z^2V^2{}_xV_y - X^2YZV^2{}_YV_z$$
$$+ X^2Z^2V^2{}_xV_y + X^4V_yV^2{}_z - 2X^3ZV_xV_yV_z - XYZ^2V^3{}_X - X_3YV_xV^2{}_z + 2X^2YZV^2{}_xV_z$$

Rearrange the terms

$$= V^2{}_z(X^2 + Y^2)[-ZYV_z + Z^2V_y + X^2V_y - XYV_x]$$
$$+ Z^2(V^2{}_x + V^2{}_y)[-YZV_z + Z^2V_y + X^2V_y - XYV_x]$$
$$+ 2ZV_z(XV_x + YV_y)[YZV_z + XYV_x - X^2V_y - Z^2V_y]$$

Rearrange the terms again,

$$= V_y(X^2 + Z^2) - Y(XV_x + ZV_z)[V^2{}_z(X^2 + Y^2) + Z^2(V^2{}_x + V^2{}_y) - 2ZV_z(XV_x + YV_y)]$$

Now use the vector definitions given equations 3.20, 3.21, 3.22 and rearrange the terms

$$= \left(\frac{\vec{r}_{xz}(\vec{r}_{xz}V_y - Y\vec{V}_{xz})(V_z\vec{r}_{xy} - Z\vec{V}_{xy})^2}{r.h(h^2{}_x + h^2{}_y)}\right).Y_d \quad \text{..... (3. 31)}$$

And 3rd term of equation 3.4 conclude to

$$= \frac{\vec{\mathrm{r}}_{\mathrm{xy}}(V_z\vec{\mathrm{r}}_{\mathrm{xy}} - Z.\vec{V}_{xy})(V_z.\vec{\mathrm{r}}_{\mathrm{xy}} - Z.\vec{V}_{xy})^2}{r.h.(h^2{}_x + h^2{}_y)}.Z_d \quad \text{..... (3. 32)}$$

Combining above three equations 3.30, 3.31, 3.32, gives

$$y_r = \frac{(V_z.\vec{\mathrm{r}}_{\mathrm{xy}} - Z.\vec{V}_{xy})^2}{r.h.(h^2{}_x + h^2{}_y)}[\vec{r}_{yz}(V_x.\vec{r}_{yz} - XV_{yz}).X_d + \vec{r}_{xz}(\vec{r}_{xz}V_y - Y\vec{V}_{xz}).Y_d + \vec{\mathrm{r}}_{\mathrm{xy}}(V_z\vec{\mathrm{r}}_{\mathrm{xy}} - Z.\vec{V}_{xy}).Z_d]$$

..... (3. 33)

Now from expression 3.27, $h^2{}_x + h^2{}_y = (V_z\vec{r}_{xy} - Z\vec{V}_{xy})^2$, above Equation then becomes

$$= \frac{1}{r.h}[\vec{r}_{yz}(V_x.\vec{r}_{yz} - X\vec{V}_{yz}) \cdot X_d + \vec{r}_{xz}(\vec{r}_{xz}V_y - Y\vec{V}_{xz}) \cdot Y_d + \vec{r}_{xy}(V_z\vec{r}_{xy} - Z.\vec{V}_{xy}) \cdot Z_d]$$

Replace r_{xy}, r_{yz}, r_{xz} from equations 3.16 and expand dot product, it gives

$$y_r = \frac{1}{r.h}[(Zh_y - Yh_z).X_d + (Xh_z - Zh_x).Y_d + (Yh_x - Xh_y).Z_d]$$

In vector form it can be written as

$$y_r = \frac{1}{r.h}[(\vec{h} \times \vec{r}) \cdot \vec{r}_d]$$

$$= (\frac{\vec{h}}{h} \times \frac{\vec{r}}{r}) \cdot \vec{r}_d \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (3.\ 34)$$

Cross Track Separation derivation:

Now substitute the sine and cosine of Euler angles, h_x, h_y and h_z in cross track expression in equation 3.5 It gives

$$z_r = \frac{\sqrt{h^2{}_x + h^2{}_y}}{h}.\frac{h_x}{\sqrt{h^2{}_x + h^2{}_y}}X_d - \frac{\sqrt{h^2{}_x + h^2{}_y}}{h}.\frac{-h_y}{\sqrt{h^2{}_x + h^2{}_y}}.Y_d + \frac{h_z}{h}.Z_d$$

$$z_r = \frac{h_x}{h}.X_d + \frac{h_y}{h}..Y_d + \frac{h_z}{h}.Z_d$$

And in vector form

$$z_r = \frac{\vec{h}}{h} \cdot \vec{r}_d \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (3.\ 35)$$

Hence radial, along track and cross track separation in master's local frame of reference is.

$$x_r(t) = \frac{\vec{r}_m(t)}{r_m(t)} \cdot \vec{r}_d(t) - r_m(t)$$

$$y_r(t) = \left(\frac{\vec{h}_m(t)}{h_m(t)} \times \frac{\vec{r}_m(t)}{r_m(t)}\right) \cdot \vec{r}_d(t)$$

$$z_r(t) = \frac{\vec{h}_m(t)}{h_m(t)} \cdot \vec{r}_d(t)$$

…… (3. 36)

Therefore the relative position of the deputy satellite in the master frame of reference can be calculated from three vectors i.e. $\frac{\vec{h}_m}{h_m}$, $\frac{\vec{r}_m}{r_m}$, and $\vec{r}_d$. Where $\frac{\vec{h}_m}{h_m}$ is a unit vector in the direction of the angular momentum vector and is perpendicular to orbital plane. The unit vector $\frac{\vec{r}_m}{r_m}$ points along the radius vector its dot product with $\vec{r}_d$ will give relative separation along $R-axis$. The vector $\frac{\vec{h}_m}{h_m} \times \frac{\vec{r}_m}{r_m} = \hat{a}_y$ is perpendicular to both $\vec{h}_m$ and $\vec{r}_m$ vectors and is in the direction of $S-axis$. The dot product of $\hat{a}_y$ with $\vec{r}_d$ will give relative distance along $S-axis$. Similarly vector $\frac{\vec{h}_m}{h_m}$ is along the direction of $W-axis$ and its dot product with $\vec{r}_d$ will give relative separation along $W-axis$.

As the coordinate transformation (for inertial coordinates systems) does not change the magnitude and direction of the vectors, so these vector remain the same whether they are calculated in a geocentric frame, perifocal coordinate frame, spherical coordinate frame or a cylindrical coordinate system with their origin at centre of earth. Hence we can use this model in any above mentioned coordinate system with the origin at the centre of the earth provided all the vectors are calculated in the same coordinate system. Here an example is presented to show the use of above model in a perifocal coordinate system for relative position calculation.

3.6 Formation Flying Difference Model:

The above developed real time formation flying model consist of three unit vectors and these unit vectors remain the same in any inertial frame of reference which has the origin at the centre of earth. To see a general application of the vector model let us take a case with the master and deputy satellite in two different orbital frames as shown in figure 3.5.

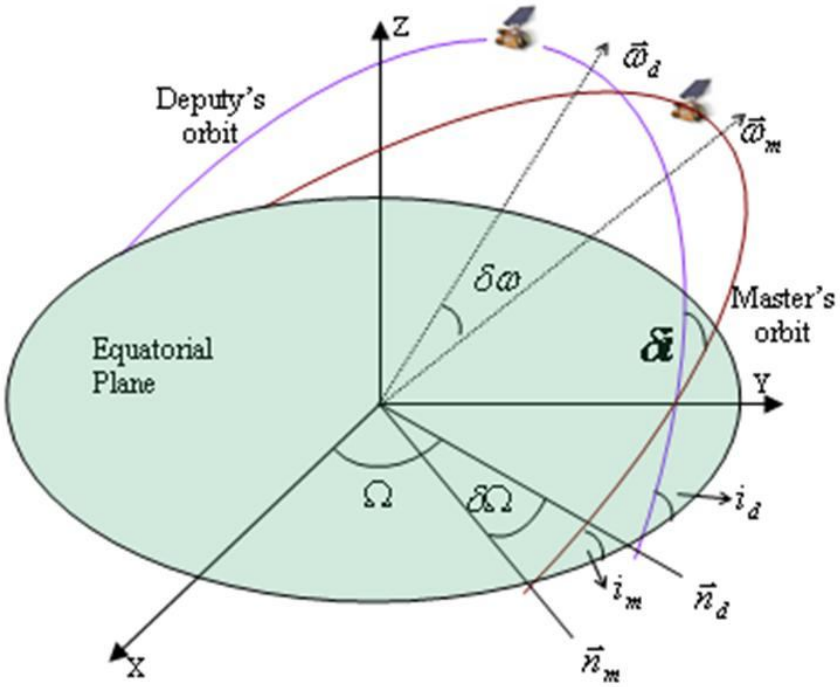


Figure 3.5: : Orbital Geometry for Difference Model

The master's orbital elements are $e_m(t), a_m(t), \Omega_m(t), \omega_m(t)\ \ \theta_m(t), i_m(t)$ and deputy satellite's orbital elements are $\Omega_d(t), \omega_d(t), \theta_d(t), i_d(t),\ \ a_d(t)\ \ e_d(t)$ and the difference between these orbital elements is

$\delta\Omega = \Omega_d - \Omega_m,\ \delta\omega = \omega_d - \omega_m, \delta\theta = \theta_d - \theta_m$

$\delta.i = I_d - I_m, \delta a = a_d - a_m, \delta.e = e_d - e_m$

Let at any time t the position and velocity vectors for master and deputy satellites in their own perifocal coordinate systems are

$$\vec{r}_m = r_m \cos\theta_m \hat{m}_x + r_m \sin\theta_m \hat{m}_y$$

$$\vec{v}_m = \frac{\mu}{P_m}\left(-\sin\theta_m.\hat{m}_x + (\cos\theta_m + e_m).\hat{m}_y, 0\right) \quad \text{..... (3. 37)}$$

$$\vec{r}_d = r_d \cos\theta_d.\hat{d}_x + r_d \sin\theta_d.\hat{d}_y$$

Where

$$P_m = a_m(1 - e_m{}^2)$$

Now

$$\vec{h}_m = \frac{\mu.r_m}{P_m}(1 + e_m \cos\theta_m)\hat{m}_z \quad \text{and}$$

$$|h_m| = \frac{\mu.r_m}{P_m}(1 + e_m \cos\theta_m) \quad \text{..... (3. 38)}$$

Where $(\hat{m}_x.\ \hat{m}_y,\ \hat{m}_z)$ and $(\hat{d}_x.\ \hat{d}_y,\ \hat{d}_z)$ are unit vectors along three coordinate axis of the master and deputy satellite's perifocal coordinate system. In order to use the equations 3.36, we need to find the deputy position vector in master's perifocal coordinate system which can be obtained through the following transformations

1. $(\delta\omega + \theta)$ Rotation around z-axis
2. i_d rotation around x-axis
3. $\delta\Omega$ -Rotation around z-axis
4. i_m -Reverse rotation around x-axis

$$[\vec{r}_{d]}]_m = \begin{bmatrix} i_m ... \text{Reverse} \\ rotation \\ around x - axis \end{bmatrix} \begin{bmatrix} \delta\Omega..Rotation \\ around \\ .z - axis \end{bmatrix} \begin{bmatrix} i_d ..Rotation \\ around \\ .x - axis \end{bmatrix} \begin{bmatrix} (\delta\omega + \theta)..Rotation \\ around \\ .z - axis \end{bmatrix}$$

After multiplications we get $[\vec{r}_d]_m$ i.e the deputy position in the master's orbital frame

$$[\vec{r}_{d]}]_m = \begin{bmatrix} [\cos\delta\Omega\cos(\delta\omega+\theta_d)-\sin\delta\Omega\cos i_d\sin(\delta\omega+\theta_d)]r_d \\ [\cos i_m\sin\delta\Omega\cos(\delta\omega+\theta_d)+\cos i_m\cos i_d\cos\delta\Omega\sin(\delta\omega+\theta_d)+\sin i_m\sin i_d\sin(\delta\omega+\theta_d)]r_d \\ [-\sin i_m\sin\delta\Omega\cos(\delta\omega+\theta_d)-\sin i_m\cos\delta\Omega\cos i_d\sin(\delta\omega+\theta_d)+\cos i_m\sin i_d\sin(\delta\omega+\theta_d)]r_d \end{bmatrix}$$

..... (3. 39)

Once we know the position vector, we can calculate relative distances. Consider radial separation equation from 3.36

$$x_r(t) = \frac{\vec{r}_m(t)}{|r_m(t)|}\cdot\vec{r}_d(t) - |r_m(t)|$$

Now substitute the values in radial separation

$$x_r(t) = \frac{r_m(\cos\theta_m,\sin\theta_m,0)}{r_m}\cdot[\vec{r}_d]_m - r_m(t)$$

$$x_r(t) = (\cos\theta_m,\sin\theta_m,0)\cdot[\vec{r}_d]_m - r_m(t)$$

After simplification it gives

$$\begin{aligned} x_r(t) = {} & (\cos\theta_m\cos\delta\Omega\cos(\delta\omega+\theta_d)-\cos\theta_m\sin\delta\Omega\cos i_d\sin(\delta\omega+\theta_d) \\ & +\sin\theta_m\cos i_m\sin\delta\Omega\cos(\delta\omega+\theta_d)+\sin\theta_m\cos i_m\cos\delta\Omega\cos i_d\sin(\delta\omega+\theta_d) \\ & +\sin\theta_m\sin i_m\sin i_d\sin(\delta\omega+\theta_d))r_d - r_m \end{aligned}$$

..... (3. 40)

Now consider $Y_r(t)$ term, which is from equation (3.36) is

$$y_r(t) = (\frac{\vec{h}}{h}\times\frac{\vec{r}}{r})\cdot[\vec{r}_d]_m$$

..... (3. 41)

$$y_r(t) = [\hat{m}_z]\times[\cos\theta_m.\hat{m}_x + \sin\theta_m.\hat{m}_y]\cdot[\vec{r}_d]_m$$

..... (3. 42)

Substitute the values from (3.39), we get

$$\begin{aligned} y_r(t) = {} & [-\sin\theta_m\cos\delta\Omega\cos(\delta\omega+\theta_d)+\sin\theta_m\sin\delta\Omega\cos I_d\sin(\delta\omega+\theta_d) \\ & +\cos\theta_m\sin\delta\Omega\cos I_m\cos(\delta\omega+\theta_d)+\cos\theta_m\cos\delta\Omega\cos I_m\cos I_d\sin(\delta\omega+\theta_d) \\ & +\cos\theta_m\sin I_m\sin I_d\sin(\delta\omega+\theta_d)]r_d \end{aligned}$$

..... (3. 43)

Now consider expression for cross track separation, which is

$$Z_r = \frac{\vec{h}}{h} \cdot [\vec{r}_d]_m$$
$$Z_r = [\hat{e}_z] \cdot [\vec{r}_d]_m$$

Substitute the values from (3.39),

$$z_r = [-\sin I_m \sin \delta\Omega \cos(\delta\omega + \theta_d) - \sin I_m \cos \delta\Omega \cos I_d \sin(\delta\omega + \theta_d) + \cos I_m \sin I_d \sin(\delta\omega + \theta_d)] \cdot r_d \quad \text{..... (3. 44)}$$

Hence the satellite formation flying model in terms of orbital elements and their difference is

$$x_r(t) = (\cos\theta_m \cos\delta\Omega \cos(\delta\omega + \theta_d) - \cos\theta_m \sin\delta\Omega \cos i_d \sin(\delta\omega + \theta_d) + \sin\theta_m \cos i_m \sin\delta\Omega \cos(\delta\omega + \theta_d) + \sin\theta_m \cos i_m \cos\delta\Omega \cos i_d \sin(\delta\omega + \theta_d) + \sin\theta_m \sin i_m \sin i_d \sin(\delta\omega + \theta_d))r_d - r_m$$

$$y_r(t) = [-\sin\theta_m \cos\delta\Omega \cos(\delta\omega + \theta_d) + \sin\theta_m \sin\delta\Omega \cos I_d \sin(\delta\omega + \theta_d) + \cos\theta_m \sin\delta\Omega \cos I_m \cos(\delta\omega + \theta_d) + \cos\theta_m \cos\delta\Omega \cos I_m \cos I_d \sin(\delta\omega + \theta_d) + \cos\theta_m \sin I_m \sin I_d \sin(\delta\omega + \theta_d)]r_d$$

$$z_r = [-\sin I_m \sin \delta\Omega \cos(\delta\omega + \theta_d) - \sin I_m \cos \delta\Omega \cos I_d \sin(\delta\omega + \theta_d) + \cos I_m \sin I_d \sin(\delta\omega + \theta_d)] \cdot r_d$$

This is another satellite formation flying model that is equally good for circular and elliptical orbits. Let's call it as **Difference Model Long Form (DMLF)**. Variation of parameter method can be used to add the affect of gravitational perturbation and atmospheric drag. The inclusion of perturbation, simulation analysis and comparison with other SFF model is given in simulation section.

Let us suppose that satellites in the formation are closely spaced such that $\delta\Omega$ and $\delta.i$ are small enough so that

$\cos\delta\Omega \cong 1 \cong \cos\delta.i$,

As δi is very small, therefore $\cos i_m \approx \cos i_d$. Hence

$$\cos\theta_m \sin\delta\Omega \cos i_d \sin(\delta\omega + \theta_d) = \cos\theta_m \sin\delta\Omega \cos i_m \sin(\delta\omega + \theta_d) \quad \text{..... (3. 45)}$$

Apply these assumptions to the equation (3.40), combine 4th and 5th terms by applying trigonometric identity.

$$x_r(t) = (\cos\theta_m \cos\delta\Omega \cos(\delta\omega + \theta_d) - \cos\theta_m \sin\delta\Omega \cos i_d \sin(\delta\omega + \theta_d)$$
$$+ \sin\theta_m \cos i_m \sin\delta\Omega \cos(\delta\omega + \theta_d) + \sin\theta_m \sin(\delta\omega + \theta_d)\cos\delta I) r_d - r_m$$

Now apply $\cos\delta\Omega \cong 1 \cong \cos\delta.i$, combine 1st and 4th term by applying trigonometric identity will give

$$x_r(t) = (\cos(\delta\omega + \delta\theta_d) - \cos\theta_m \sin\delta\Omega \cos i_d \sin(\delta\omega + \theta_d)$$
$$+ \sin\theta_m \cos i_m \sin\delta\Omega \cos(\delta\omega + \theta_d)) r_d - r_m$$

Use assumption on 2nd and 3rd term and apply trigonometric identity, will give simplified form for radial separation

$$x_r(t) = (\cos(\delta\omega + \delta\theta) - \sin\delta\Omega \cos I_m \sin(\delta\omega + \delta\theta)) r_d - r_m \quad \text{..... (3. 46)}$$

Consider equation 3.43 and apply assumption from 3.45 to terms 4th, 5th and apply trigonometric identity, will give

$$y_r(t) = [-\sin\theta_m \cos\delta\Omega \cos(\delta\omega + \theta_d) + \sin\theta_m \sin\delta\Omega \cos I_d \sin(\delta\omega + \theta_d)$$
$$+ \cos\theta_m \sin\delta\Omega \cos I_m \cos(\delta\omega + \theta_d) + \cos\theta_m \cos\delta I \sin(\delta\omega + \theta_d)] r_d$$

Now $\cos\delta\Omega \cong 1 \cong \cos\delta.i$, apply trigonometric identity on 1st and 4th terms, will give

$$y_r(t) = [\sin(\delta\omega + \delta\theta) + \sin\theta_m \sin\delta\Omega \cos I_d \sin(\delta\omega + \theta_d)$$

$$+\cos\theta_m \sin\delta\Omega\cos I_m \cos(\delta\omega+\theta_d)]r_d$$

Now apply assumption on 2nd and 3rd term and use trigonometric identity will give simplified form of along track separation.

$$y_r(t)=[\sin(\delta\omega+\delta\theta)+\sin\delta\Omega\cos I_m\cos(\delta\omega+\delta\theta)]r_d \quad \text{..... (3. 47)}$$

Finally consider cross track equation 3.44 and apply assumptions from equation 3.45, gives

$$z_r=[\delta i\sin(\delta\omega+\theta_d)-\delta\Omega\sin I_m\cos(\delta\omega+\theta_d)]r_d \quad \text{..... (3. 48)}$$

Therefore the simplified form for satellite formation flying model for closely placed formation in terms of difference of orbital elements from equation 3.46, 3.47, 3.48 is

$$\left.\begin{aligned} x_r(t)&=(\cos(\delta\omega+\delta\theta)-\sin\delta\Omega\cos I_m\sin(\delta\omega+\delta\theta))r_d-r_m\\ y_r(t)&=[\sin(\delta\omega+\delta\theta)+\sin\delta\Omega\cos I_m\cos(\delta\omega+\delta\theta)]r_d\\ z_r&=[\delta i\sin(\delta\omega+\theta_d)-\delta\Omega\sin I_m\cos(\delta\omega+\theta_d)]r_d \end{aligned}\right\} \quad \text{..... (3. 49)}$$

The DMSF in equation 3.49 can be written as

$$\left.\begin{aligned} x_r(t)&=(\cos(\delta u)-\sin\delta\Omega\cos I_m\sin(\delta u))r_d-r_m\\ y_r(t)&=[\sin(\delta u)+\sin\delta\Omega\cos I_m\cos(\delta u)]r_d\\ z_r&=[\delta i\sin(\delta u+\theta_m)-\delta\Omega\sin I_m\cos(\delta u+\theta_m)]r_d \end{aligned}\right\}$$

Where $\delta u=u_d-u_m$ and u_m, u_d are argument of latitude and are defined as

$u_m=\omega_m+\theta_m$, and $u_d=\omega_d+\theta_d$

Let's call this simplified formation flying model as the **Difference Model Simplified Form (DMSF)**. Hence by using the definition of equation 3.36 of relative separation for perifocal coordinate system gave a very simple and unique model. Its performance and the comparison to other existing SFF models are presented in the next section.

Now equation 3.46 and 3.47 can be written as

$$\left[\frac{(x_r(t)+r_m)}{r_d}\right]^2 = \cos^2(\delta\omega+\delta\theta)+\sin^2\delta\Omega\cos^2 I_m \sin^2(\delta\omega+\delta\theta)$$
$$-2\cos(\delta\omega+\delta\theta)\sin\delta\Omega\cos I_m \sin(\delta\omega+\delta\theta) \quad \text{..... (3. 50)}$$

$$\left[\frac{y_r(t)}{r_d}\right]^2 = \sin^2(\delta\omega+\delta\theta)+\sin^2\delta\Omega\cos^2 I_m \cos^2(\delta\omega+\delta\theta)$$
$$+2\sin(\delta\omega+\delta\theta)\sin\delta\Omega\cos I_m \cos(\delta\omega+\delta\theta) \quad \text{..... (3. 51)}$$

Add above two equations

$$\left[\frac{(x_r(t)+r_m)}{r_d}\right]^2 + \left[\frac{y_r(t)}{r_d}\right]^2 = (1+\sin^2\delta\Omega\cos^2 I_m) \quad \text{..... (3. 52)}$$

Which is a equation of circle in RSW frame of reference with centre at $(-r_m,0)$ and radius $r_d\sqrt{(1+\sin^2\delta\Omega\cos^2 I_m)}$. It gives a very important identity about the formation flying. It gives the relation between radial separation and along track separation as a circle. This identity can also be used to verify the correctness of radial and along track separation behavior for any model. The centre point and radius of this circle are variable for elliptic orbit and remains fixed for circular orbits in equatorial plane or when $\delta\Omega = 0$. The circle defined by equation 3.52 and its geometrical representation in RSW frame of reference is shown in figure 3.6 for the case, when radius and centre of circle are constant.

Now from equation 3.52, the along track separation can be written explicitly in terms of radial separation, which is another novelty of this model and is given

$$y_r(t) = \sqrt{r_d^2(1+\sin^2 \delta\Omega \cos^2 I_m) - (x_r + r_m)^2} \quad \text{..... 3. 53}$$

Where r_m and r_d are in geocentric inertial frame and x_r is radial separation in formation flying frame (RSW). As $(\delta\omega + \delta\theta)$ is positive the root of equation 3.53 will be positive and as $(\delta\omega + \delta\theta)$ becomes negative the root of equation 3.53 will be negative. This equation shows that radial and along track separations can be expressed explicitly in terms of each other. The simulation analysis and usefulness of this equation is given in the simulation section.

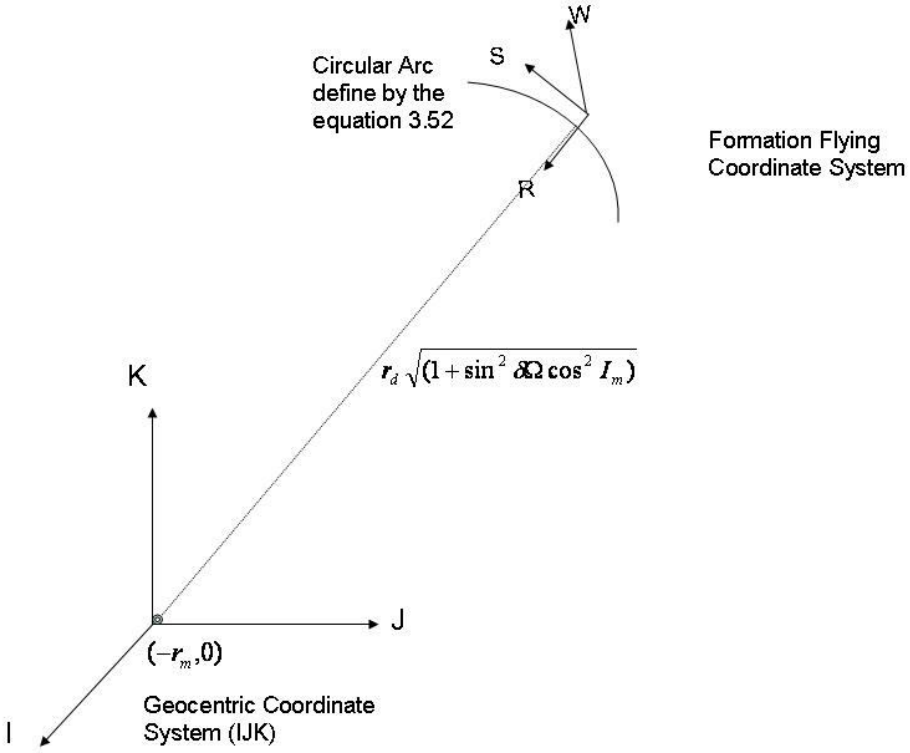


Figure (3.6): A figure showing the graphical interpretation of the explicit equation 3.52

An analysis is presented here to estimate error introduces in difference model short form (DMSF) due to the assumptions used in its derivation. i.e

$\cos \delta\Omega \cong 1 \cong \cos \delta.i$, and

$\cos\theta_m \sin \delta\Omega \cos i_d \sin(\delta\omega + \theta_d) = \cos\theta_m \sin \delta\Omega \cos i_m \sin(\delta\omega + \theta_d)$

The orbital parameters used for the analysis are as

Rd=Rm (Km)	7714
Thetam (rad)	1.570796326794
Thetad (rad)	1.56079632679
Inclination –Im (rad)	1.570896326794
Inclination -Id (rad)	1.570896326794
$\delta.i$ (rad)	0
$\delta\omega$ (rad)	0

Plot in figure 3.7 is showing deviation of DMSF model's results from DMLF model as $\delta\Omega$ is varied from zero to 0.01 in steps of 0.0001 radians, while keeping $\delta.i$ constant. It shows that the affect due to assumption about $\delta\Omega$ is most on along track separation. For $\delta\Omega$ up to 0.004 radians, the deviation in radial, along track and cross track separation is very small (less than 50 cm) and if $\delta\Omega$ goes up to 0.003 radians, the deviation in radial, along track and cross track separations are less than 30cm. The plot in figure 3.8 is showing the deviation in DMSF from the difference model long form (DMLF) as $\delta.i$ varies from 0 to 0.01 in steps of 0.001 radians, while $\delta\Omega$ is kept constant. Radial separation is showing most deviation as $\delta.i$ varies. For $\delta.i$ up to 0.001 radians, the deviation in radial separation is less than 3m and the deviation in along track, cross track separations are zero.

The dependence of cross track separation on variations in $\delta\Omega$ is studied by varying $\delta\Omega$ from 0 to 0.1 in steps of 0.0001, while keeping other parameters in equation 3.48 constant. The plot is shown in figure A.1 (Appendix A). It shows that cross track separation increases with the increase in $\delta\Omega$

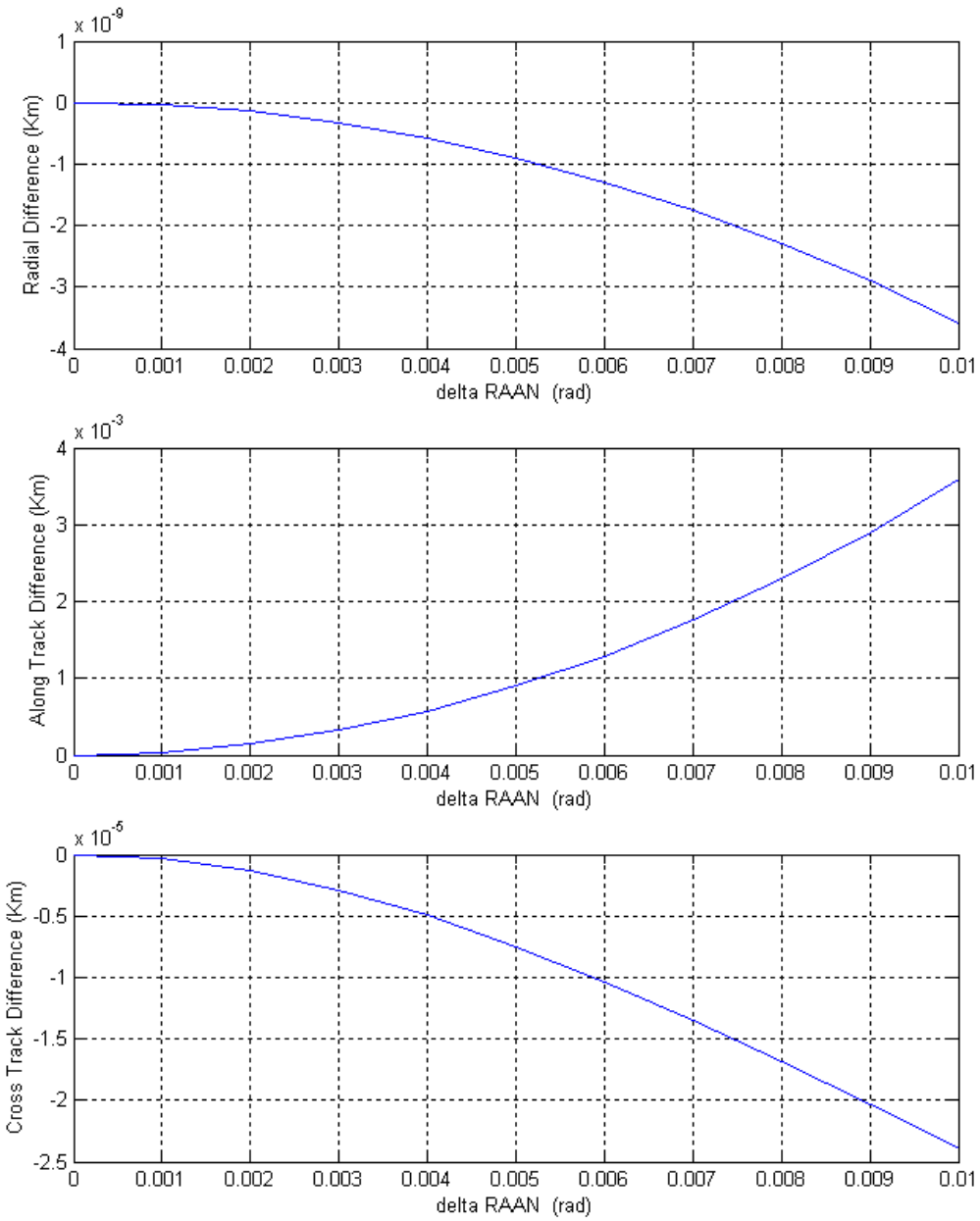


Figure 3.7: Plot showing the difference between Long and Short Form model as $\delta\Omega$ varies

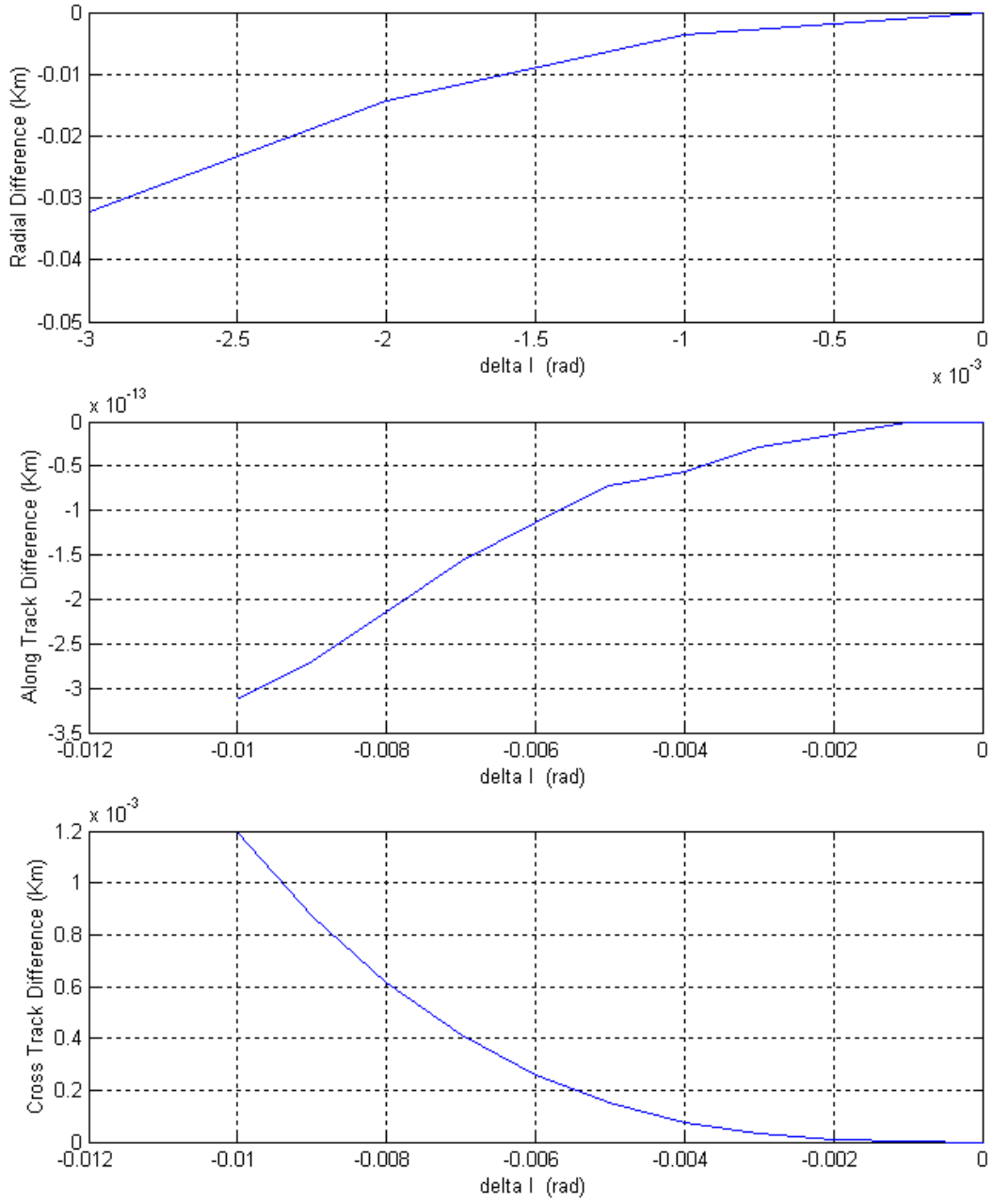


Figure 3.8: A plot showing the difference between Long and Short Form verses difference in inclination angle

B – Simulation

A real time satellite formation flying model is developed in terms of three unit vectors. In this section a simulation algorithm is developed for this model and other models discussed in chapter2. Perturbations due to the non spherical earth and atmospheric drag are considered during simulations and perturbation models and their effects on relative dynamics are presented here. Different SFF models discussed use different initial conditions to start their simulations, some use initial positions of the satellites in the formation in geocentric Cartesian coordinate some required relative states between the satellites, some required orbital elements and some need differences in orbital elements. So a common input file is developed which generates the same initial conditions for all the models. Matlab/SIMULINK was used to develop and run the simulations. At first offline simulations were run to evaluate the performance of the Real Time Satellite Formation Flying (RTSFF) model by comparing its results with other models. A network of two real time single board computers were developed and the RTSFF were downloaded on to it. Real time simulations were run by using Real Time Windows Target toolbox of the Matlab. Its results were compared with offline simulation results to verify its performance for the real time test bed.

3.7 Simulation Input File:

To run the simulation of a formation flying model some initial information is needed about all the satellites in the formation and their relative states with respect to master satellites. The RTHIL formation flying models is defined in terms of three unit vectors $\hat{r}_m$, $\hat{h}_m$ and $\hat{r}_d$, so these unit vectors must be defined at time zero in an inertial geocentric frame (IJK frame). Hill's equations are defined in terms of relative position and velocity vector so to run the simulation by using Hill's equation the relative position vector (x_r, y_r, z_r) and velocity vector $(\dot{x}_r, \dot{y}_r, \dot{z}_r)$ of the deputy satellite in the master's local frame of reference(RSW) is needed as initial conditions. The COWPOKE Model is defined in terms of the difference of orbital elements and master orbital elements, hence the simulation for COWPOKE Model needs the orbital elements of master satellites and orbital elements difference of master and deputy satellites as initial conditions. Three sets of input data are needed for the simulation, but all these input data sets are transformable from one form to other form. Assuming that the

master's orbital elements $e_m(t), a_m(t), \Omega_m(t), \omega_m(t), \theta_m(t), i_m(t)$ and deputy satellite's $e_d(t), a_d(t), \Omega_d(t), \omega_d(t)\ \ \theta_d(t), i_d(t)$ are known.

The magnitude of the radius vector for the master satellite is

$$r_m = \frac{a_m(1-e_m^2)}{1+e_m\cos\theta_m} \qquad\qquad r_d = \frac{a_d(1-e_d^2)}{1+e_d\cos\theta_d}$$

Components of radius vector in perifocal coordinate system is

$$\vec{r}_m = [r_m\cos\theta_m, r_m\sin\theta_m, 0] \qquad\qquad \vec{r}_d = [r_d\cos\theta_d, r_d\sin\theta_d, 0]$$

This position can be transformed into a geocentric frame through following an Euler transformation

$$[\vec{R}_m]_{IJK} = [M_t]\begin{bmatrix} r_m\cos\theta_m \\ r_m\sin\theta_m \\ 0 \end{bmatrix}$$

$$[\vec{R}_d]_{IJK} = [M_t]\begin{bmatrix} r_d\cos\theta_d \\ r_d\sin\theta_d \\ 0 \end{bmatrix}$$

Where $[M_t]$ is transfer matrix from perifocal coordinate system to IJK (Geocentric frame of reference) frame and is given by

$$[M_t] = \begin{bmatrix} c\Omega_m & -s\Omega_m & 0 \\ s\Omega_m & c\Omega_m & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & ci_m & -si_m \\ 0 & si_m & ci_m \end{bmatrix}\begin{bmatrix} c\omega_m & -s\omega_m & 0 \\ s\omega_m & c\omega_m & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{..... (3. 54)}$$

Where c and s are short for cosine and sine trigonometric ratios. Velocity vector in perifocal frame

$$\left.\begin{aligned} \vec{v}_m &= \sqrt{\frac{\mu}{a_m(1-e_m^2)}}[-\sin\theta_m, (\cos\theta_m + e_m), 0] \\ \vec{v}_d &= \sqrt{\frac{\mu}{a_d(1-e_d^2)}}[-\sin\theta_d, (\cos\theta_d + e_d), 0] \end{aligned}\right\} \quad \text{..... (3. 55)}$$

Transform it into geocentric frame by using the transformation given in 3.53

$$[\vec{V}_m]_{IJK} = \sqrt{\frac{\mu}{P_m}}.[M_t]\begin{bmatrix} -\sin\theta_m \\ \cos\theta_m + e_m \\ 0 \end{bmatrix}$$

$$[\vec{V}_d]_{IJK} = \sqrt{\frac{\mu}{P_d}}.[M_t]\begin{bmatrix} -\sin\theta_d \\ \cos\theta_d + e_d \\ 0 \end{bmatrix} \quad \text{..... (3. 56)}$$

Transform position and velocity from IJK(Geocentric frame of reference) to RSW (formation flying frame associated with master satellite) frame.

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix}_{RSW} = [A]\begin{bmatrix} X_d \\ Y_d \\ Z_d \end{bmatrix}_{IJK} - \begin{bmatrix} R_m \\ 0 \\ 0 \end{bmatrix} \quad \text{..... (3. 57)}$$

$$\begin{bmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{Z}_r \end{bmatrix}_{d/RSW} = [A]\begin{bmatrix} \dot{X}_d \\ \dot{Y}_d \\ \dot{Z}_d \end{bmatrix}_{IJK} \quad \text{..... (3. 58)}$$

$$\begin{bmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{Z}_r \end{bmatrix}_{m/RSW} = [A]\begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{Z}_m \end{bmatrix}_{IJK} \quad \text{..... (3. 59)}$$

Where $[A]$ is a transfer matrix from IJK to RSW frame and is given as

$$[A] = \begin{bmatrix} \cos u_m & \sin u_m & 0 \\ -\sin u_m & \cos u_m & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i_m & \sin i_m \\ 0 & -\sin i_m & \cos i_m \end{bmatrix}\begin{bmatrix} \cos\Omega_m & \sin\Omega_m & 0 \\ -\sin\Omega_m & \cos\Omega_m & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{..... (3. 60)}$$

The RSW rate of rotation with respect IJK frame of reference is

$$\omega = \begin{bmatrix} 0 \\ 0 \\ \sqrt{\frac{\mu}{a^3}} \end{bmatrix} \quad \text{..... (3. 61)}$$

From Vallado and McClain p 52, [25] the inertial (IJK frame) and rotational (RSW) velocities are related by

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{z}_d \end{bmatrix}_{RSW} = \begin{bmatrix} \dot{X}_d \\ \dot{Y}_d \\ \dot{Z}_d \end{bmatrix}_{RSW} - \omega \times \begin{bmatrix} r_m \\ 0 \\ 0 \end{bmatrix}_{RSW} \quad \text{.....(3. 62)}$$

$$\begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{z}_m \end{bmatrix}_{RSW} = \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{Z}_m \end{bmatrix}_{RSW} - \omega \times \begin{bmatrix} r_m \\ 0 \\ 0 \end{bmatrix}_{RSW} \quad \text{..... (3. 63)}$$

The relative velocity in RSW frame is

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{z}_r \end{bmatrix} = \begin{bmatrix} \dot{x}_m \\ \dot{y}_m \\ \dot{z}_m \end{bmatrix}_{RSW} - \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{z}_d \end{bmatrix}_{RSW} \quad \text{..... (3. 64)}$$

All inputs to satellite formation flying models are known. Simulations were run by using different sets of orbital elements. The details for different formation flying scenario and simulation results are presented here.

3.8 Simulation Algorithm and Perturbation:

In order to use RTSFF model, the geocentric coordinates of master and deputy satellites should be known at each time step. These geocentric coordinates (X, Y, Z) can be calculated by solving two body dynamic equations given as.

$$\left.\begin{aligned} \ddot{X} &= -\frac{\mu X}{r^3} \\ \ddot{Y} &= \frac{Y}{X}\ddot{X} \\ \ddot{Z} &= -\frac{\mu Z}{r^3} \end{aligned}\right\} \qquad \text{.....(3. 65)}$$

As LEO satellites are considered in this project. LEO satellites orbit varies from 300 to 1000 Km. At these attitude satellites are under some disturbing forces, which tend to deviate them from Keplarian trajectory. These perturbation forces are.

- Gravitational variations due to the oblateness of the Earth
- Atmospheric drag and lift
- Solar Radiation Effect
- Gravitational pull of other lunar Bodies, like Sun, Moon etc.
- Effect of Inhomogeneous Earth

In LEO J2 and atmospheric drag are the most prominent perturbing components. In this project only the effect of J2 and atmospheric drag perturbations are considered on formation flying drift.

3.9 Perturbation Acceleration due to Nonspherical Earth:

In deriving the Keplerian Trajectory equation, it was supposed that earth is sphere, but actually it is not. The Earth is flattened at the pole and bulges out at the equator. The mass distribution in earth is also not homogeneous.

It is difficult to model the exact model of gravity. A Formation Flying mission named GRACE has been launched on 17th, March 2002 to map accurately earth's gravity field. A gravity map[34] developed by University of Texas Centre for Space Research from GRACE data is shown in the figure (3.9). For theoretical purposes a simple and close approximation is to consider the earth as an ellipsoid. The mass distribution is symmetric about the north-south axis of the earth. The components of perturbed acceleration due to non sphericity in geocentric frame are given by Bate R. [33] by using Vinti [35] gravitational potential function.

$$\ddot{X} = \frac{\partial \Phi}{\partial x} = -\frac{\mu x}{r^3}[1 - J_2 \frac{3}{2}(\frac{r_e}{r})^2(5\frac{z^2}{r^2} - 1) + J_3 \frac{5}{2}(\frac{r_e}{r})^3(3\frac{z}{r} - 7\frac{z^3}{r^3}) - J_4 \frac{5}{8}(\frac{r_e}{r})^4(3 - 42\frac{z^2}{r^2} + 63\frac{z^4}{r^4})$$

$$- J_5 \frac{3}{8}(\frac{r_e}{r})^5(35\frac{z}{r} - 210\frac{z^3}{r^3} + 231\frac{z^5}{r^5}) + J_6 .....] \quad .....(3.66)$$

Since mass distribution about north-south axis of the earth is symmetric, therefore

$$\ddot{Y} = \frac{\partial \Phi}{\partial y} = \frac{y}{x}\ddot{X} \quad ..... (3.67)$$

and z-component is

$$\ddot{Z} = \frac{\partial \Phi}{\partial z} = -\frac{\mu z}{r^3}[1 + J_2 \frac{3}{2}(\frac{r_e}{r})^2(3 - 5\frac{z^2}{r^2}) + J_3(\frac{r_e}{r})^3(10\frac{z}{r} - \frac{35}{3}\frac{z^3}{r^3} - \frac{r}{z})$$

$$- J_4 \frac{5}{8}(\frac{r_e}{r})^4(15 - 70\frac{z^2}{r^2} + 63\frac{z^4}{r^4}) - \frac{J_5}{8}(\frac{r_e}{r})^5(315\frac{z}{r} - 945\frac{z^3}{r^3} + 693\frac{z^5}{r^5} - 15\frac{r}{z}) \quad ...(3.68)$$

Where Φ is Viniti [35] gravitational potential function. The first term in the above expression is the two body acceleration and the rests of the terms are the perturbed accelerations due to the oblatness of earth. The values of J_s in these expressions are determined through experimental observations. The values of these coefficients by Bate R. are [33]

$J_2 = (1082.64 \pm 0.03) \times 10^{-6}$

$J_3 = (-2.5 \pm 0.1) \times 10^{-6}$

$J_4 = (-1.6 \pm 0.5) \times 10^{-6}$

$J_5 = (-0.15 \pm 0.1) \times 10^{-6}$

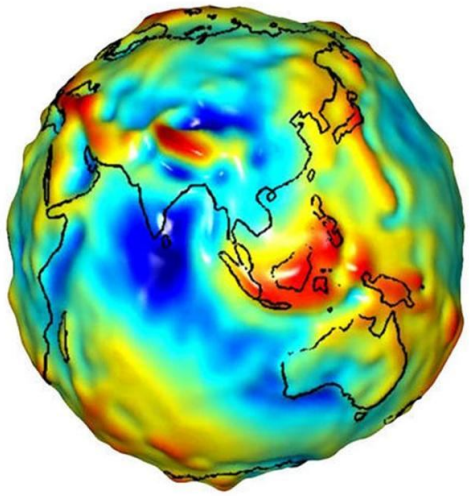


Figure 3.9 :Gravity Model mapped by GRACE mission

3.10 Atmospheric Drag:

At low earth orbits (<1000Km) the atmosphere presents resistance to spacecraft flight. This resistance depends upon the frontal area of the spacecraft, drag coefficient, atmospheric density, relative velocity and altitude. Drag is a non conservative force and hence takes energy away from the orbit. Thus semi major axis and period of the orbit decreases. The velocity of the satellite increases as Kepler's law must be satisfied. It is difficult to predict exact atmospheric drag because of the uncertainties associated with atmospheric fluctuation and drag coefficient. At perigee, the satellite is closest to earth so atmospheric drag will be maximum here. Due to this negative impulse at perigee, the orbit will become more circular after each revolution. This circular orbit then shrinks further and further and finally spiral s rapidly in dense atmosphere. A simple formulation for Atmospheric perturbation acceleration in Geocentric Equatorial coordinate system is given by Bate R and Chobotove as [32, 33]

$$\ddot{\vec{r}} = -\frac{A}{2m} C_d \rho V_a \dot{\vec{r}}_a \quad \text{..... (3. 69)}$$

Where

C_d = Dimensionless Drag coefficient associated with A

A = Cross sectional area perpendicular to the direction of motion.

m = Mass of the Vehicle.

ρ = Atmospheric density at Vehicle's altitude.

$V_a = \left|\dot{\vec{r}}_a\right|$ = Speed of Vehicle relative to the rotating atmosphere.

$$\dot{\vec{r}}_a = \begin{bmatrix} \dot{X} + \dot{\theta} Y \\ \dot{Y} - \dot{\theta} X \\ \dot{Z} \end{bmatrix} \quad \text{and} \quad \dot{r} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \quad \text{..... (3. 70)}$$

$\dot{\theta}$ = Rate of rotation of the Earth.

X, Y and Z are geocentric equatorial coordinate system.

Earth completes its one rotation in 23 hours 56 min 4.09054 Sec (7.2921e-005 rad/sec) and it is assumed that the atmosphere rotates at the same rate.

Where X, Y and Z refers to Geocentric Equatorial Coordinate system and speed relative to rotating atmosphere is

$$V_a = \sqrt{\dot{X}^2 + 2\dot{\theta}\dot{X}Y + \dot{\theta}Y^2 + \dot{Y}^2 - 2\dot{\theta}X + \dot{\theta}^2 X^2 + \dot{Z}^2} \quad \text{..... (3. 71)}$$

The Earth's atmospheric density is affected by a number of factors including the amount of solar radiation striking the atmosphere and by the Earth's magnetic field. A number of density models are in use today.

A analytical atmospheric model, which considers the gravity and temperature variation along the altitude is used here to estimate the variation in the atmospheric density with altitude and is given by Frank J R as[36].

$$\frac{\rho}{\rho_i} = \left[\frac{L_z}{Tm_i}(z - z_i) + 1\right]^{-\left\{\frac{g_0}{RL_z}\left[\frac{RL_z}{g_0} + 1 + b\left[\frac{Tm_i}{L_z} - z_i\right]\right]\right\}} \exp\left[\frac{g_0 b}{RL_z}(z - z_i)\right]$$

..... (3. 72)

Where

ρ = Atmospheric Density

R = Specific Gas constant at sea level = $287.0 J/(Kg.K)$

Lz = Temperature Gradient at Geometric height z (K°/m)

Tm = Molecular Temperature (K°)

$b = 2/R_E$, where R_E is radius of earth

g_0 = Sea level Gravity Constant = $9.8 m/\sec^2$

The subscripts i denote the values at the beginning of ith Atmospheric layer. In this model atmosphere is considered as a layered structure, each layer characterized by a distinct value of the thermal gradient. The values for ρ, Lz and Tm for different layer can be taken from Table 2.1 and Table 2.5 of Frank J R [36].

As this RTSFF model is being developed for a real time SFF test bed, it can perform in open loop and close loop real time simulation. In order to run close loop real time hardware in the

loop simulation, the feedback thrust affect should be taken in to account. If actual thrusters are connected in loop, then a thrust sensor a Analogue to Digital converter and scale factor can be used to include its affect into RTSFF model. Once the thrusters along three axis are known, then satellite dynamics can be adjusted by including their affect as

$$\ddot{x} = \frac{T_x}{m}, \qquad \ddot{y} = \frac{T_y}{m}, \qquad \ddot{z} = \frac{T_z}{m}$$

In case actual thrusters are not connected in loop, then their software or transfer function can be included. Thrust calculation for ion thruster is given by Thomas L [37]as

$$T = \left(2.q.n.V_0.A\right)$$

$$T = \left(\frac{2}{V_0}.\frac{m_p}{q}\right)^{1/2}.P$$

$$I_{sp} = \frac{1}{g}\left(\frac{2qV_\circ}{m_p}\right)^{1/2} \qquad \text{.....(3. 73)}$$

Where T is thrust

I_{sp} = Specific Impulse

m_p = mass of particles

q = charge on particle

V_0 = Accelerating potential

P = Jet Power

These thrust affect can be included in RTSFF dynamic model with the help of above mentioned thrust equation along three axes. In this project software model is used as actual

thrusters were not available. In the absence of guidance and control algorithm, the affect of thrusters has been set to zero by setting $T_x = T_y = T_z = 0$ during the simulation.

In order to solve the perturbed acceleration equations for satellite's state vector, Cowell's methods were used. This is a simple and straight forward method and solves the two body perturbation problem by using a numerical integration technique. It is easy to program in SIMULINK block diagram. The perturbed equation of motion for two body problem with J2 and atmospheric perturbations in Cartesian coordinate is given by Sidi and Bate [33, 38] following equations

$$\left.\begin{aligned}
\ddot{X} = F_x &= \frac{\mu X}{r^3}\left[-1+\frac{3}{2}J_2 R_e{}^2\left(5\frac{Z^2}{r^4}-\frac{1}{r^2}\right)-\frac{A}{2m}C_d\rho V_a\left(\dot{X}+\dot{\theta}Y\right)+\frac{T_x}{m}\right] \\
\ddot{Y} = F_y &= \frac{\mu Y}{r^3}\left[-1+\frac{3}{2}J_2 R_e{}^2\left(5\frac{Z^2}{r^4}-\frac{1}{r^2}\right)-\frac{A}{2m}C_d\rho V_a\left(\dot{Y}-\dot{\theta}X\right)+\frac{T_y}{m}\right] \\
\ddot{Z} = F_z &= \frac{\mu Z}{r^3}\left[-1+\frac{3}{2}J_2 R_e{}^2\left(5\frac{Z^2}{r^4}-\frac{3}{r^2}\right)-\frac{A}{2m}C_d\rho V_a\dot{Z}+\frac{T_z}{m}\right]
\end{aligned}\right\} \quad \ldots\ldots\ldots\ldots \text{(3. 74)}$$

with initial condition $r(t_0)$ and $v(t_0)$.

The main advantage of this method is the simplicity in implementation and formulation. Any number of perturbations can be handled at a time. There are two sources of errors associated with this method. The first is called truncation error, while the second is known as round-off error.

Truncation errors can be reduced by using smaller integration step size, but smaller step size will severely affect the computational speed and accumulative error due to round off. Because of the high precision computer, rounding errors, are no longer an issue. High speed computers have made it possible to reduce the integration step size considerably without slowing down the computational speed. This in turn reduces the truncation error. As a whole we can say the advent of high speed and high precision computer has decreased manifold the integration errors associated with Cowell's Method.

The Cowell's method can handle large magnitude perturbations. This method is used to simulate the real time formation flying model developed in this research. Runge-Kutta order 4 integration method is used for numerical integration. As the intention in developing this

formation flying model is to use it for real time hardware in the loop simulation, a fixed step size integration method is used here.

In order to add perturbation affects in COWPOKE equations, the variation of parameters method was used. The first order earth oblateness perturbation affects the secular changes in the Right ascension in ascending node Ω, argument of perigee ω, and mean anomaly *M* in the following way.[32]

$$\dot{\Omega} = -\frac{3n\,\mathrm{Re}^2\,J_2\cos i}{2p^2}$$

The expression for $\delta\Omega$ by Sabol [10] is given as

$$\delta\Omega = \Omega m - \Omega d$$

$$\delta\Omega = \delta\Omega_0 + (\dot{\Omega}_m - \dot{\Omega}_d) * t \quad \text{.....(3. 75)}$$

$$\dot{\omega} = \frac{3nJ_2\,\mathrm{Re}^2}{4p^2}(4 - 5\sin^2 i) \quad \text{..... (3. 76)}$$

$$\dot{M}_{j2} = \frac{3nJ_2\,\mathrm{Re}^2\sqrt{1-e^2}}{4p^2}(2 - 3\sin^2 i) \quad \text{..... (3. 77)}$$

Where $n = \sqrt{\frac{\mu}{a^3}}$ is the mean motion, Re is earth equatorial radius and J_2 is 2nd order earth zonal harmonic coefficient.

Now the expressions for $\delta\omega$ and $\delta\theta$ by Sabol [10] are given as

Similarly

$$\delta\omega = \delta\omega_0 + (\dot{\omega}_m - \dot{\omega}_d) * t \quad \text{..... (3. 78)}$$

$$\omega_m = \frac{3}{4}\sqrt{\frac{\mu}{a^3}}\,\frac{R^2{}_e J_2}{[a(1-e^2)]^2}(4 - 5\sin^2 i) \quad \text{..... (3. 79)}$$

$$\omega_d = \frac{3}{4}\sqrt{\frac{\mu}{(a+\partial a)^3}}\frac{R^2{}_e J_2}{[(a+\partial a)(1-(e+\partial e)^2)]^2}(4-5\sin^2(i+\partial i)) \quad \text{.....(3. 80)}$$

The true anomaly difference can be find out from the following expression[10]

$$\delta\theta = \delta M + 2(e+\delta e)\sin(M+\delta M) - 2e\sin M$$

So to find the change in true anomaly the change in mean anomaly must be determined first.

For no perturbation $\dot{M} = n$ and for J_2 case, $\dot{M} = n + \dot{M}_{j2}$ [32] with $\dot{M}_{j2}$ from equation 3.77

δM can be calculated as

$$\delta M = \delta M_0 + (\omega_{xyd} - \omega_{xym})*t$$

Where

$$\omega_{xym} = \sqrt{\frac{\mu}{a^3}}\left[1+\frac{3R_e^2 J_2\sqrt{1+e^2}}{4[a(1-e^2)]^2}(2-3\sin^2 i)\right]$$

$$\omega_{xyd} = \sqrt{\frac{\mu}{(a+\delta a)^3}}\left[1+\frac{3R_e^2 J_2\sqrt{1+(e+\delta e)^2}}{4[(a+\delta a)(1-(e+\delta e)^2)]^2}(2-3\sin^2(i+\delta i))\right]$$

To find the disturbance on orbital parameter due to atmospheric drag, variation of parameter method was used given by Roy [39].

$$\left.\begin{aligned}
da &== \frac{\rho}{BC}a^2\frac{(1+e^2+2e\cos\theta)^{3/2}}{(1+e\cos\theta)^2}d\theta \\
de &= -\frac{\rho}{BC}a(1-e^2)\frac{(1+e^2=2e\cos\theta)^{3/2}}{(1+e\cos\theta)^2}(\cos\theta+e)d\theta \\
di &= 0 \\
d\Omega &= 0 \\
d\omega &= -\frac{\rho}{BC}\frac{a(1-e^2)}{e}\frac{(1+e^2+2e\cos\theta)^{1/2}}{(1+e\cos\theta)^2}\sin\theta d\theta
\end{aligned}\right\} \quad \text{..... (3. 81)}$$

Where

ρ = Atmospheric density, m = mass of satellite

$BC =$ Ballistic coefficient of satellite and $BC = \dfrac{m}{C_D A}$

The final Keplerian orbital element differences can be evaluated as

$$\begin{aligned} \delta a &= \delta a_0 + (da_d - da_m)t \\ \delta e &= \delta e_0 + (de_d - de_m)t \\ \delta \omega &= \delta \omega_{j2} + (d\omega_d - d\omega_m)t \end{aligned} \quad \text{..... (3. 82)}$$

Where subscript 0 denote initial given orbital element difference and $d\omega_{j2}$ is argument of perigee difference with J2 perturbation.

Satellite formation flying models and their perturbations are programmed in Matlab/SIMULINK[40] and offline simulations were run with 1 sec simulation time step. RK4 integration method was selected. Satellite formation flying simulations were run for different formation flying scenarios to compare their performances.

Case 1: First consider the scenario where master and deputy satellites are moving in same orbital plane with $\omega_m = \omega_d$ and $\Delta I = 0$, $a_m = a_d$, $\Omega_m = \Omega_d$, $e_m \neq e_d \neq 0$.

Orbital parameters of master and deputy satellites for the simulation are given in the Table 1

Data for atmospheric drag are given in Table 2

Parameter	Master Satellite	Deputy Satellite	Difference
Major Axis a (Km)	7714	7714	0
Eccentricity e	0.00001	5.79e-004	0.000569
Inclination – I (rad)	1.57089632679	1.57089632679	0.0
RAAN-- Ω (rad)	1.57089632679	1.57089632679	0.0
Argument of perigee- ω	0	0	0.0
True Anomaly θ (rad)	1.57079632679	1.56079632679	0.01

Table 1: Keplerian Master and Deputy Orbital Elements and differences

Parameter		Value
Step Size (Seconds)		1
Mass (Master)	(Kg)	1000
Mass (Deputy)	(Kg)	1000
Cross section area (Master)	(Km^2)	0.000001
Cross section area (Deputy)	(Km^2)	0.000001
Master Drag Coefficient		2
Deputy Drag Coefficient		2.001

Table 2: Master and Deputy satellite's Data for Atmospheric drag

The affect of Earth oblateness and atmospheric drag perturbations on orbital elements are considered in these simulations. The drag coefficient of Master and Deputy Satellites are slightly different and varies from 2 to 2.001, which results in 250000 Kg/Km^2 differences in ballistic coefficient. The satellites are in a polar orbit formation and due to this cross track separation will not show any drift with time (regression of line of node will be zero in this case). In this case cross track separation will remain constant. As both orbits are elliptical and their argument of perigee do not coincide therefore radial and along track separation will show sinusoidal behavior. The amplitude of the sinusoidal separation will depend upon the eccentricity. A comparison of along track separation between RTSFF, Hills and COWPOKE model for case 1 is given in figure (3.10). It is showing sinusoidal behavior. The altitude of both orbits is nearly 1300 Km. At this height the affect of atmospheric drag is insignificant. Along track motion shows a drift in Hill's result. It is because of the reason that $\dot{y}_0 \neq -2 \cdot n \cdot x_0$ and the secular term in Hill's along track equation do not vanish completely.

The COWPOKE and RTSFF models show similar behavior for along track motion. The error plot for COWPOKE and Hill's models with respect to RTSFF model is shown in figure (3.12). It shows zero error initially and it then grows with time. The reason for these errors is the assumptions used in these models. Hill's simulation results show more deviation from the RTSFF model's results with time, because Hill's model is more sensitive to deviation from centric orbits and frozen orbits $\dot{y}_0 = -2 \cdot n \cdot x_0$. These models are good for short time but then with the time the error start accumulating.

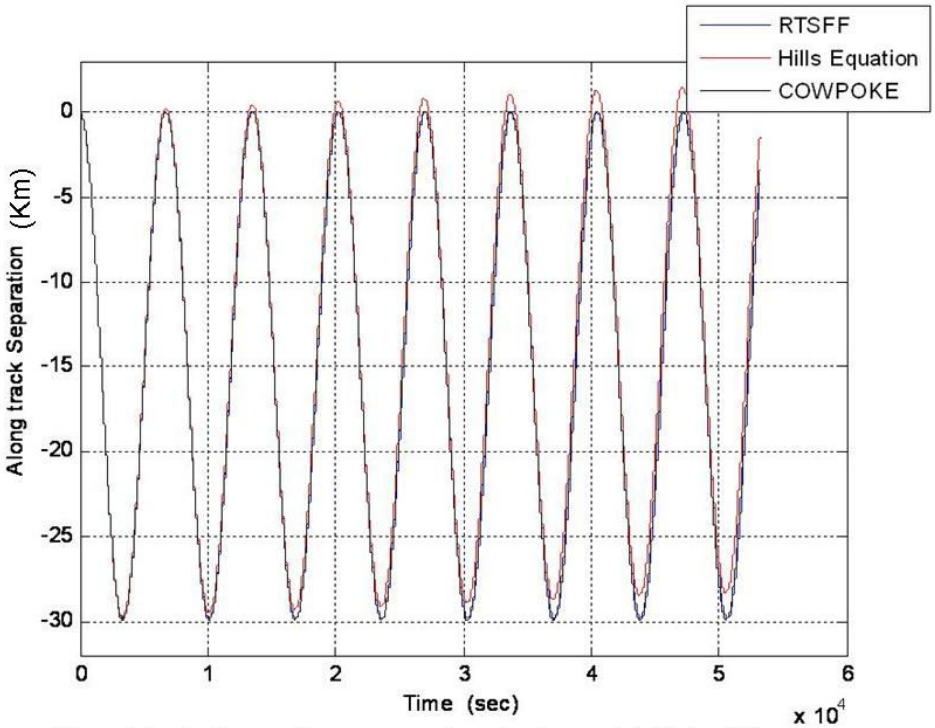
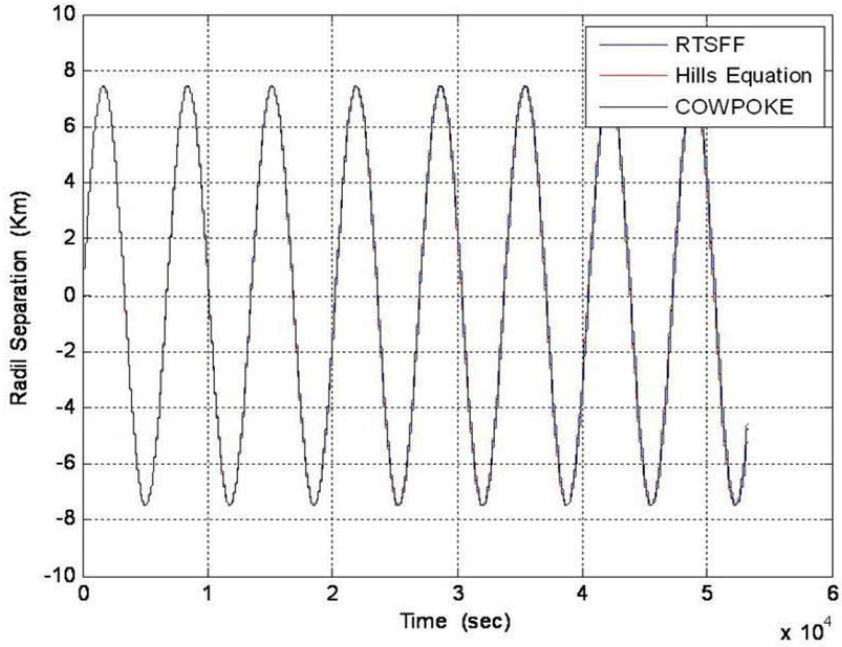


Figure 3.10: Along Track Separation comparison between RTSFF, Hill and COWPOKE equations

The radial separation plot is sinusoidal and is given in figure 3.11. Radial separation oscillates between two extreme from -7.5 to 7.5Km. As the eccentricity is very small and at this small value all models show a very much similar trend. The error plot given in figure 3.13, shows that Hill's model give more error than COWPOKE model, because Hill's model are meant for circular orbits and any deviation from circular orbits will introduce errors. The cross track separation remains zero for all models.



Figure(3.11): Radial Separation Comparison for RTSFF, Hill's andCOWPOKE Equations

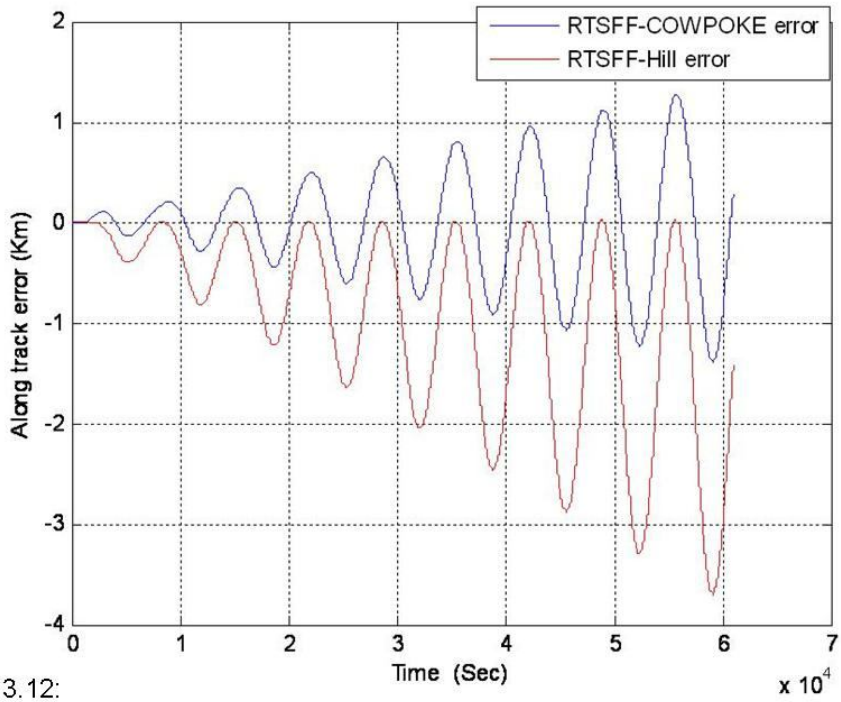


Figure 3.12: Along Track error between RTSFF and COWPOKE and Hill's Equations

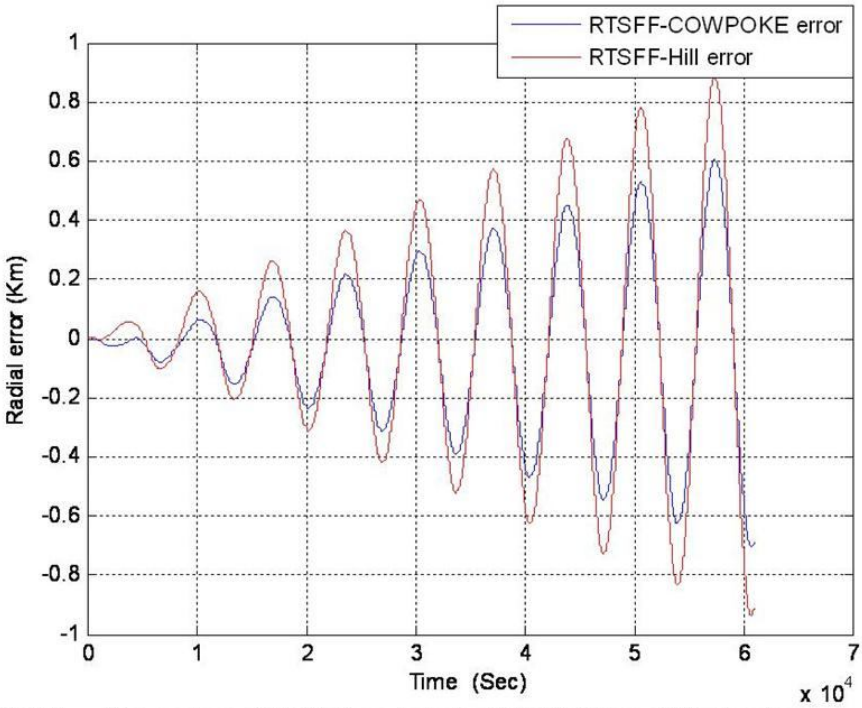


Figure 3.13: Comparison of RTSFF model with COWPOKE and Hill Equations

The plots between along track separation and radial separation are given in figure 3.14 and 3.16 for RTSFF and COWPOKE model. In this case, the orbits of both satellites are elliptical, hence the vectors r_m and r_d are not constants. Therefore the plot shown in figure 3.14 and 3.16 are not circle. As the cross track separation in this case is zero and it is a case of in plane elliptic formation, the plot between radial and along track separation should be elliptical as described in equation 10 under in plane elliptical case by His [41].

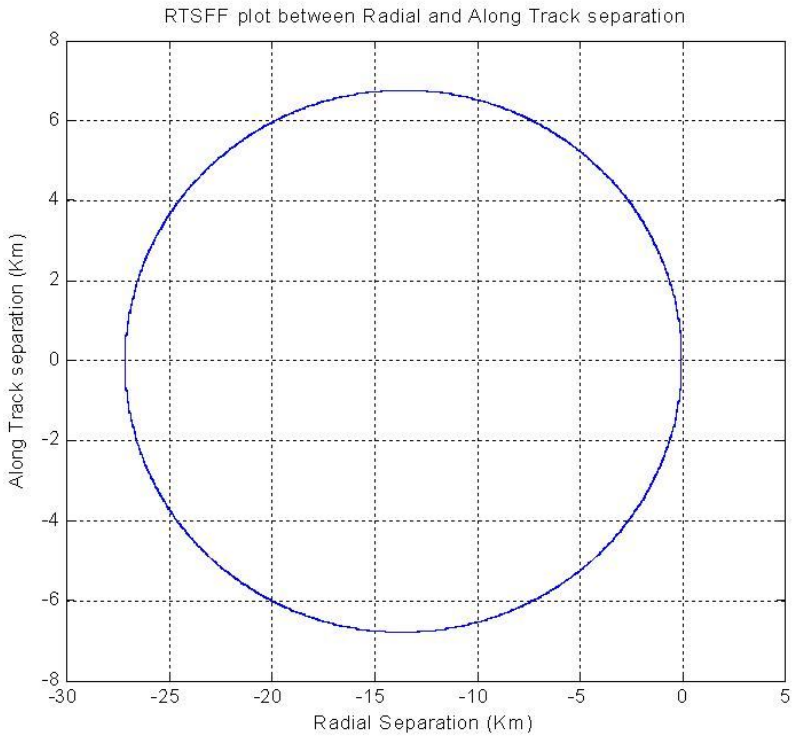


Figure 3.14: A plot between radial and along track separation

By using this explicit equation along track separation is calculated from radial separation. The result is then compared with the along track separation result obtained through simulation of RTSFF model. These plots and their comparison are shown in figure 3.15. Comparison shows that along track separation calculated from explicit equation 3.53 is exactly matching the results obtained through simulation of RTSFF model.

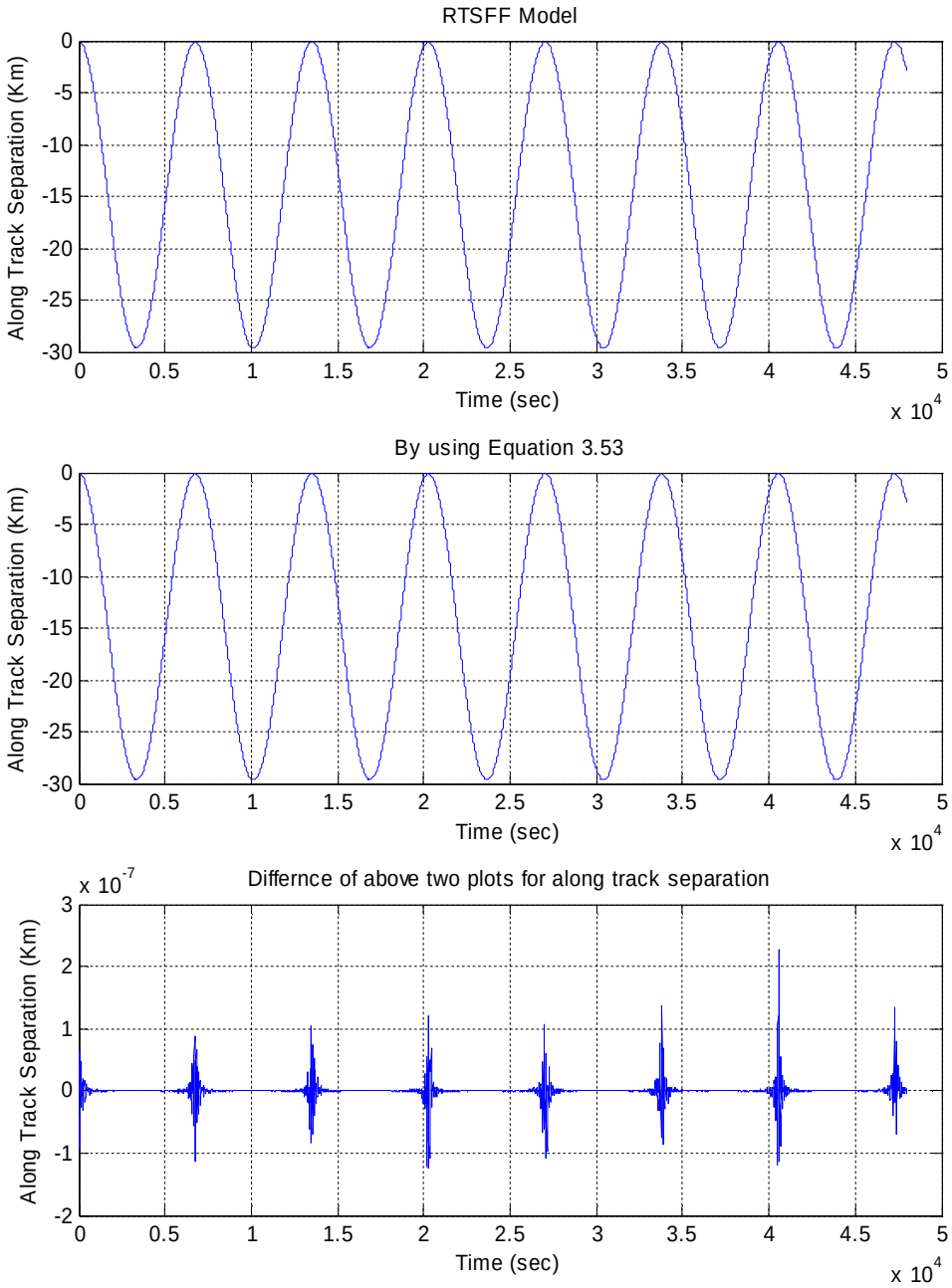


Figure (3.15): A plot showing the result and comparison of explicit equation with RTSFF result

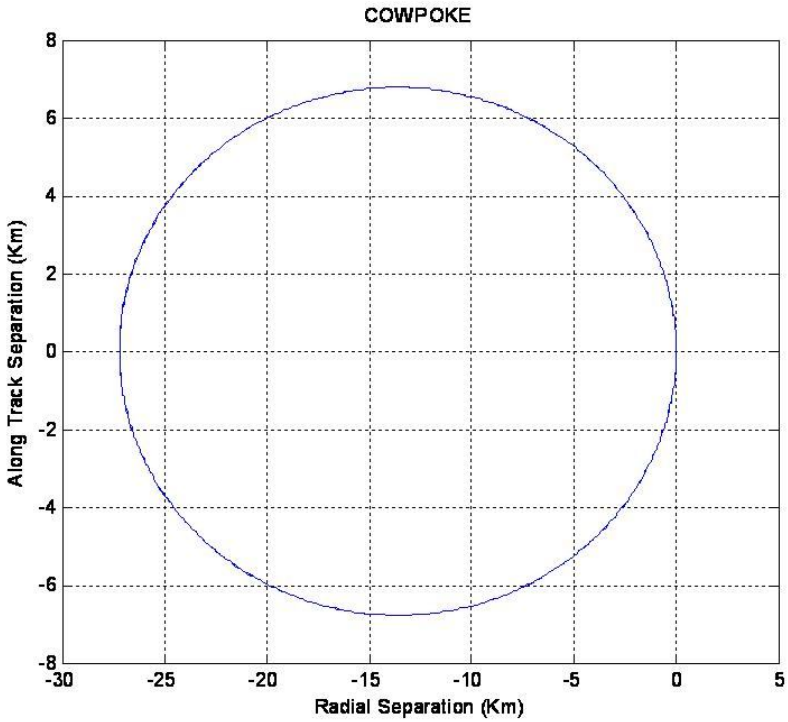


Figure 3.16: A plot between radial and along track separation for COWPOKE model

Case 2: Another formation flying scenario studied was master and deputy satellites are moving in different orbital planes with $\omega_m \neq \omega_d$, $e_m \neq e_d \neq 0$ and either $\Delta I \neq 0$, or $\Omega_m \neq \Omega_d$. The orbital parameters for this simulation are given in Table 3, and atmospheric drag data is used given in table 2.

Parameter	Master Satellite	Deputy Satellite	Difference
Major Axis a (Km)	7178.1363	7178.1363	0
Eccentricity e	10e-8	6.965597e-5	6.955596999e-005
Inclination – I (rad)	0.78539816339	0.78551881102732	1.2064762986e-004
RAAN-- Ω (rad)	0	2.5830872929e-8	2.58308729e-8
Argument of perigee-	0	4.712822530293	4.712822530293
Mean Anomaly M (rad)	0	1.570362759956	1.570362759956

Table 3: Keplerian Master and Deputy Orbital Elements and differences

In this scenario gravitational perturbation will affect the relative motion and that will be proportional to the distance between the satellites. Both master and deputy satellites have

direct orbits and inclination angles less than 63.4 degrees. In this case the apogee will rotate in the direction of motion and the master satellite will have a slightly higher rate of apogee rotation than the deputy satellite. Hence along track separation will increase in one part of the orbit and will decrease in the other part. This drift can be seen in the simulation results for the along the track separation in figure (3.17)

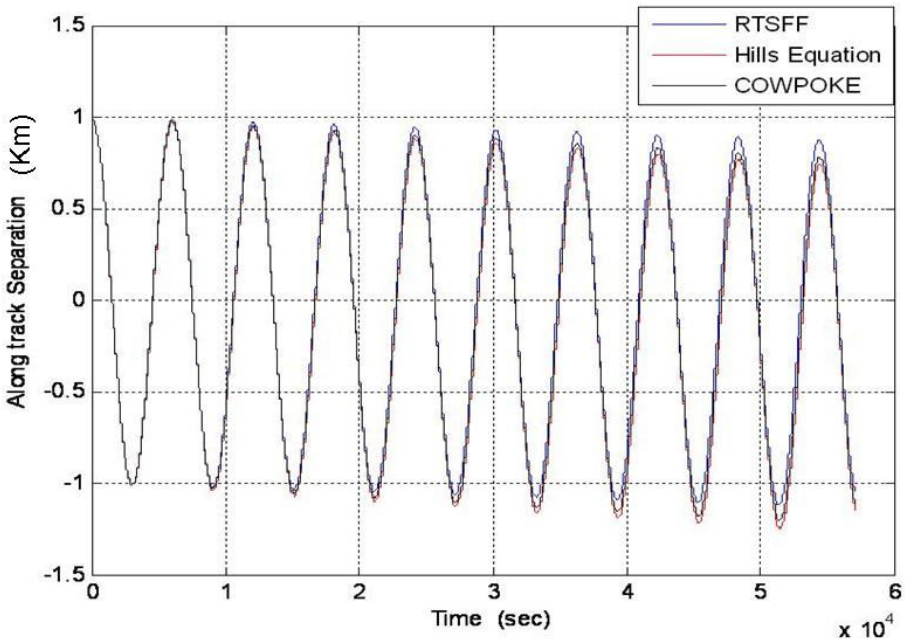


Figure (3.17): Along Track Separation results for RTSFF, COWPOKE and Hill's Equation

The error plot in figure 3.18 shows that COWPOKE and Hill's model give more drift compared with the RTSFF model. Extra drift in Hill's simulation result is again due to the reason that $\dot{y}_0 \neq -2 \cdot n \cdot x_0$. COWPOKE model give curvilinear separation in terms of radians, while converting these curvilinear separations to their respective sector length also introduces some error as the radius is not constant over the sector length.

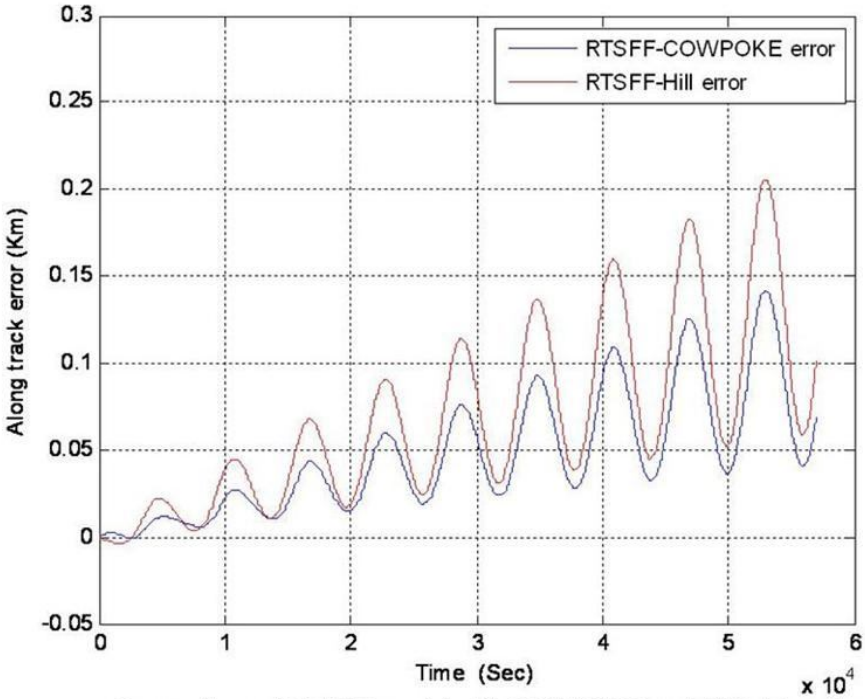


Figure 3.18: Comparison of RTSFF model with COWPOKE and Hill's Equation for along track separation

Both satellites have equal semi major axis length and same time period. The radial separation and its sinusoidal behavior in this case are due to difference in eccentricity and difference in argument of perigee. As the difference in argument of perigee is 90 degree and the deputy satellite is slightly more eccentric (apogee of deputy satellite is bigger than apogee of master satellite), the radial separation should show sinusoidal behavior with slightly more positive peak value than negative peak value. The radial separation simulation results are shown in figure (3.19), which shows the same behavior as discussed above. The error plot for radial separation given in figure (3.20) shows that radial error in COWPOKE and Hill's results increases with time. In the COWPOKE model the relative separations are measured in spherical coordinate system. In spherical coordinate system, the radial separation is measured by taking the difference of the master and the deputy satellite's position vector, while all other SFF models discussed here, measures radial separation in formation flying frame (RSW).

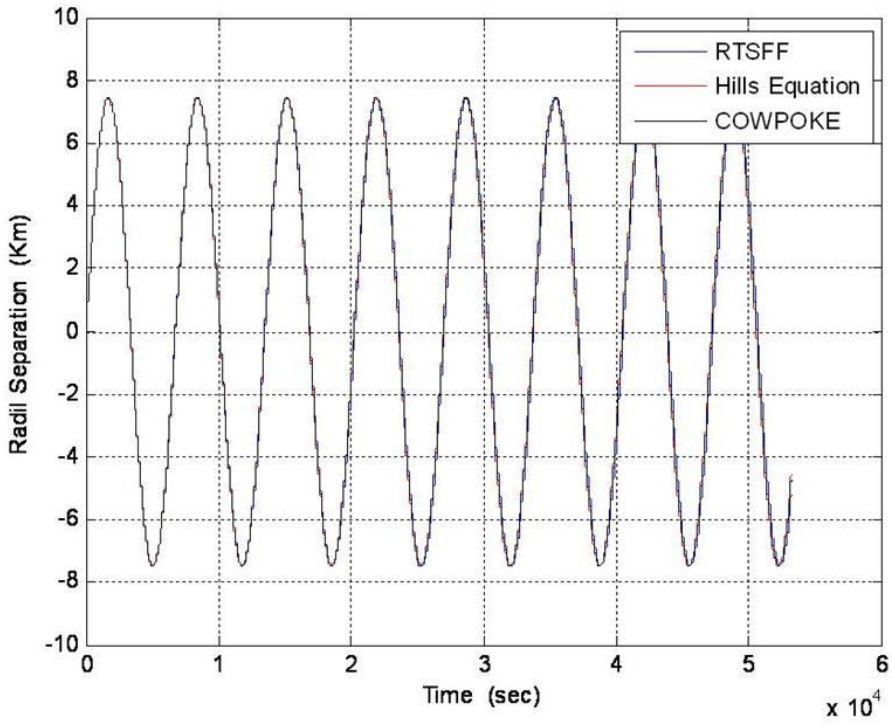


Figure 3.19: Radial Separation Comparison for RTSFF, Hill's and COWPOKE Equations

For small along track separation, the error in radial separations measured in two frames will be small. With the increase in the along track separation, the error in the predicted radial separation using the COWPOKE model will increase. Due to the regression of nodes, the $\delta\Omega$ increases, which results in the increase of cross track separation. The cross track separation for three models is presented in figure (3.21). The amplitude of cross track separation increases and half of the time separation is positive and half of the time it is negative as was expected. Initially cross track separation varies between 0.8654 and 0.8648 Km. After 16 hours, amplitude of cross track separation expands to range 0.8680 to 0.8673 Km as is shown in figure A.2 (Appendix A). The dominant source of error in the COWPOKE model for cross track separation is approximation in the expression of true anomaly Chris Sabol [10], which increases as eccentricity increases. Initially both COWPOKE and Hill's model shows zero error with RTSFF model for cross track separation, but as time passed the error starts accumulating.

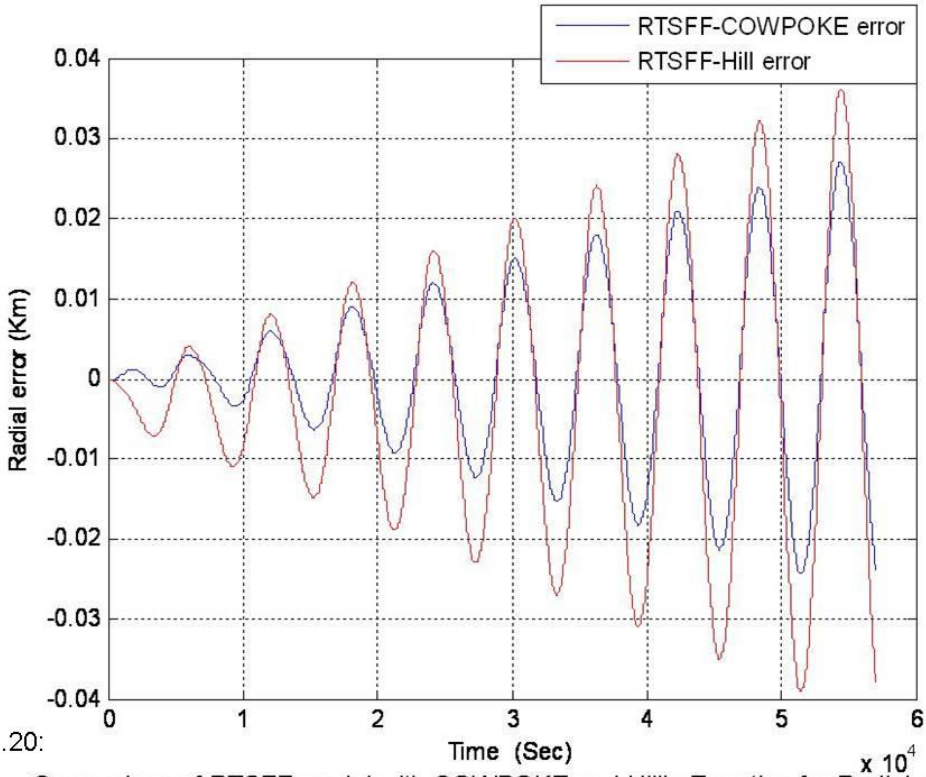


Figure 3.20: Comparison of RTSFF model with COWPOKE and Hill's Equation for Radial error

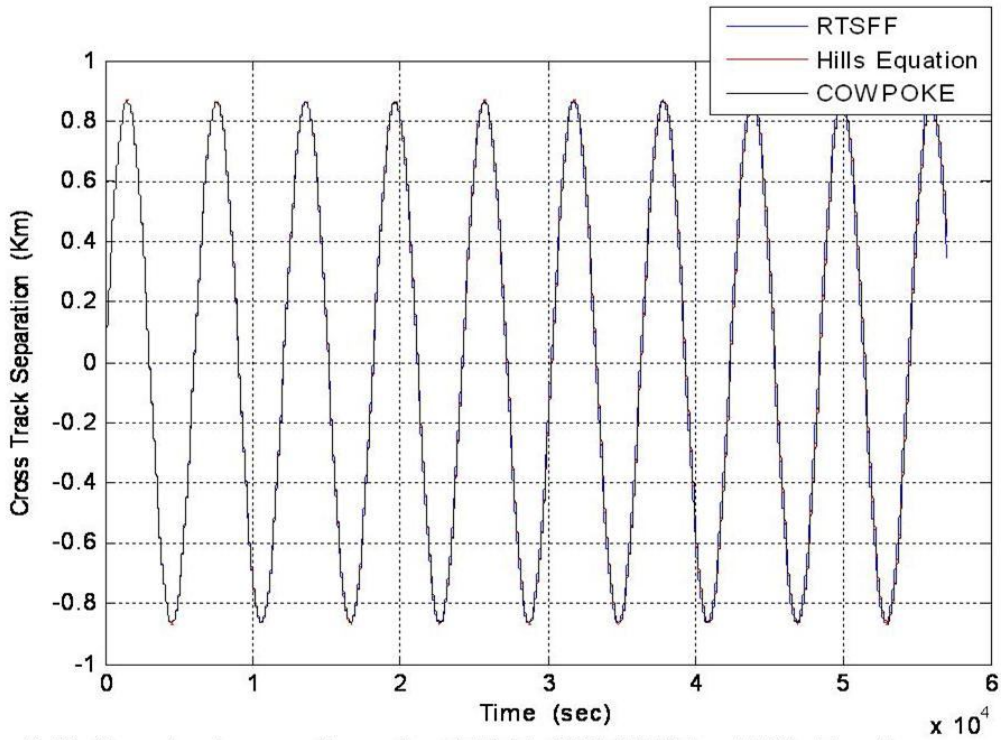


Figure 3.21: Cross track separation using RTSFF , COWPOKE and Hill's Equation

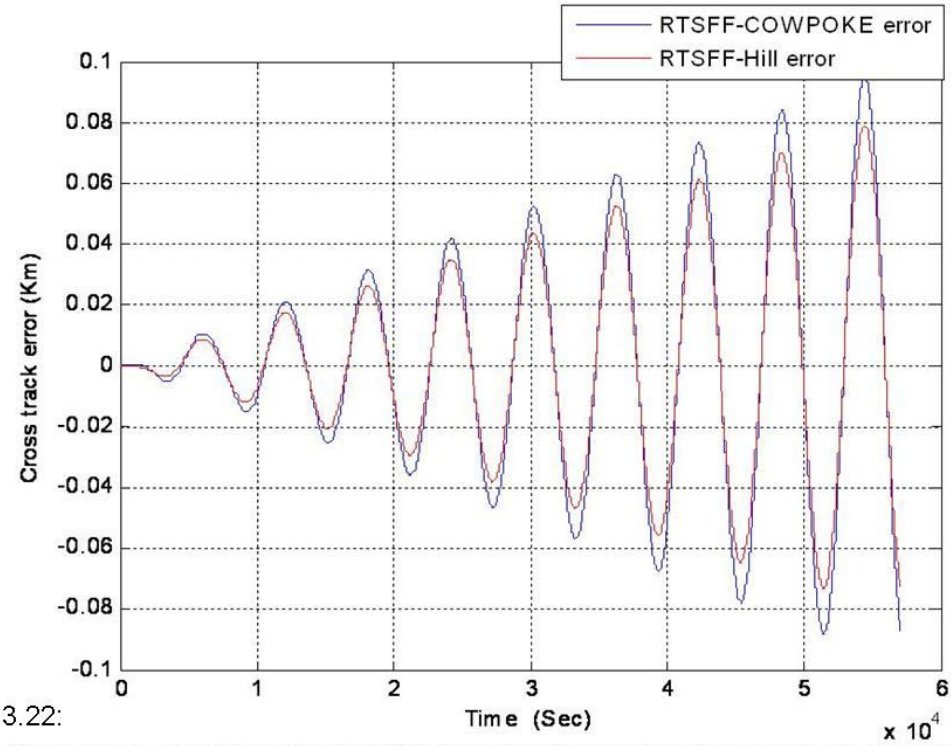


Figure 3.22: Comparison of RTSFF with COWPOKE and Hill's Equation for cross track separation

The simulations for these models were run for different formation flying scenario and were found quite satisfactory. RTSFF model shows very accurate results as there is no assumption used in its derivation. COWPOKE model's results follow RTSFF results very closely for very close formation, but error start growing as soon as formation separation increases. Especially radial separation is very sensitive to separation between formations. Hill's model follows the RTSFF model's results, but the errors are more than COWPOKE' error. The error in Hill's results increases more as orbits become more and more eccentric and separation between the satellites increases. The assumption used in the derivation of COWPOKE and Hill' model impose some restriction and limitation on them, e.g. Hill's model solution is only stable for orbits satisfying the relation $\dot{y}_0 \neq -2 \cdot n \cdot x_0$. The truncation in Fourrier-Bassel series in COWPOKE derivation introduces truncation error Chris Sabol [10].

Plots between along track and radial separation for the RTSFF and COWPOKE models are shown in figure 3.23 and 3.24. Each point of this plot satisfies the equation 3.52. The center and radius of the circle is not fixed because the orbits are elliptical.

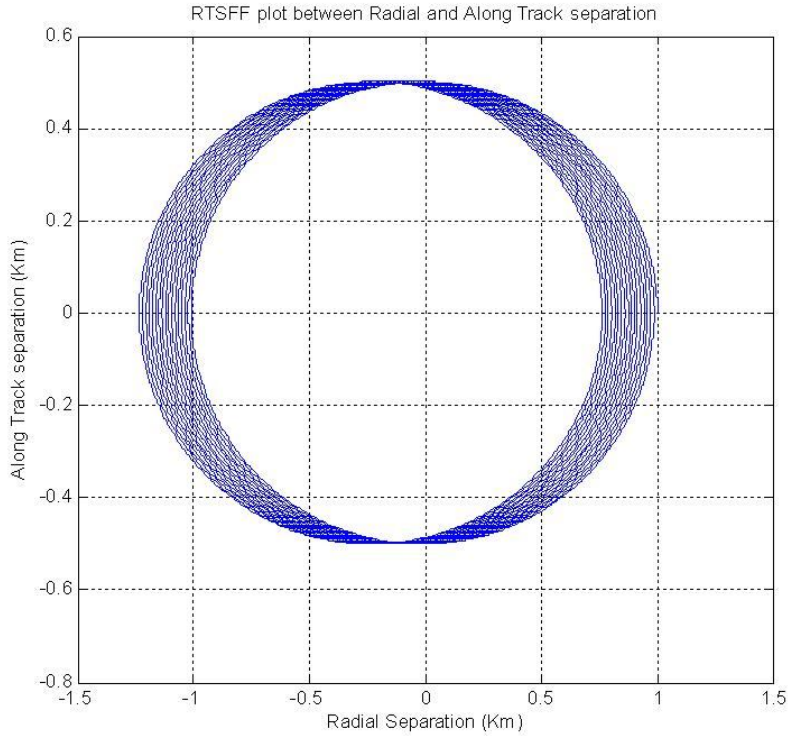


Figure 3.23: A plot between radial and along track separation for RTSFF model

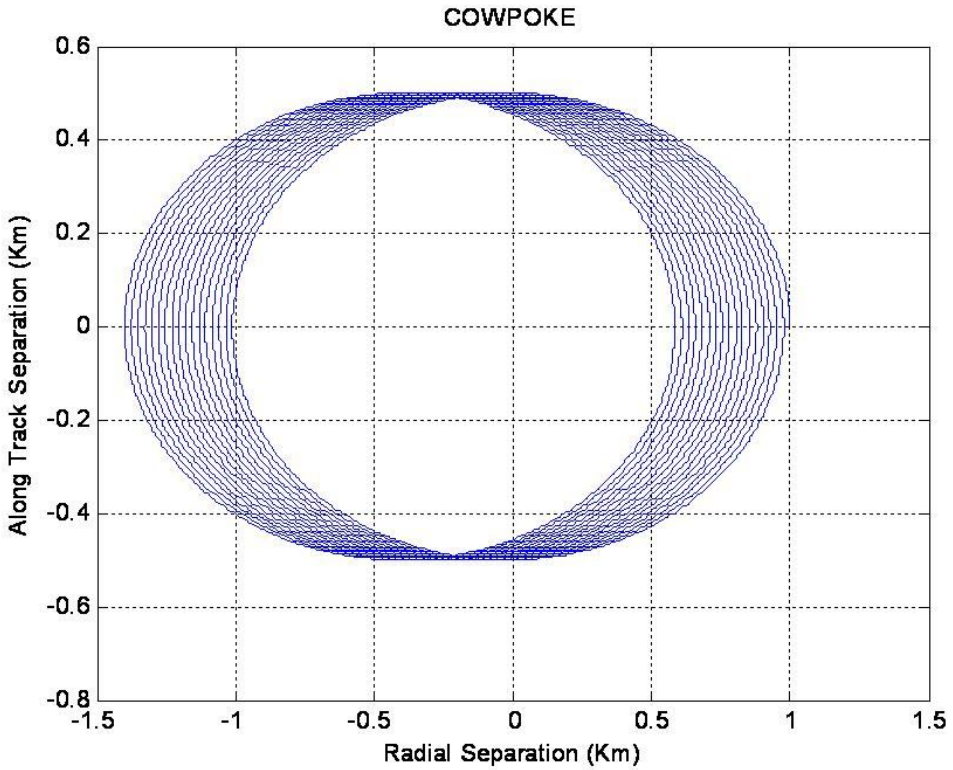


Figure 3.24: A plot between radial and along track separation for COWPOKE model

3.11 Short Form SFF model Simulation and comparison:

The Short Form model was developed to show the applicability of RTSFF model in a perifocal coordinate frame. This is a unique model that gives very simplified expressions for relative separation in terms of difference of orbital elements. Only two assumptions were used in its derivation that are $\cos\delta\Omega \cong 1 \cong \cos\delta i$. There is no limitation on eccentricity of orbits and along track separation. Offline Simulations were run again by using Matlab/SIMULINK. Orbital parameters and data for atmospheric drag are given in the table below.

Parameter	Master Satellite	Deputy Satellite	Difference
Major Axis a	7078.1363	7078.1363	0
Eccentricity e	0.00699	0.001	0.00599
Inclination – I	1.57079632679	1.57089581056	9.948376e-5
RAAN-- Ω (rad)	1.57079632679	1.57089581056	9.948376e-5
Argument of	0	0	0.0
True Anomaly θ	1.57079632679	1.56079632679	-0.01

Parameter		Value
Step Size (Seconds)		1
Mass (Master)	(Kg)	1000
Mass (Deputy)	(Kg)	1000
Cross section area (Master)	(Km^2)	0.000001
Cross section area (Deputy)	(Km^2)	0.000001
Master Drag Coefficient		2
Deputy Drag Coefficient		2.001

The simulation results for short form model and its comparison with other models are presented here in figure (3.25) to figure (3.29). The simulation results show that Short Form model is better than COWPOKE and Hill's model. It shows better results as compared to COWPOKE and Hill's models, for more eccentric orbits and for greater along track separation. Also it is easy to visualize expected separations from the Short Form model for different formation flying scenario without running simulation, like if $\delta i = \delta\Omega = 0$, then cross

track separation will be zero, and formation will lie in x-y plane with radial and along track motion showing sinusoidal behavior as

$$x_r(t) = \cos(\delta\omega + \delta\theta) r_d - r_m \qquad y_r(t) = \sin(\delta\omega + \delta\theta) r_d$$

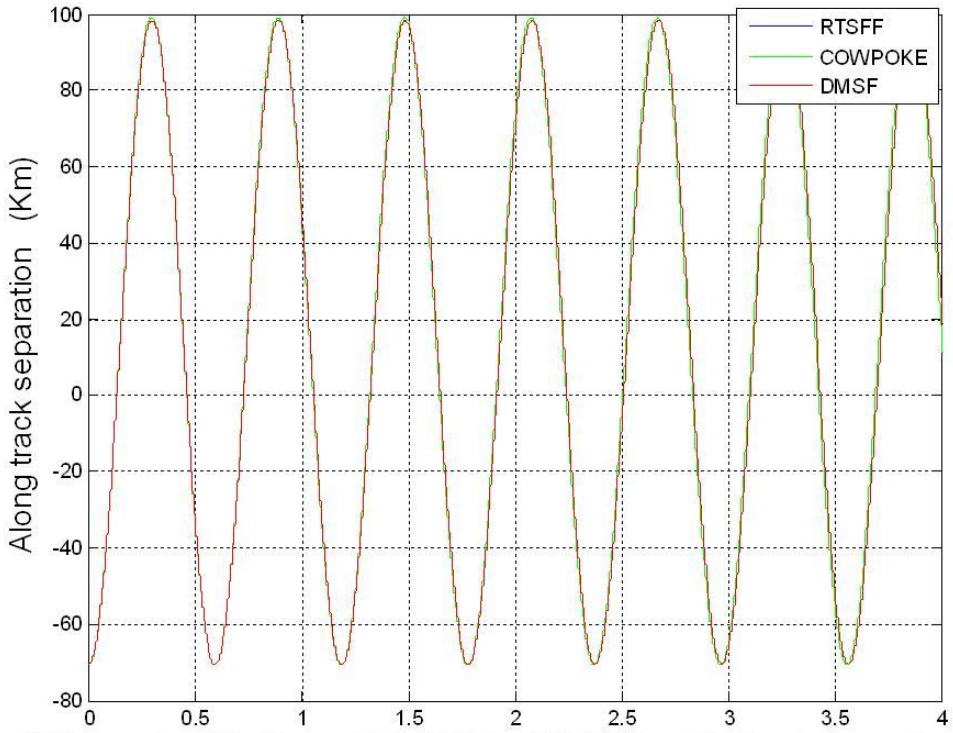
If $(\delta\omega + \delta\theta) = 0$ and $\delta\Omega = 0$, then along track separation will be zero, $X_r(t) = r_d - r_m$ and $z_r(t) = \delta i \sin\theta_m$.

For polar orbit formation, i..e $I_m = 90^\circ$, Then relative separations come out as

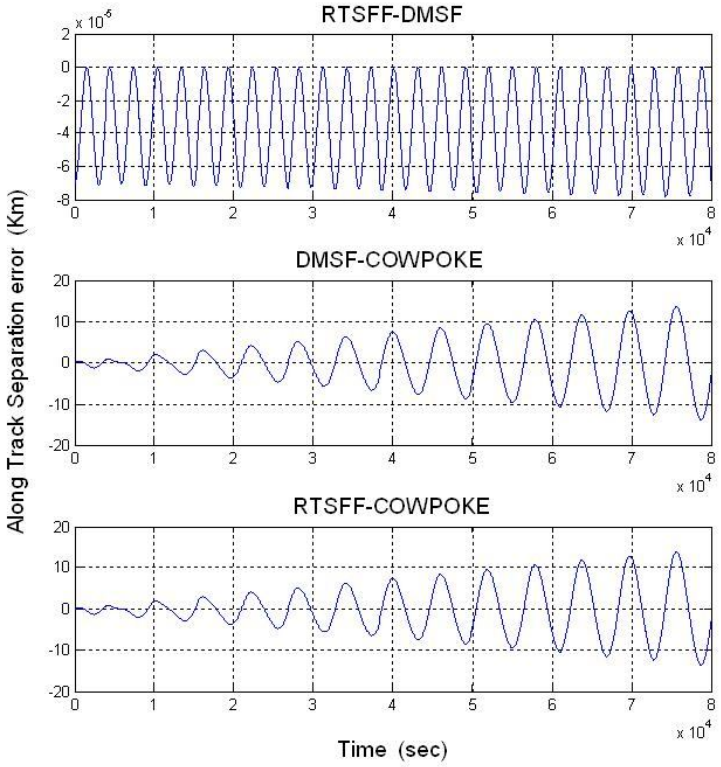
$$x_r(t) = \cos(\delta\omega + \delta\theta) r_d - r_m$$

$$y_r(t) = \sin(\delta\omega + \delta\theta) r_d$$

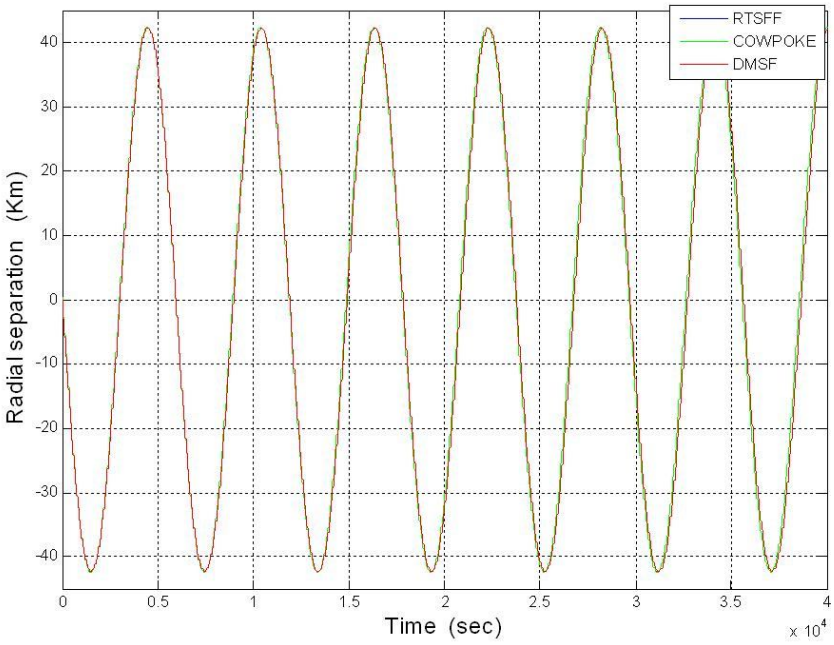
$$z_r(t) = [\delta i \sin(\delta\omega + \theta_d) - \delta\Omega \cos(\delta\omega + \theta_d)] r_d$$



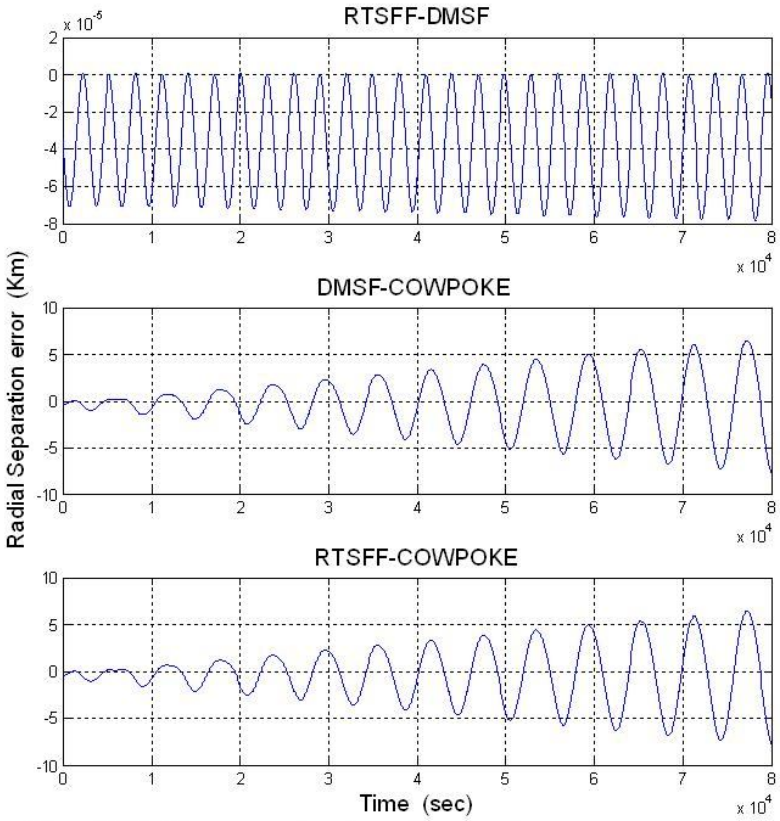
Figure(3.25):Comparison of Short form model with RTSFF and COWPOKE model for along track separation



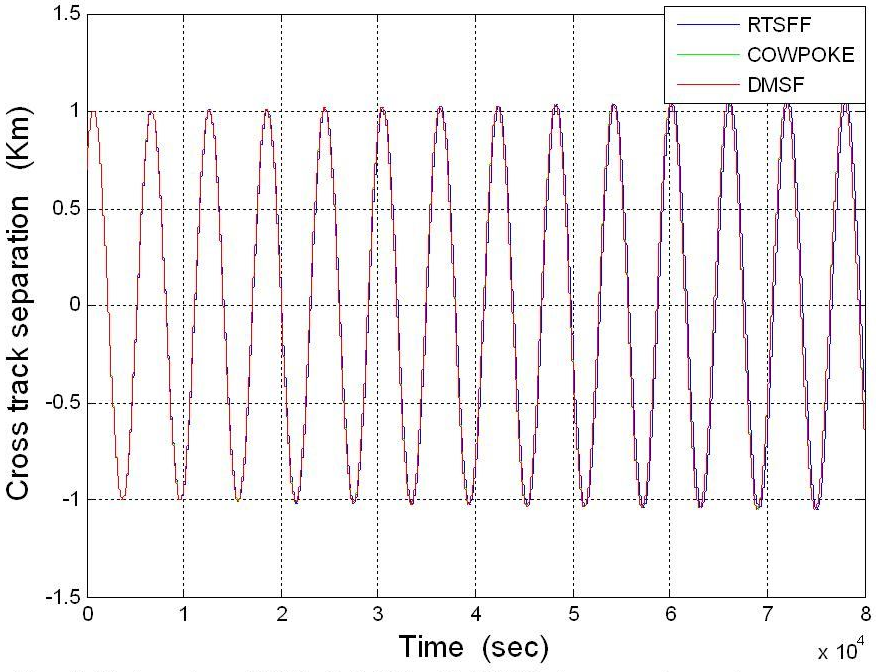
Figure(3.26):Comparison of DMSF, RTSFF and COWPOKE model for along track separation error



Figure(3.27): Comparison of DMSF mode with RTSFF and COWPOKE for radial separation



Figure(3.28): Error comparison for radial separation



Figure(3.29): Comparison of DMSF with RTSFF and COWPOKE for cross track separation

3.12 Real Time Simulation [31, 42]:

The first phase of the project has been completed successfully. In this phase a real time model was downloaded on to two single board computers (JUKI 745E[43]) with Windows OS, Matlab and SIMULINK[40] running on them, as shown in figure 3.30. Real Time Windows Target[44] is used to convert the Windows operating system into a real time operating system, to run the real time simulation and for hardware interfacing. Visual C was used to write hardware interfaces and user code in C-sfunction (A Simulink facility to add user code in Simulink model). A serial link was developed between the satellites to transmit master current states to the deputy satellite. The deputy satellite computes its relative states in the master frame of reference. A real time simulation has been run for more than 2 hours with a 1 sec simulation time step. In this simulation setup, a leader-follower configuration is used.

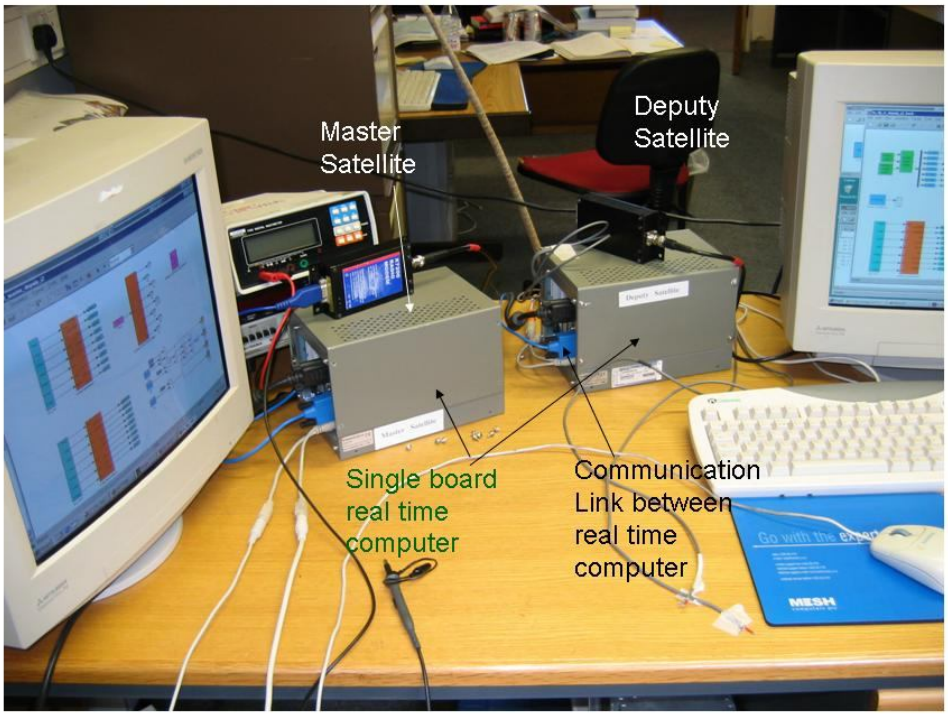


Figure 3.30: Real Time Spacecraft Formation Flying simulation setup

All the simulations on different computers were synchronized through a hardware trigger signal to make sure that real time formation flying simulation starts with the same initial conditions as was used for offline simulation. GIS (Graphical Interfacing Software) from the external computer were used to trigger the real time simulations on all computers

simultaneously. Matlab's file 'rt_stub.c' is modified in such a way that it does not start the simulation until it receives a digital trigger signal from GIS. A copy rt_stub.c file is provided in Appendix F.. As soon as trigger signal arrives, it initializes the simulation timers and starts the real time simulation. It makes sure that at each step time the master transmit its current states and the deputy reads this data during the same time step to calculate its relative states. The relative states (position and velocity) are displayed in real time and are recorded in Matlab workspace along with simulation time. A step by step procedure for real time simulation setup with two real time computers is described here as

- Take two computers with any Windows OS, Matlab/SIMULINK, RTW, RTWT and Visual C installed on it. Run RTWT setup utility that will convert Windows operating system into RTOS. Visual C will be used to develop C S-function.
- Break up the RTSFF model into parts. Download these dynamic equations onto a real time computer and name it 'master satellite'.
  - $$\ddot{X}_m = F_x = \frac{\mu X}{r^3}\left[-1+\frac{3}{2}J_2 R_e^{\,2}\left(5\frac{Z^2}{r^4}-\frac{1}{r^2}\right)-\frac{A}{2m}C_d\rho V_a\left(\dot{X}+\dot{\theta}Y\right)+\frac{T_x}{m}\right]$$
  - $$\ddot{Y}_m = F_y = \frac{\mu Y}{r^3}\left[-1+\frac{3}{2}J_2 R_e^{\,2}\left(5\frac{Z^2}{r^4}-\frac{1}{r^2}\right)-\frac{A}{2m}C_d\rho V_a\left(\dot{Y}-\dot{\theta}X\right)+\frac{T_y}{m}\right]$$
  - $$\ddot{Z}_m = F_z = \frac{\mu Z}{r^3}\left[-1+\frac{3}{2}J_2 R_e^{\,2}\left(5\frac{Z^2}{r^4}-\frac{3}{r^2}\right)-\frac{A}{2m}C_d\rho V_a\dot{Z}+\frac{T_z}{m}\right]$$
- Install a copy of the above equations and the relative motion equation given below on to another real time computer and name it deputy satellite.
  - $$x_r(t) = \frac{\vec{r}_m(t)}{r_m(t)}\cdot\vec{r}_d(t) - r_m(t)$$
  - $$y_r(t) = \left(\frac{\vec{h}_m(t)}{h_m(t)}\times\frac{\vec{r}_m(t)}{r_m(t)}\right)\cdot\vec{r}_d(t)$$
  - $$z_r(t) = \frac{\vec{h}_m(t)}{h_m(t)}\cdot\vec{r}_d(t)$$
- Connect the two real time computers through a serial cable (COM1). Write a C S-function that takes absolute states from the master dynamic equation during each time step and transmit it safely to deputy satellite real time computer. Similarly write a program on the deputy's real time computer that accepts information from the master computer with proper handshaking. The copy of these two C S-functions are given in Appendix F.

- Connect each real time computer to the external computer through a parallel cable. GIS software on the external computer will use this cable to trigger the simulation on the real time computers.
- Compile the simulation on both simulation computers. Connect to the hardware and press start simulation option. Now the simulation on both computers are ready and waiting for the trigger signal.
- Press START button on GIS software that will issue a trigger signal through parallel port to the simulation computer.
- Now the simulation starts running in real time and data can been seen in real time on both computer. The deputy computer is storing relative states with a simulation time stamp. Both computers are also sending their absolute states to COM2 serial port, which is supposed to be connected to GPS simulator.

The input file containing initial conditions for formation flying simulation is given below[27].

Table 4: Initial relative conditions for Formation Flying Simulation

Parameter	Master Satellite	Deputy Satellite
Major Axis, a (Km)	7178.1363	7178.1363
Eccentricity, e	1.0 e-8	1.0 e-8+6.964463e-5
Inclination Angle, I (deg)	45	45.0069126
Argument of Perigee, (deg), ω	0	270.02484058
RAAN, (deg), Ω	0	1.48e-6
True Anomaly , (deg), θ	0.0	89.98314028217037
$(\omega + \theta) = U_d$ (deg)	0.0	0.00798086217031
Radial Separation, x (Km)		0
Along Track Separation, (Km)		-1
Across Track Separation, (Km)		0

The plots for real time simulation are presented in figure 3.31. The results of real time simulations and offline simulations were compared and are shown in figure 3.32. There is an error of less than 10^{-8} Km in all cases. This clearly shows that real time simulation setup, simulation synchronization method, data exchange interface files on both computers and execution of real time model in the real time hardware in loop simulation with 1 sec simulation time step gives results similar to the offline simulation. This simulation results demonstrate that this setup of distributed network of real time computers with formation flying algorithm distributed on them can be used in hardware in the loop simulation test bed without losing any

accuracy or loss of data for any length of time depending upon computer's memory and speed of serial link. A conference paper with the title 'A Comparison of Spacecraft Formation Flying Models and use in Real Time Hardware in the Loop Simulations' has been presented at 4th Workshop on Satellite Constellations and Formation Flying at Brazil. [42].

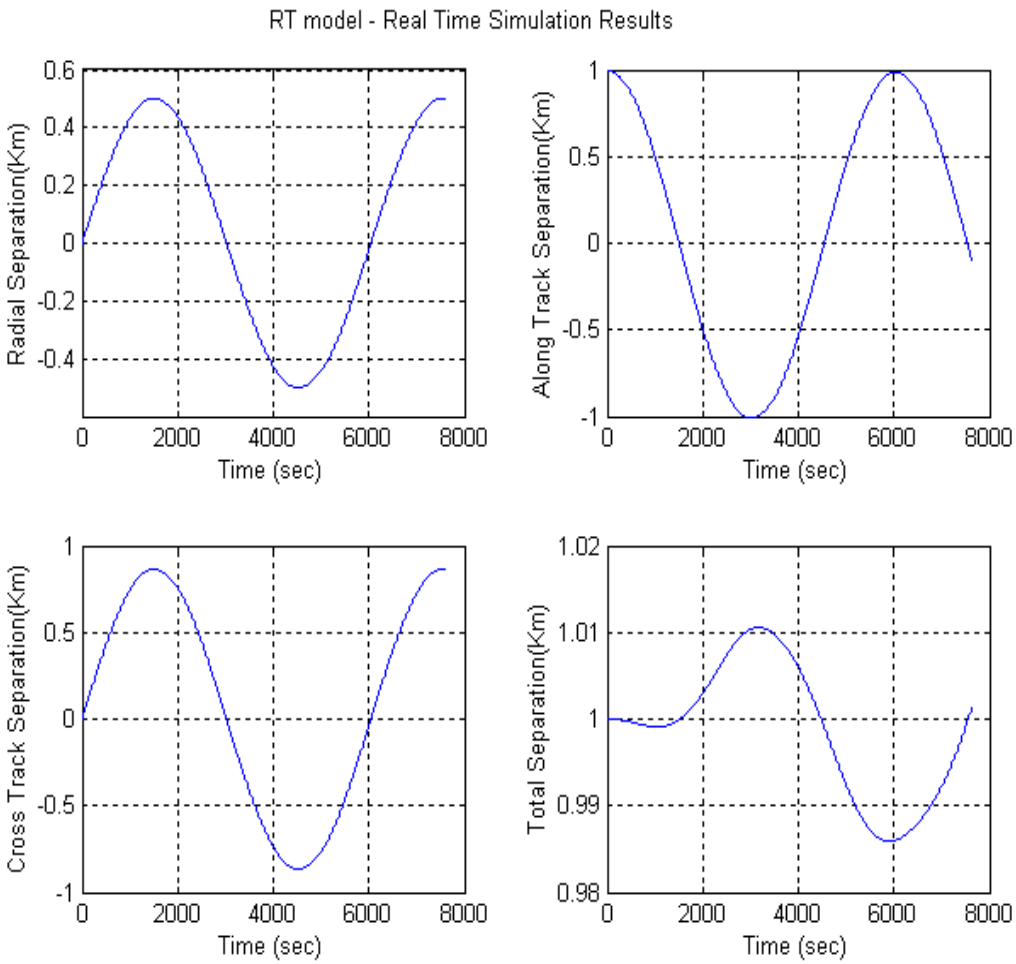


Figure (3.31): Real Time Simulation results by using RTSFF model on two real time computers with parameters defined in Table 4.

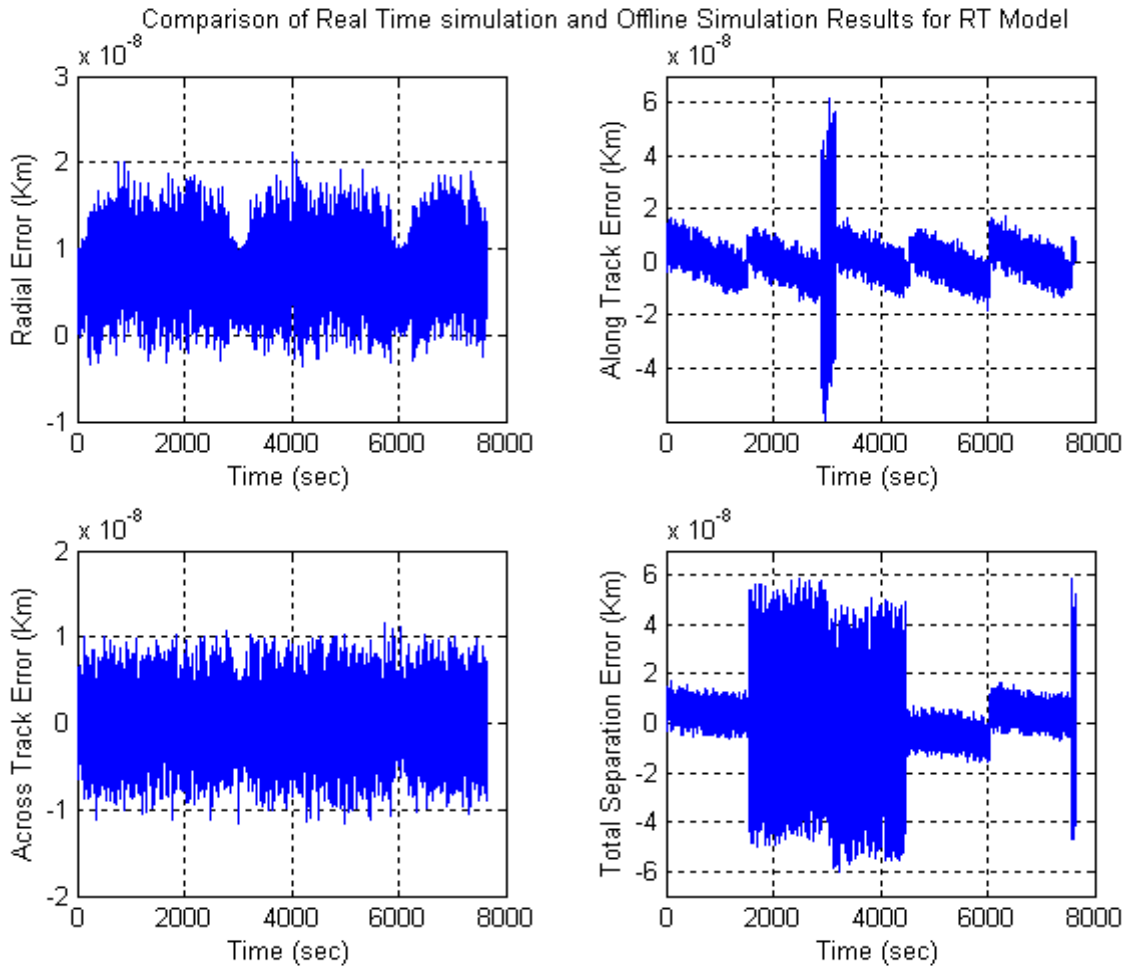


Figure (3.32): Comparison of RTSFF model with Real Time simulation and offline simulation results

3.13 Testing of the RTSFF Model:

For the verification of the RTSFF model different cases were run. Two formation flying scenarios were selected from Sabol [10, 26]. First a LEO formation flying mission with elliptical orbits was selected. Simulations were run for both the RTSFF model and the COWPOKE model for one orbit. Simulation plots and error plots are given from figure 3.33 to 3.36. RTSFF simulation results are compared with COWPOKE simulation results. Sabol in his paper compare and verify COWPOKE results with the truth data determined by the algorithm given by Vallado[25]. The comparison plots of truth data with COWPOKE model are taken from Sabol published paper [10] and are given in figure 3.37 and 3.38 The simulation results and the comparison of the RTSFF and COWPOKE models reveals that both models show similar behavior. A comparison of figures 3.37 and 3.34 and figures 3.38 and 3.36 shows that the errors are very similar. This suggests that the RTSFF model is closer to the truth data and hence more accurate than COWPOKE model.

The second formation flying scenario is for a LEO circular orbit formation. Simulations were run for Hill's and RTSFF models. For circular orbits, Hill's equations are accurate. The RTSFF model's simulation results were compared with Hill's model. Figure 3.39 shows the error plot between the RTSFF and Hill's models for cross track separation. After one orbit (6052 sec) the error becomes equal to the error at zero time. The variation in error during the orbit is due to the reason that periodic variations in orbital elements are also considered in RTSFF model whereas it is not included in Hill's simulation. Similarly the figure 3.40 shows the error plot between the RTSFF and Hill models for radial separation. The error after each orbit remains equal to the error at zero time. It shows that there is no drift between the results of the two models simulation results and both models show the same results except the differences due to periodic variations in orbital elements.

In circular formation flying with no perturbations the secular drift in along track, cross track and radial separation should be zero and the relative separation should come back to the same relative states after one orbit. This can be verified by plotting along track separation against radial separation. In the case of zero secular drift, the plot should be closed. A simulation was run for 7 orbits and the plot between the radial and along track separation is

shown in figure 3.41. It shows a complete close plot without any drift. When the J2 perturbation is switched ON and the simulation is run again for 7 orbits nit can be seen from Figure 3.42 that there is drift after each orbit due to the J2 perturbation affect. This indicates the stability and accuracy of RTSFF model for long time simulation.

This analysis and the simulation results show that the RTSFF model is good and accurate for both circular and elliptical orbits and shows very similar results to a truth data model for elliptical orbits.

Simulation results without any perturbation:

Table 5: Formation flying initial orbital parameters for elliptical orbit [10]

Parameter	Master Satellite	Difference
Major Axis a (Km)	7000	0.01
Eccentricity e	0.01	0.01
Inclination – I (rad)	0.785	0.01
RAAN-- Ω (rad)	0	0.01
Argument of perigee- ω (rad)	4.712	0.01
Mean Anomaly M (rad)	1.571	0.01

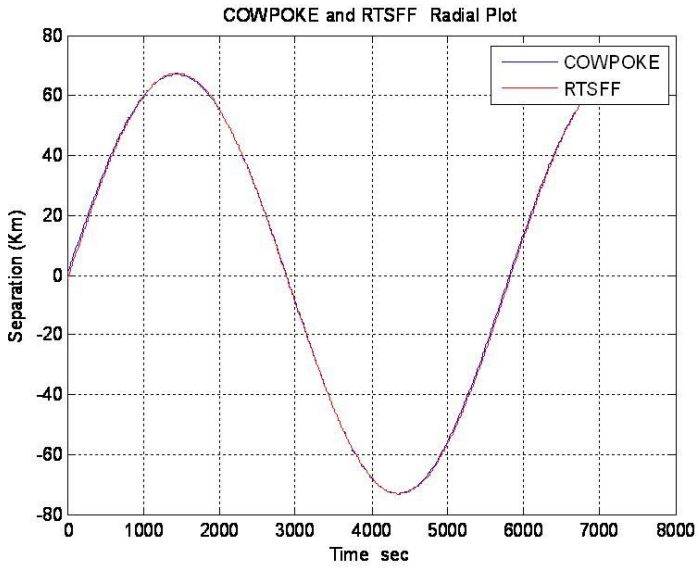


Figure 3.33: Radial separation results for COWPOKE and RTSFF model

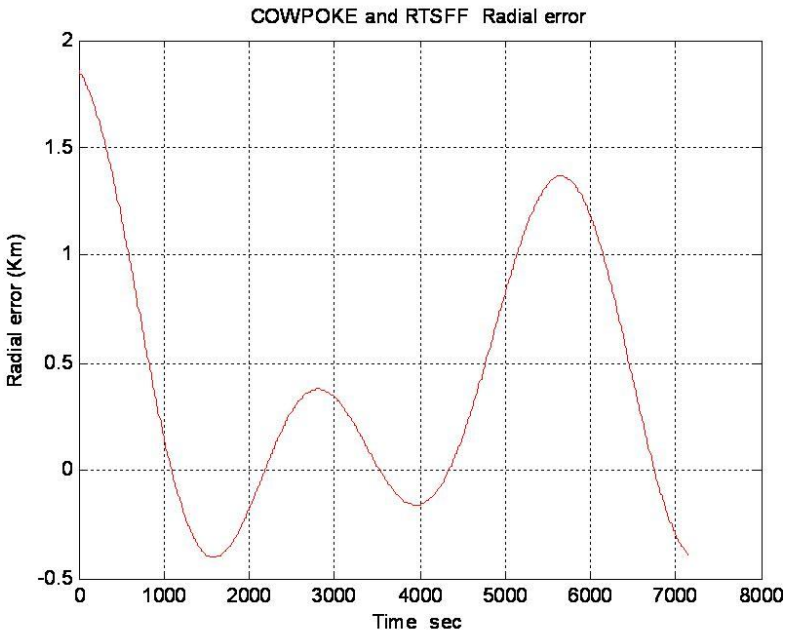


Figure 3.34: Radial separation error between COWPOKE and RTSFF model

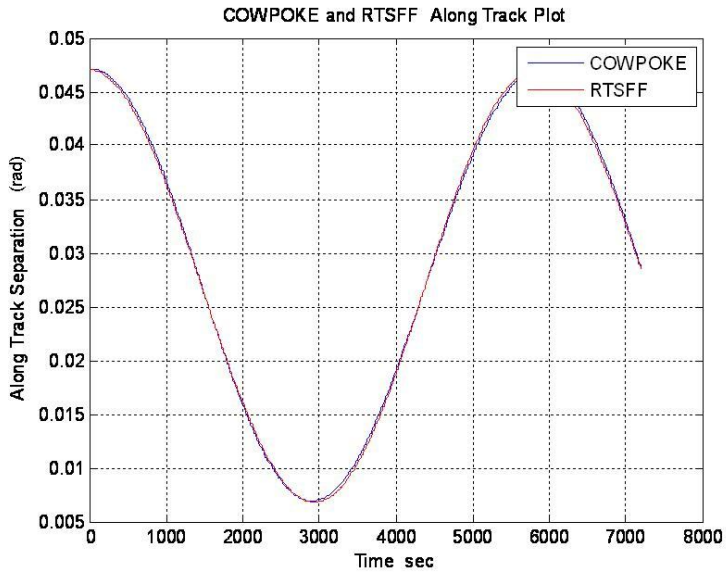


Figure 3.35: Along Track separation results for COWPOKE and RTSFF model

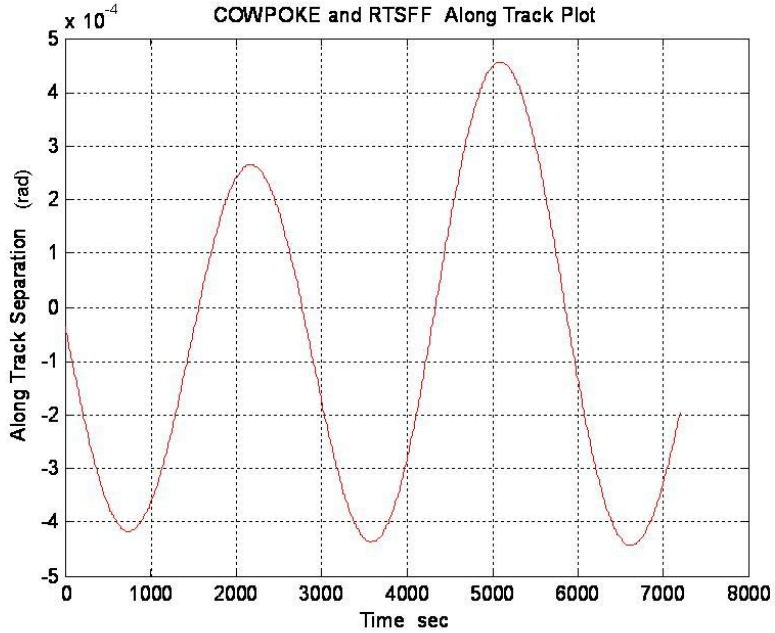


Figure 3.36: Along Track separation error between COWPOKE and RTSFF model

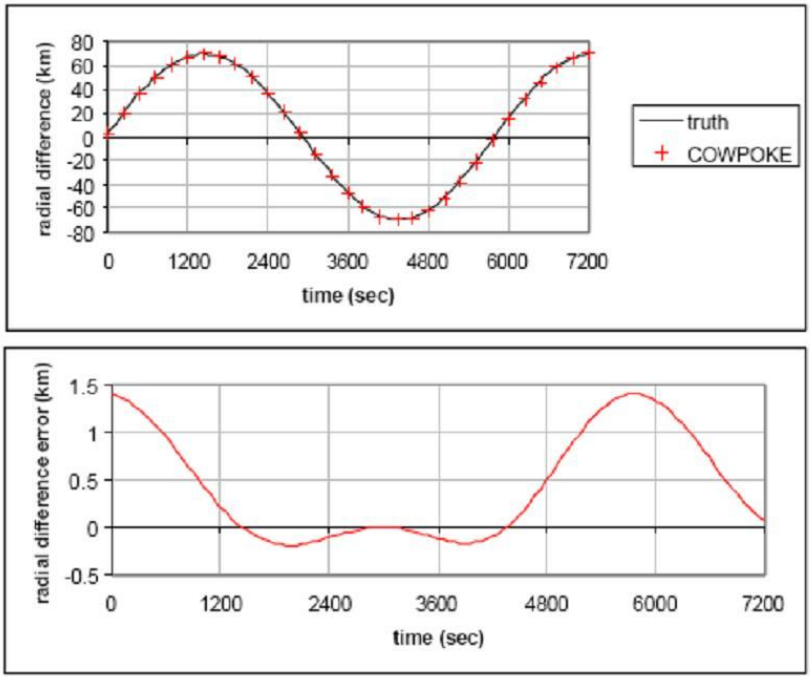


Figure 3.37 : LEO test case Radial Differences and COWPOKE errors

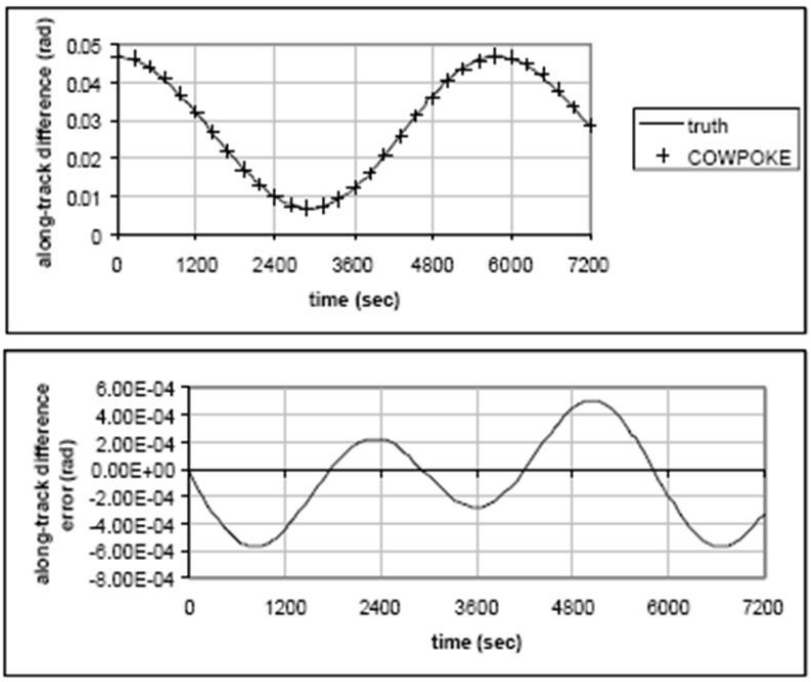


Figure 3.38: LEO test case Angular Along Track Differences and COWPOKE Errors

Simulation of RTSFF Model for circular Orbit with out any perturbation

Table 6: Formation Flying initial Parameters for a perfect circular LEO orbit

Parameter	Master Satellite	Deputy Satellite	Difference
Major Axis a (Km)	7178.1363	7178.1363	0
Eccentricity e	0	0	0
Inclination – I (rad)	0.78539816339	0.78551881102732	0.00691260
RAAN-- Ω (rad)	0	2.5830872929e-8	2.58308729e-8
Argument of perigee-	0	4.712822530293	4.712822530293
Mean Anomaly M (rad)	0	1.570362759956	1.570362759956

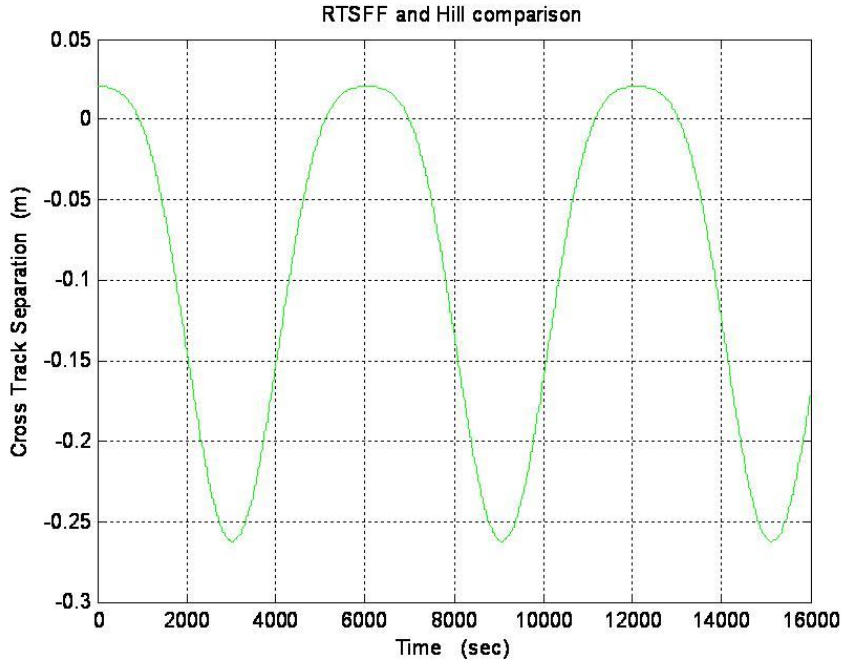


Figure 3.39: RTSFF and Hill's equations comparison for near circular orbit

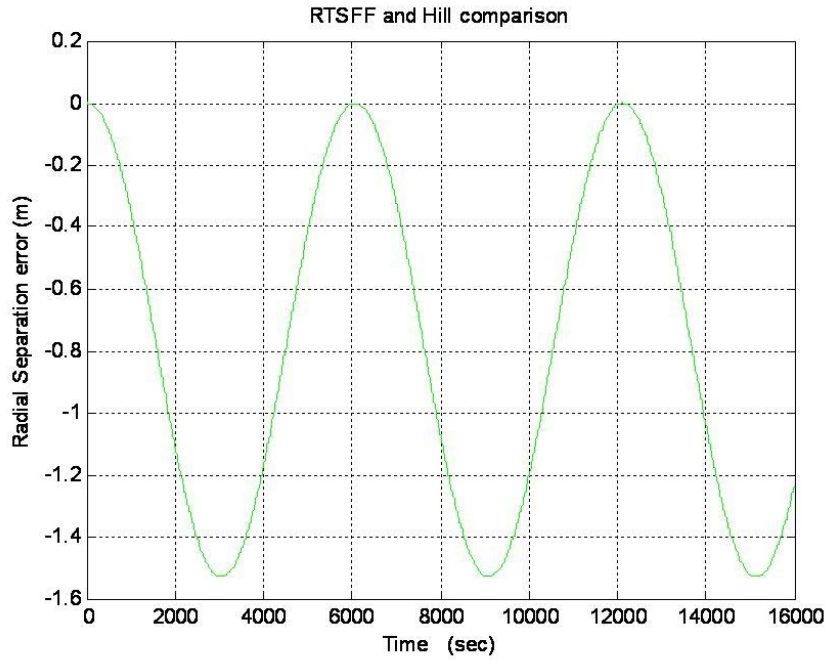


Figure 3.40: RTSFF and Hill's equations comparison for radial separation for near circular orbit

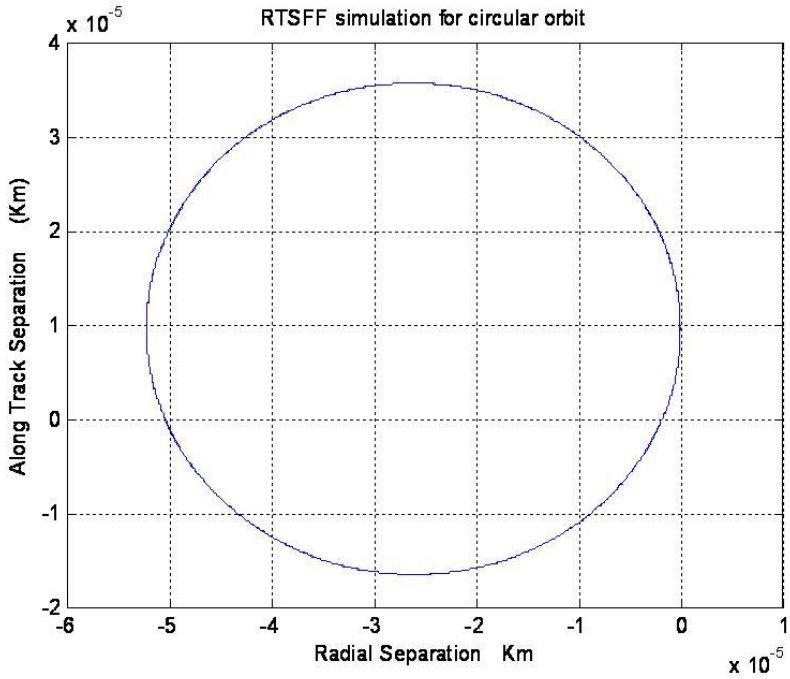


Figure 3.41: Radial against Cross Track separation Plot for RTSFF model for circular orbit

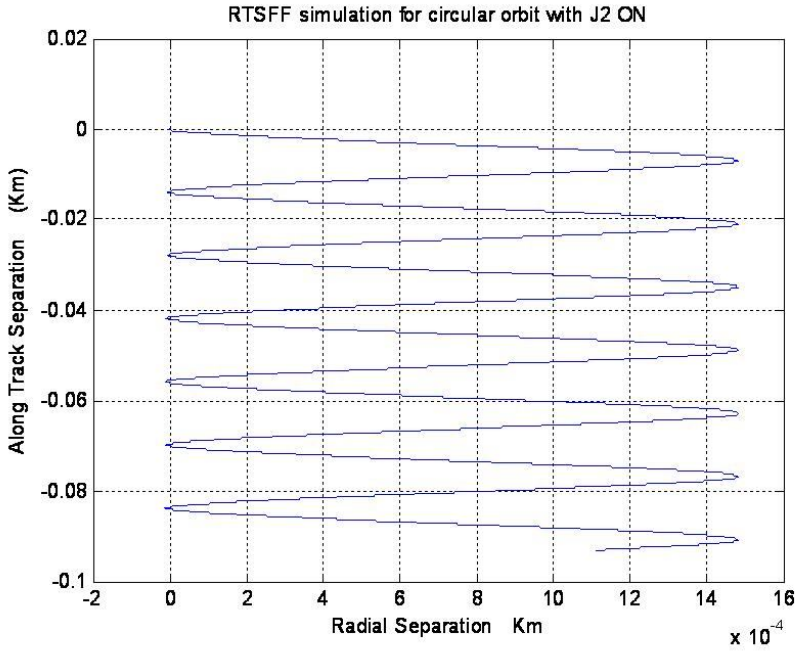


Figure 3.42: Radial against Cross Track separation Plot for RTSFF model for circular orbit with J2 ON

SECTION 2

GPS and Relative Navigation System

Chapter 4

GPS as Relative Navigation Sensor

For satellites in low Earth orbit (LEO) the use of the Global Positioning System (GPS) poses an attractive alternative to other relative navigation sensors in terms of accuracy, robustness, flexibility, and acquisition cost. GPS provides not only accurate relative states (<10m absolute position and 1m/sec velocity accuracy and 2 cm relative separation and <0.5 mm/sec relative velocity accuracy [45]) but also provide absolute states of the satellite. GPS provides highly accurate timing information for onboard time synchronization and enables simultaneous measurements from spacecraft within the formations. With multiple antennas on one vehicle, GPS is capable of attitude determination as well. With these range of capabilities, GPS is a strong candidate for a relative navigation sensor in SFF. At this point it is worthwhile to present a brief overview of the GPS, its observation and its use in space as navigation sensor.

4.1 Global Positioning System (GPS):

The GPS is a constellation of satellites that transmits information modulated on a radio signal. At the time of writing there are 30 satellites in the GPS constellation. These satellites are spread around the world in such a way that at any position on the earth, at least four GPS satellites will be visible to the user. GPS satellites are divided into 6 groups. Each group has a minimum of 4 satellites in the same orbital plane and is inclined at 55 degree with equator. They have 12 hour circular orbits at an altitude of 26,560 Km. Orbits are separated by 60 degrees in ascending node.

GPS satellites transmit messages on two different frequencies, L1 (1575.42MHz) and L2 frequency (1227.60MHz). The information on L2 is encrypted and is restricted to DoD authorized users. For this reason and also because the dual frequency GPS are very expensive, the relative navigation system used in this research is developed around a single frequency GPS receiver. The

Figure 4.1: GPS constellation

L1 frequency contains C/A and P(Y) signals. Only the C/A signal is available for civilian users. Each GPS satellite is assigned a unique PRN (pseudo random sequence no) signature to transmit information. This signature consists of 1023 zeros and ones. This unique identifier serves two purposes.

1. Identification. Unique signature means receiver knows from which satellite, signal is originated.
2. Signal transmits time measurement.

The Following information (navigation message) is transmitted by each GPS satellite at 50 bits per sec.

1. Satellite time and synchronization signals
2. Precise orbital data (Ephemeris Data)
3. Time correction information to determine exact satellite time
4. Approximate orbital data for all satellites (almanac data)
5. Correction signals to calculate the signal transmit time
6. Data on the ionosphere
7. Information on satellite health

The GPS receiver generates the same structure signals as transmitted by the satellites. These locally generated signals are then shifted on the time axis to match with the incoming signals. This time shift Δt on the time axis is the signal travel time and multiplying it with the speed of light will give the pseudo range measurement. Another parameter that is measured is the carrier frequency of the input signal (Doppler shift). These measurements need to be corrected for ionospheric delay, GPS satellite and GPS receiver's clock errors and cycle slip ambiguity.

4.2 Carrier Phase Measurement:

Carrier phase is measured in a carrier tracking loop as a result of a phase lock loop that tries to maintain a zero difference between the received and locally generated signal. The oscillator in a GPS satellite and in a receiver is used to generate timed signals like C/A and P-codes. The phase measurement formed in the receiver is the difference between the phase of local receiver and phase of received signal called as carrier beat phase. The carrier beat phase (in units of cycles) is

$$\phi_{b_j}^{\ i}(T_j) = f.T_j^i + f.(\delta t_u(T_j) - \delta t_{gps}(T_j)) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad 4.1$$

Where T_j is time of reception and T_j^i is signal transit time from satellite j to receiver i. f is frequency, δt_u is receiver's and δt_{gps} is GPS Satellite's clock errors.

The transit time term consists of two parts, the time taken by the signal to travel between the satellite and receiver, with the speed of light (in a vacuum). The 2nd part accounts for the extra time taken by the signals due to travel through ionosphere. This change in speed due to the ionosphere introduces a phase delay term. Hence above equation in terms of metric units can be written as (Chris Rizos, [46])

$$\Phi_{bj}^{i}(T_j) = \phi_{bj}^{i}(T_j)\lambda = \rho(T_j) + c.(\delta t_u(T_j) - \delta t_{gps}(T_j)) - d_{ion} \quad \text{.....} \quad (4.2)$$

Tropospheric delay is the result of atmospheric effects in the troposphere and stratosphere combined (up to 50Km). Tropospheric delay is not considered here as receiver satellites are moving above this height.

The actual measured carrier beat phase differs from the above modeled carrier beat phase by.

1. Firstly the measured carrier beat phase has random noise associated with the measurement process (multipath, receiver noise, orbital error etc). This is contained in an additional terms ε_ϕ.
2. Secondly, when the carrier beat phase is measured in the receiver, the measurement is ambiguous with regards to the number of integer cycles between the satellite and receiver. There are therefore an unknown integer number of N cycles difference between the measured carrier beat phase and above modeled carrier beat phase. The number N is called the integer ambiguity.
3. Thirdly, if the receiver at any time fails to track a satellite's signal and integer ambiguity changes in between, then a cycle slip has occurred.

The model for the measured carrier beat phase therefore includes these additional terms. Cycle slips are usually determined independently during a pre-processing step and hence not considered in carrier beat phase model. The observed integrated carrier beat phase model is (Chris Rizos, [46])

$$\Phi_{bj}^{i}(T_j) = \rho(T_j) + c.(\delta t_u(T_j) - \delta t_{gps}(T_j)) - d_{ion} + \lambda N + \varepsilon_\phi \quad \text{.....} \quad (4.3)$$

The integrated carrier beat phase measurement is shown in figure (4.2) (Chris Rizos, [46])

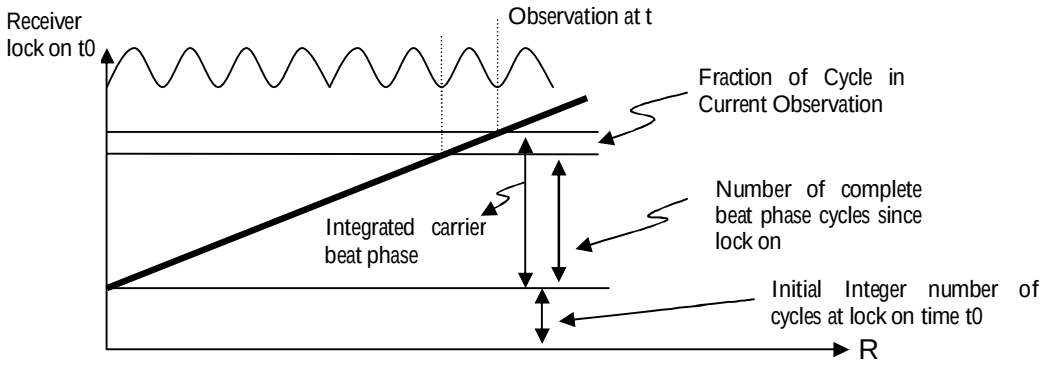


Figure (4.2): Integrated carrier beat phase measurement.

4.3 Pseudo Range Measurement:

Pseudo range is measured in a correlator, where the incoming GPS signal is compared with an internally generated replica. The internally generated GPS signal is dragged along the time axis to match with the received GPS signal. The distance covered by replica signal to match incoming signal gives the information about signal travel time from the GPS satellite to the receiver Δt. Using the analogy of the carrier phase measurement modeling, the pseudo range measurement between satellite i and receiver j comes out to be

$$P_j^i(T_j) = \dot{\Phi}_{b_j}^{\ i}(T_j) = \rho(T_j) + c.(\delta t_u(T_j) - \delta t_{gps}(T_j)) + d_{ion} + \varepsilon_P \quad \ldots\ldots\ldots\ldots\ldots \quad (4.4)$$

It has no cycle ambiguity term and also the sign of ionospheric delay is reversed as ionospheric delay increases the range. Where $\rho(T_j) = \|r - R\|$ is true range, $\vec{r}$ is satellite observation point and $\vec{R}$ is position vector of the receiver. d_{ion} is ionospheric delay. ε_P represent all other errors including receiver noise, loop errors, multipath error, orbital error etc. The geometric range and clock errors with reference to GPS time is shown in figure (4.3) (Chris Rizos, [46])

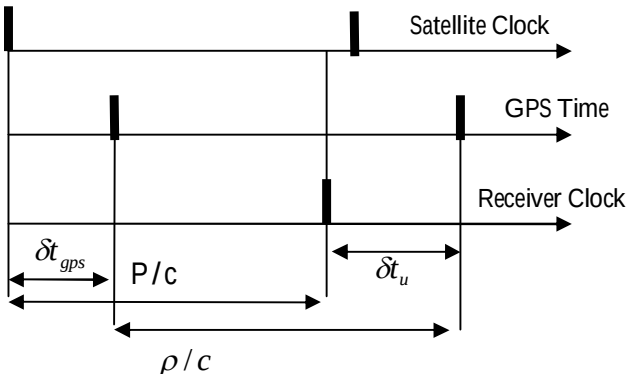


Figure (4.3): Geometric Range and Clock

4.4 Doppler Shift:

Doppler shift is linearly dependent on radial velocity and hence can be used for velocity determination in a real time navigation solution. The equation of Doppler shift can be derived by taking the time derivative of equation 4.3. In that case constant ambiguity term is eliminated

$$D(t) = \dot{\rho}(t) + c(\delta f_u - \delta f_{gps}) - \dot{I} + \varepsilon_D \quad \text{..... (4. 5)}$$

δf_u and δf_{gps} are receiver's and satellite's clock drift rate. $\dot{\rho}(t)$ is range rate, ε_D denotes any other errors, $\dot{I}$ is ionospheric path delay rate.

4.5 GPS Observation Biases and possible solutions:

The observations in receiver are biased due to uncertainties and error. In order to find true absolute pseudo range and accurate relative state vectors, these biases and uncertainties should be removed or at least minimized from the observations. These biases and errors are

a. Ephemeris uncertainties
b. Receiver clock error
c. Carrier phase ambiguity
d. Half Cycle Ambiguity
e. Satellite clock error
f. Ionospheric Delay
g. Cycle Slip

There are different techniques and methods to handle these, e.g

- Those biases which are linearly correlated across data sets can be eliminated by differencing.
- Those biases which are linearly correlated across epochs can be eliminated by differencing between epochs.
- The biases can be considered known, being adequately modeled,
- Data smoothing algorithms.
- Prediction methods, like Kalman Filtering
- Or the biases can be simply ignored.

The ionosphere is the upper part of atmosphere that commences at 85km. The GPS signal consists of different frequency codes modulated on carrier waves. As a consequence of different velocities a group delay and a phase advance occurs. Therefore the pseudo range measurement appears to be too long, while the phase range measurement appears to be too short. The correction term for carrier phase is positive and for pseudo range correction is negative. The amount of difference in both cases is same.

The broadcast navigation message contains a crude predicted ionospheric correction model. This model can reduce the effect of ionospheric delay by up to 50%. Another approach is to difference between two receivers for the same satellite. Effective cancellation is a function of baseline. For short baseline the two signals to two receivers travel through same ionosphere and hence ionospheric delay effectively cancel between the receivers.

The effect of satellite clock error is cancelled by differencing between receivers for the same GPS satellite with the same time tag (Single differencing). Receiver's clock error is considered as unknown and is calculated during Kalman filtering estimation or Least Square Estimation.
Cycle slip occurs, when the receiver loses lock to a particular GPS satellite. It

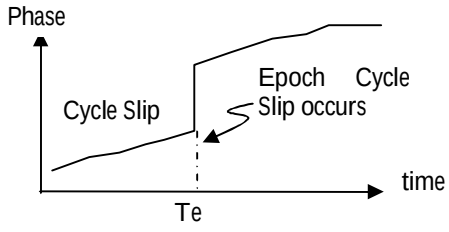


Figure (4.4): The sequence of carrier phase measurement due to occurrence of cycle slip

causes a jump in the observed carrier phase measurement by an integer number of cycles figure (4.4). It is therefore required to correct the integer ambiguity number with the size of the jump.

To detect a cycle slip, some testing quantities are required. There are different techniques to formularize testing quantity for single frequency receiver like, undifferenced approach, single difference, double difference, triple difference, and phase-code range combination. In a phase-code range combination, difference between carrier phase and code range gives testing quantity formula as

$$\lambda\Phi_j^i(t) - P_j^i(t) = \lambda N - 2.d_{ion}(t) \quad \text{..... 4. 6}$$

4.6 Half Cycle Ambiguity:

When a GPS receiver locks to a satellite its initial counter value is either zero or an arbitrary integer number. The final phase measurement consists of the sum of the fractional part (i.e., 0° - 360°) plus the current value of the zero-crossing counter. This phase measurement may be affected by 180° phase reversal (due to modulated C/A code) because the PLL is insensitive to these reversals. When this happens, the phase measurements have inverted polarity with half-cycle errors which result in half cycle ambiguities. Half cycle ambiguity is usually resolved during acquisition and tracking process in GPS receivers, so is not a problem for users.

4.7 Space use of GPS as Navigation Sensor:

The Global Positioning System was meant for ground users. In order to use the GPS receivers for space application, the different signal's dynamic characteristics like Doppler velocity, frequency shift, rate of change of Doppler velocity, bandwidth of GPS signal, no of visible satellites, angle of GPS satellites with the local horizon (along track axis) should be known for LEO satellite, with respect to formation flying frame (RSW). These parameters help to decide about the tracking loop update rate, acceleration limit, selection of SAW input filter, order of phase lock loop models, selection of elevation angle. This information also helps in navigation software algorithm to choose only those GPS satellites with low relative dynamics in order to reduce error due to Doppler shift estimation. A simulation model was developed in Matlab/SIMULINK and different simulations were run with different constraints and

parameters to understand the different dynamic characteristics of GPS signals and the relative dynamic geometry between LEO satellite and different visible GPS satellites. This model can also be used to estimate the positions of GPS satellites in a searching algorithm at GPS receiver's start up.

The GPS transmitting antenna is always pointing towards the centre of earth and the transmitting antenna has a beam width of 21.3^0 . The geometry of the GPS beam and earth is shown in figure (4.5)[47]. Any LEO satellite in the beam and not in the shadow of the earth will receive the GPS signals [47]. Let us define a local frame of reference associated with receiver satellite. The origin of the local coordinate system lies at antenna. X-axis is along the radial direction, y-axis at right angle along the increasing velocity. Z-axis is perpendicular to this x-y plane. yz-plane will define the local horizon plane for the GPS antenna. Therefore the GPS satellite visibility, Doppler velocity and frequency shift analysis is performed with respect to this formation flying frame attached to the master satellite (RSW). The GPS satellites point their transmitting antenna towards centre of earth and main L1 beam makes an angle of 21.3° with the line joining centre of earth. Any LEO satellite within this beam and not in the shadow of the earth will see the GPS satellite. An orbital dynamic model is used to simulate 29 GPS satellites and one LEO satellite in ECEF frame. The position vectors are then transformed into the local frame of the LEO satellite by using a Formation flying relative model. The number of GPS satellites visible to the LEO satellite is recorded, their total Doppler velocity along the Line of Sight is recorded and the Doppler frequency shift is calculated against time and angle. The orbital dynamic model used is given as

$$\ddot{X} = -\frac{\mu x}{r^3}[1 - J_2 \frac{3}{2}(\frac{r_e}{r})^2(5\frac{z^2}{r^2} - 1)$$

$$\ddot{Y} = \frac{y}{x}\ddot{X}$$

$$\ddot{Z} = -\frac{\mu z}{r^3}[1 + J_2 \frac{3}{2}(\frac{r_e}{r})^2(3 - 5\frac{z^2}{r^2})] \quad \text{.....(4. 7)}$$

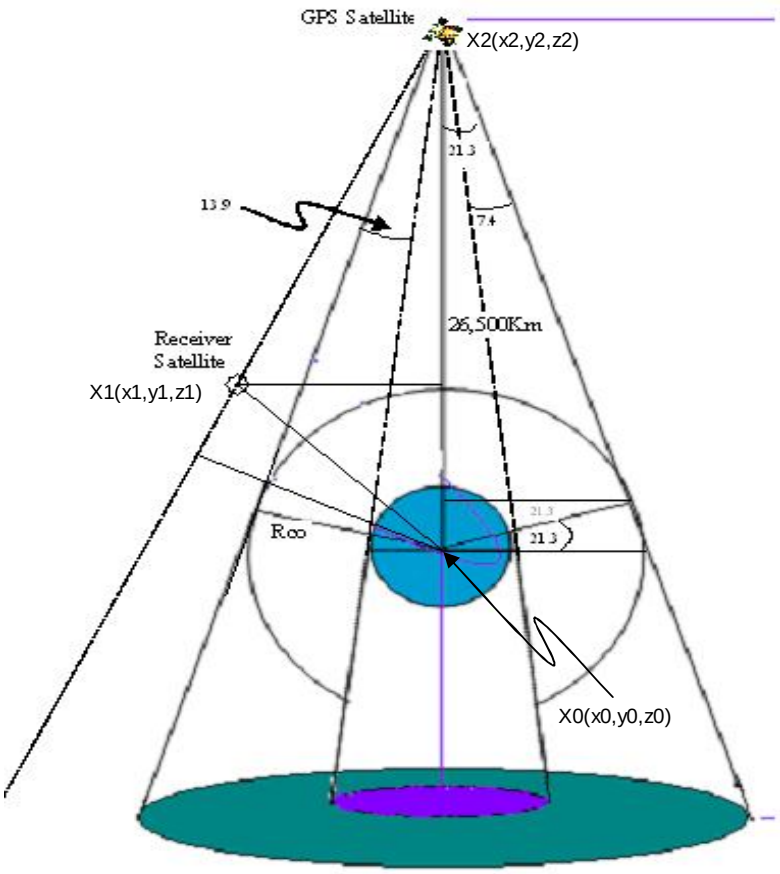


Figure (4.5): GPS beam geometry

This model is programmed in SIMULINK for 30 satellites with different orbital parameters. The orbital parameters for GPS satellites are obtained from[48]. The relative dynamics geometry between GPS satellites and LEO satellite is shown in figure 4.6. It shows the position of GPS satellites, $L\hat{O}S$ (Line of Sight) vector and projected angle with local horizon at different times with respect to LEO satellite. This is shown in two dimensions for simplicity. θ is the angle that GPS satellite makes with local horizon centered at center of earth and ξ is the angle GPS satellite makes with local horizon centered at LEO satellite's receiver antenna.

$\bar{X_1}\bar{X_2}$ is line of sight vector ($L\hat{O}S$) and is given as

$$L\hat{O}S = \frac{O\vec{X}_2 - O\vec{X}_1}{\left|O\vec{X}_2 - O\vec{X}_1\right|} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.8)$$

The angle between GPS satellite position vector R_{gps} and $L\hat{O}S$ vector R_{rg} is β and is calculated by applying cosine law on the triangle

$$\beta = \cos^{-1}\left(\frac{R_{rg}^2 + R_{gps}^2 - R_r^2}{2R_{rg}.R_{gps}}\right) \quad \text{.....(4. 9)}$$

Where R_r is LEO satellite's position vector. If β is less than 21.3° and LEO satellite is not in the shadow of the earth, that GPS satellite will be visible to the LEO satellite.

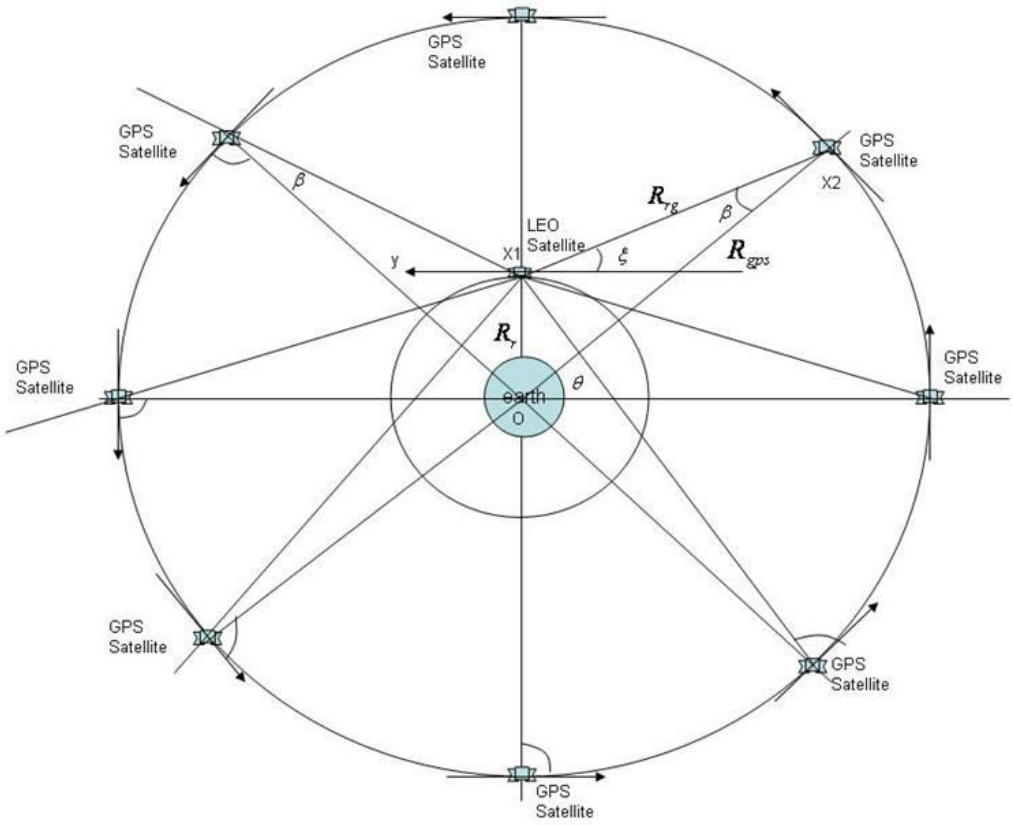


Figure 4.6 Geometrical Representation of LEO Satellite and GPS Satellite relative dynamics

If $\beta < 21.3^\circ$ and $R_{rg} < R_{gps}$, GPS satellite will be visible

If $\beta < 21.3^\circ$ and $R_{rg} > R_{gps}$, but $\beta > 13.9^\circ$, GPS satellite will be visible

If $\beta < 21.3^{\circ}$ and $R_{rg} > R_{gps}$, but $\beta < 13.9^{\circ}$, GPS satellite will be in earth's shadow

A simulation is run for a LEO satellite. To simplify analysis the orbital plane of LEO satellite coincides with orbital plane of group 1 GPS satellites. LEO Satellite's orbital parameters are given as.

Eccentricity:	0.6325244904E-002
Orbital Inclination(rad):	1.04719755119660
SQRT(A) (m 1/2):	2.645751311064591e+003
Right Ascen at Week(rad):	-0.1717062294E+001
Argument of Perigee(rad):	-1.710363487
Mean Anomaly(rad):	0.2741693674E+001

The simulation result for a number of visible GPS satellites is shown in figure (4.7). The simulation result shown that number of GPS satellites visible to LEO satellites are more than 14. For absolute and relative navigation at least 4 GPS satellites should be visible at a time.

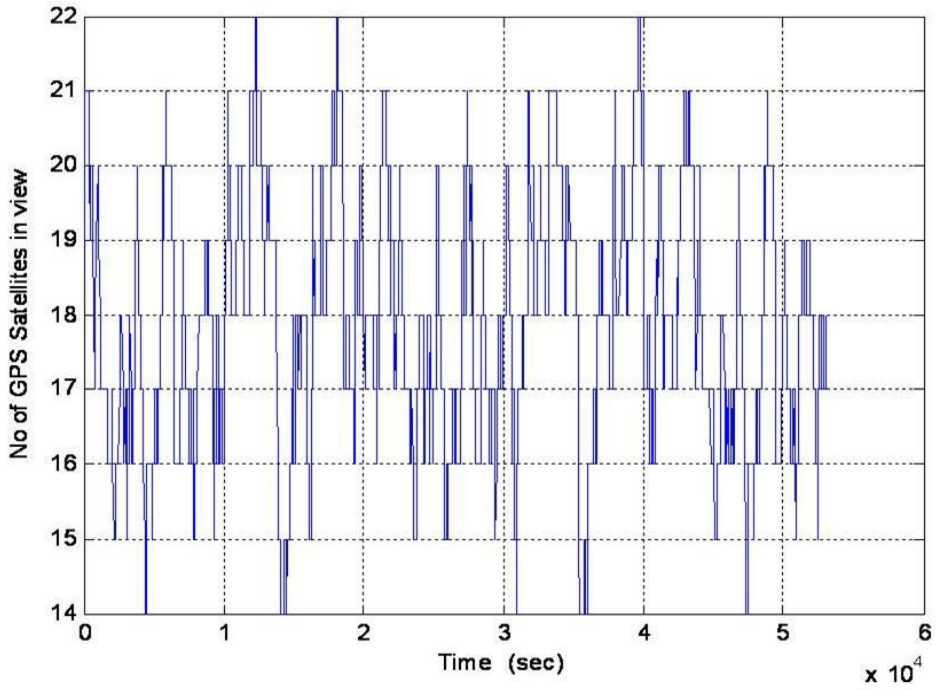


Figure 4.7: GPS Satellites in view to LEO satellite

The position vectors for each GPS satellite is then transformed into LEO satellite's local frame of reference using the following SFF model.

$$x_r = \frac{\vec{r}}{r}\cdot\vec{r}_{GPS} - r, \qquad y_r = (\frac{\vec{h}}{h}\times\frac{\vec{r}}{r})\cdot\vec{r}_{GPS}, \qquad z_r = \frac{\vec{h}}{h}\cdot\vec{r}_{GPS}$$

Where x_r, y_r, z_r are GPS satellites position coordinates in LEO satellite's local frame of reference. This local frame of reference lies at the receiver's antenna. The angle of the GPS satellite with respect to the local horizon (yz-plane) is given as

$$\xi = \arccos(\frac{{y^2}_r + {z^2}_r}{r_{gps}\cdot|y_r + z_r|}) \quad \text{.....(4. 10)}$$

Where r_{gps} is GPS satellite's position vector in the local frame of reference. Now the Doppler velocity due to GPS satellite and LEO satellite along $\hat{LOS}$ vector is

$$\left.\begin{aligned} V_{d1} &= \vec{V}_{rec}.\hat{LOS} \\ V_{d2} &= -\vec{V}_{GPS}.\hat{LOS} \\ V_{dT} &= V_{d1} + V_{d2} \end{aligned}\right\} \quad \text{..... (4. 11)}$$

Where V_{dT} is total Doppler velocity along $\hat{LOS}$. The simulation result for Doppler velocity due to GPS and LEO satellite's dynamics are given in figure 4.8. The Doppler velocity only due to GPS Satellite varies between two extremes 1 Km/sec and -1 Km/sec. The maximum Doppler velocity occurs at $\theta = 15.5^\circ$ and this is the angle when GPS satellite crosses the local LEO horizon. Minimum Doppler velocity occurs when GPS satellite makes 90 degree angle with the horizon i.e GPS satellite comes on top of LEO satellite. Therefore there is an area near the horizon that has high relative dynamics and an area around $\theta = \xi = 0$ that has low relative dynamics. A high dynamic area will produce a large Doppler frequency shift in the L1 carrier frequency. On the other hand Doppler frequency rate of change is low near the horizon and a maximum at $\theta = \xi = 0$. Doppler frequency change rate decides the update rate for

tracking loop. Tracking loop is responsible for adjustment of Doppler frequency shift and alignment of input and locally generated C/A code. Resultant Doppler frequency shift due to Doppler velocity in carrier and C/A signals is given as

$$f_d = \frac{f.v_{dT}}{c} \quad \text{..... (4. 12)}$$

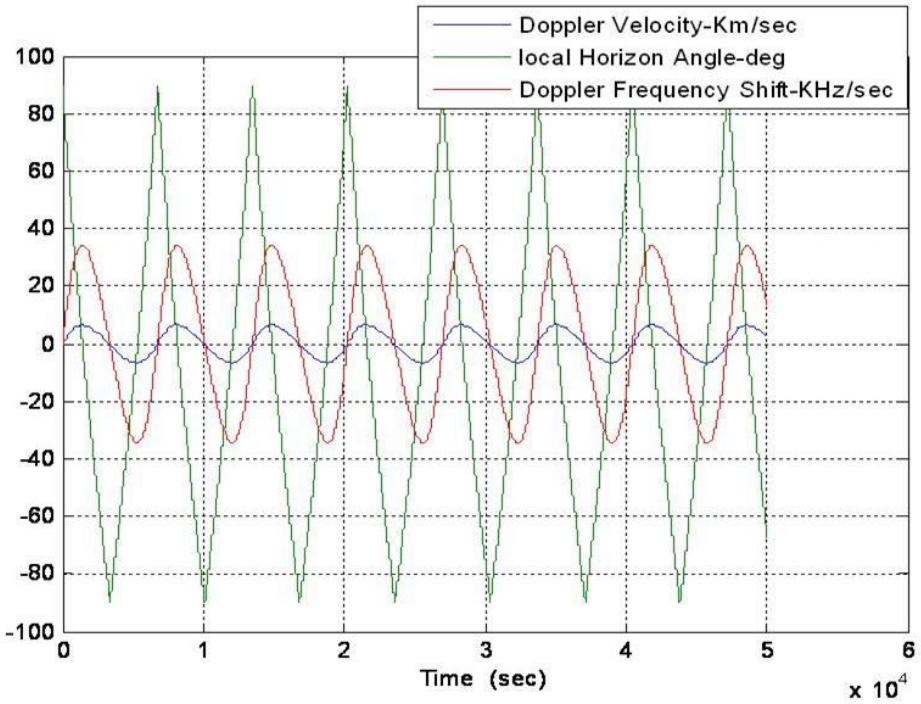


Figure (4.8): Doppler Effect on L1 signal due to Group 1 GPS Satellite

Where f is carrier or C/A frequency, v_{dT} is total Doppler velocity and c is the speed of light ($3.0 \times 10^8 m/\sec$). As the LEO satellite lies in the orbital plane of group 1 GPS satellites, the maximum Doppler velocities and Doppler frequency shift will occur for the GPS satellites in group1. All other GPS satellites will show lower Doppler velocity and frequency shift than group1 satellites.

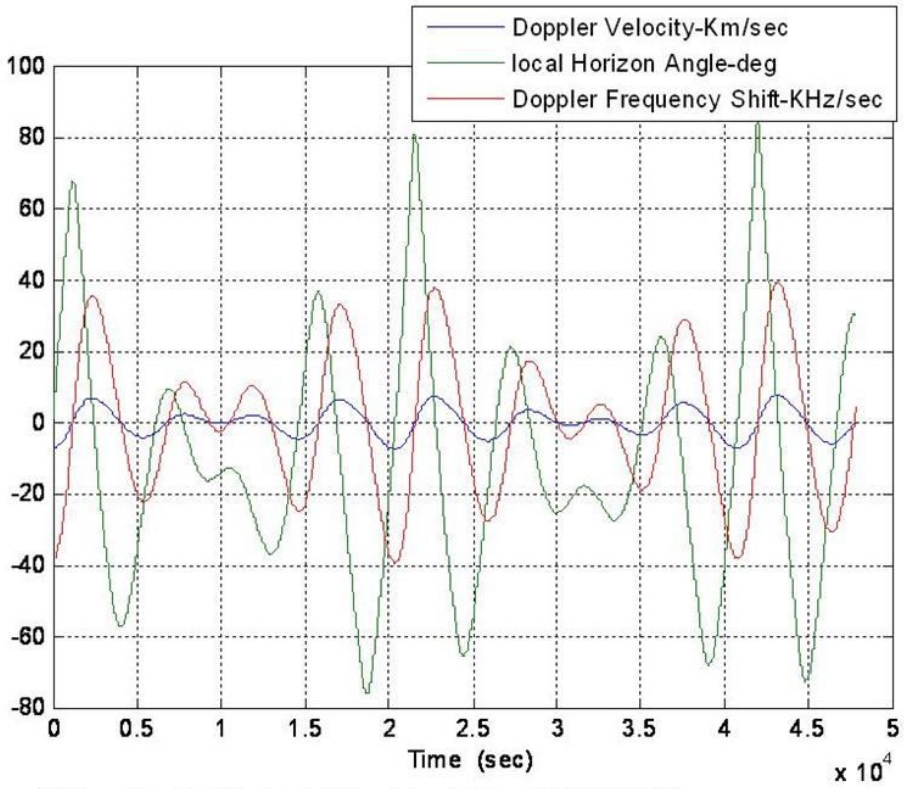


Figure 4.9: Doppler Effect on L1 Signal due to Group 3 GPS Satellite

The Doppler frequency shift determines the searching frequency range during acquisition phase. The simulation results for 'number of visible GPS satellites' and Doppler frequency shift of L1 signal reveals that if a GPS receiver in low earth orbit, whose input searching frequency bandwidth is at least 1575.42MHz $\pm 20KHz$ and whose tracking loop can track L1 signal with Doppler shift of $\pm 20KHz$ it will be able to see and track minimum of 4 GPS satellites. As a minimum of 4 GPS satellites are required for navigation algorithm so this GPS receiver can be used for space navigation purposes. The simulation result for visible GPS satellites with the restriction of $\pm 20KHz$ Doppler frequency shift is given in figure 4.10.

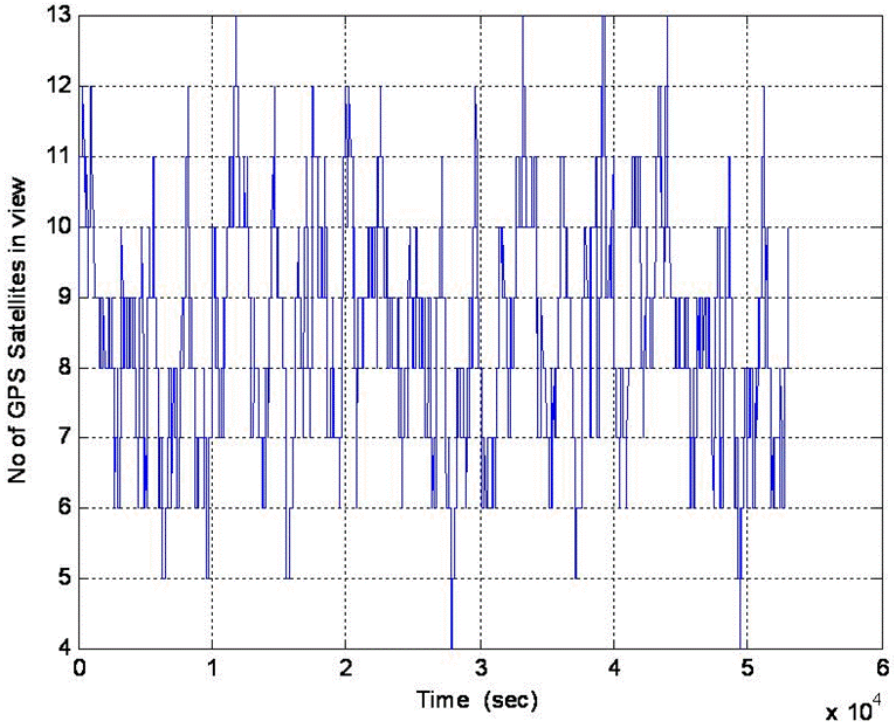


Figure 4.10: GPS satellites in view to LEO satellite with $\pm 20KHz$ Doppler shift restriction

The Doppler frequency shift in group1 satellites is $\pm 33.35KHz$ and average rate of change of Doppler frequency is around 20Hz/sec. If the tracking loop bandwidth is 1Hz then tracking loop must be updated every 50msec (1/20). If LEO satellite moves opposite direction than GPS satellites in group 1, then Doppler frequency shift varies between $\pm 40KHz$, and corresponding Doppler frequency change rate varies from $24Hz/\sec$ to $33Hz/\sec$. During the correction phase of the LEO satellite, its acceleration increases, in that case tracking update loop needs to be high depending upon the LEO satellite acceleration.

The input frequency bandwidth of antenna and SAW filter of GPS are more than 500KHz to few MHz range[49]. The tracking loop bandwidth of the GPS LEA-4T (used in this project) is 210Hz/sec. It can track GPS signals while moving with acceleration of 4g, which is sufficient to track a GPS signals on nano satellite in LEO orbit.■

4.11 Relative Navigation:

In relative navigation, the relative position of deputy satellites and their relative velocities with respect to master frame of reference are calculated. The formation flying geometry is presented in figure 4.11. The GPS receivers provide raw measurements, pseudo range, carrier phase and Doppler measurements. These raw information and ephemeris data are used to calculate absolute and relative states. Relative states are independent of the absolute navigation algorithm. Both absolute and relative navigation algorithms require data from at least 4 GPS satellites. In an absolute navigation algorithm, GPS satellites which are far away from each other are selected for better position accuracy. But relative navigation algorithm does not demand any such requirements, rather it is avoided to use GPS satellites near local horizon, because GPS signal coming from GPS satellite near local horizon travel more distances in ionosphere, which introduces more error. Also the Doppler shift in GPS signal is very high for GPS satellite near local horizon. Before starting the relative navigation algorithm, it is supposed that all errors, corrections have either been eliminated or calculated except receiver clock error. A relative navigation algorithm uses differencing techniques to calculate relative states. Single differencing is the difference of raw measurements between two GPS receivers for the same satellites at the same time tag. It gives linear equation, with four unknowns, Δx, Δy, Δz, and $\Delta \delta t$ where Δx, Δy, Δz are components of relative position vector and $\Delta \delta t$ is the receiver clock error. Single difference for 4 GPS satellites will give four equations with four unknowns. These equations are then solved either by using the Least Square Estimation technique or Kalman Filter Estimation method. The Least Square Estimation is a simple method but it is less accurate. The Kalman Filter Estimation method is more complicated, requires more computational power and also is more accurate. Differencing of Doppler measurement will give equation with 4 unknowns, $\Delta \dot{x}$, $\Delta \dot{y}$, $\Delta \dot{z}$ and $\Delta \dot{\delta t}$. Where $\Delta \dot{x}$, $\Delta \dot{y}$, $\Delta \dot{z}$ are relative velocity components and $\Delta \dot{\delta t}$ is receiver clock rate. Similarly four or more than four equations are required to solve for these four unknowns.

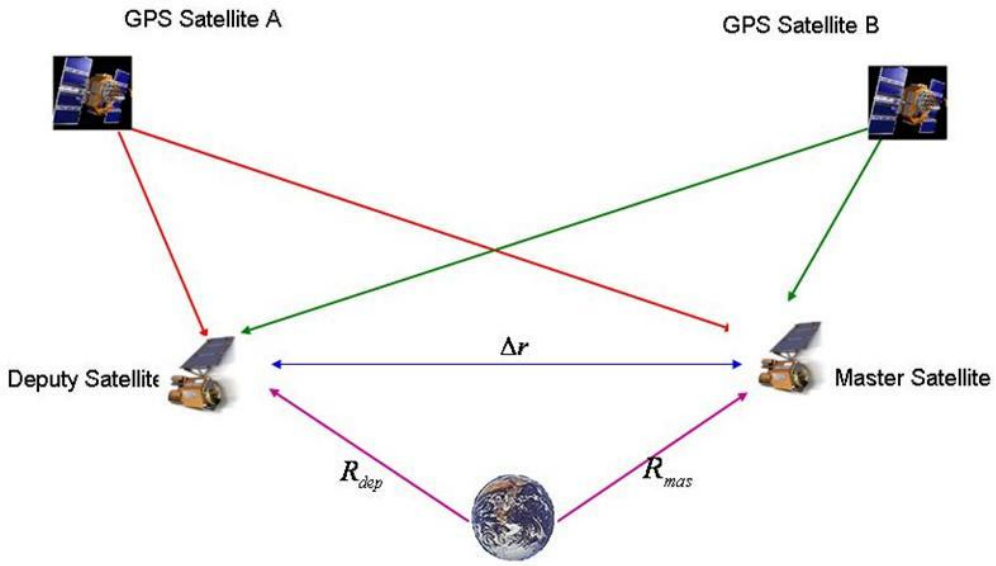


Figure 4.11: Formation Geometry

The Least Square estimation or Kalman filter estimation can be used to solve for four unknowns. Some smoothing techniques have been suggested Montenbruck [11] Egemen Imre [50] to improve the accuracy with Least Square Estimation approach. The use of Kalman filter estimation needs measurement model and relative dynamic model in ECEF frame. The single difference measurement model given by Montenbruck [11].

$$\left.\begin{aligned}
\Delta P(t) &= \left|r_m(t) + \Delta r - r_{gps}(t-\tau)\right| - \left|r_m(t) - r_{gps}(t-\tau)\right| + c\Delta\delta t \\
\Delta\Phi(t) &= \left|r_m(t) + \Delta r - r_{gps}(t-\tau)\right| - \left|r_m(t) - r_{gps}(t-\tau)\right| + c\Delta\delta t + \Delta N\lambda \\
\Delta D(t) &= e_{dep}^T \Delta\dot{r}(t) - e_m^T(\dot{r}_m(t) - \dot{r}_{gps}(t-\tau)) + c\Delta\delta f
\end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots (4.13)$$

Where c (m/sec) is speed of light

r_m, r_{gps} master and GPS satellite's position vector in meters in ECEF frame

Δr is relative distance between master and deputy satellites

ΔP, $\Delta\Phi(t)$ is difference in pseudo range and carrier range respectively in meters

$\Delta D(t)$ is range rate (m/sec)

δf is clock drift rate . It is unit less term

$$e(t) = \frac{\vec{r}(t) - \vec{r}_{gps}(t-\tau)}{\left| r(t) - r_{gps}(t-\tau) \right|}$$

The relative dynamic model in ECEF frame is given by Franz D Busse [45]

$$\Delta\ddot{r}_{ij} = \frac{\mu}{r_m^3}\left[r_m - \frac{r_m^3(\vec{r}_m + \Delta\vec{r})}{\sqrt{(r_m^2 + 2\vec{r}_m\Delta r + \Delta r^2)^3}} \right] + 2\vec{\omega}_e \times \Delta\dot{\vec{r}} + \vec{\omega}_e \times (\vec{\omega}_e \times \Delta\vec{r}_{ij}) + w_{\Delta r} \quad \ldots\ldots\ldots (4.14)$$

Where ω_e is earth's rotation rate, μ is earth's gravitational constant, $w_{\Delta r}$ is process noise due to other forces and higher order gravity terms. Kalman filter initialization and its gain calculations are bit tricky and improper initialization may cause divergence. Double differencing is the difference between raw measurements between receivers and between GPS satellites with the same time tag. It eliminates errors in a better way, but it gives correlated equations, which is difficult to solve and need more computational power. A more detailed description of differencing techniques is given at the end of Appendix G. It is shown that both double differencing and single differencing gave same accuracy to relative navigation algorithm by Franz D Busse [51] In this project single differencing techniques are used. It is supposed that satellites are closely placed so that the GPS signal approaching to them are traveling through same environmental conditions. Now single differencing will cancel out GPS satellite's clock error and ionospheric correction terms.

4.12 GPS Simulator:

The satellite navigation receiver development and testing is a complex job. It requires ability to control and repeat specific GPS RF signal characteristics (based on GPS receiver's dynamics) for development and validation of navigation algorithms. A GPS signal simulator provides this flexibility. The GPS simulator should have two or more than two outputs to simulate two or more than two GPS receiver's signals.

A GPS simulator can be hardware based or a numerical simulator. The main components of a numerical GPS simulator are, Vehicle Dynamics, GPS receiver, GPS constellation. The Vehicle dynamics components simulate the orbital dynamics of the satellites in the formation. Instead of using an orbital propagator for GPS constellation, most of the numerical simulator uses almanac data to simulate GPS constellation. A software version of GPS receiver receives information from GPS constellation simulator and from vehicle's dynamics and generate out put information either in NMEA or binary format. A Numerical GPS simulator eliminates the need for an actual GPS receiver during development and validation process. It is less expensive (100USD to 700USD), less accurate than the hardware based GPS simulator, but good for initial navigation software development in lab environment. GPSLab is one of the examples of numerical GPS simulator. Hardware based GPS simulator only simulate vehicle's dynamics and GPS constellation. It calculates navigation messages, modulates it on carrier frequency and broadcast as RF signal.

In this project hardware is designed in such a way that both numerical and hardware based GPS simulator can be connected for navigation software development and validation. It is recommended to use numerical simulator at first to initially develop and validate navigation algorithm. A details description on these two options is given in chapter 6.

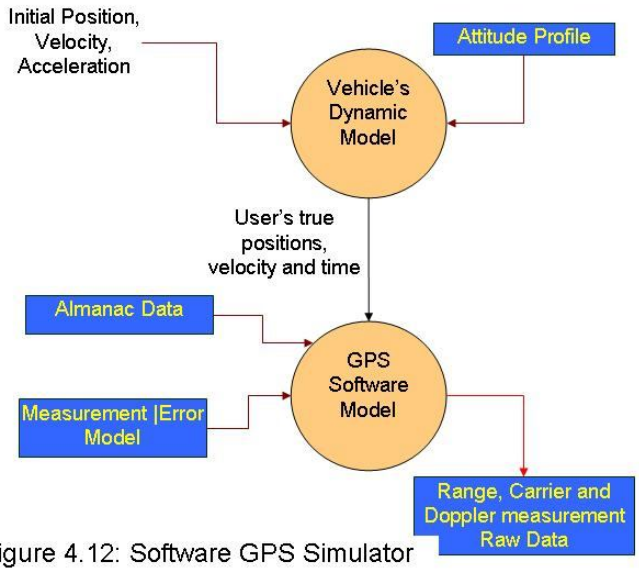
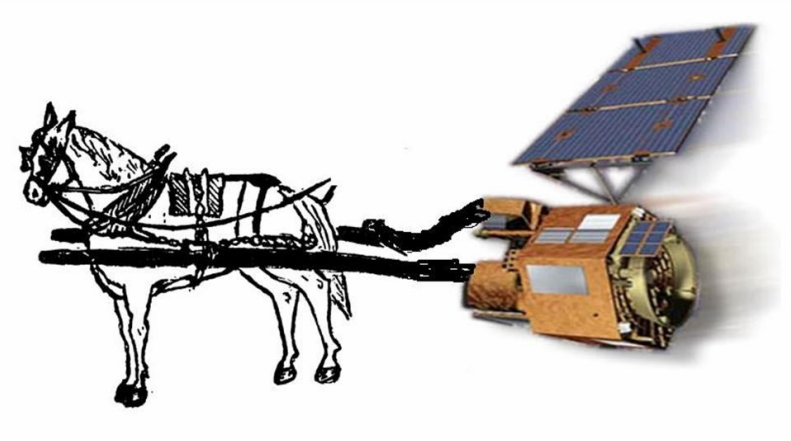


Figure 4.12: Software GPS Simulator

It was therefore decided to develop a low cost real time hardware in the loop SFF simulation test bed, which provides not only hardware components level validation but also provide a platform for relative navigation software development in real time. This can consider SFF orbital dynamics in the navigation software development and can accommodate real hardware components in the loop. Specific flight scenarios can be run for repetitively for software development purposes that can provide not only real relative navigation system for navigation software development, but also provide a PC platform for quick initial assessment of the relative navigation algorithm.

Relative Navigation System



Chapter 5

Relative Navigation System

5.1 Introduction:

For a LEO satellite formation a GPS poses an attractive alternative for navigation sensor in terms of accuracy, robustness, flexibility, and acquisition cost. GPS provides highly accurate timing information for onboard and across the board time synchronization and enables simultaneous measurements from spacecraft within the formations. Due to the advantages associated with GPS as a relative navigation sensor, a single frequency GPS based relative navigation system was chosen for this project. Radio communication was selected as a mean to exchange information between the satellites. A brief summery of the relative navigation system is presented here and more detailed description is given in the following chapter.

In this project it was intended to develop a real time relative navigation system for a test bed which should be cost effective, reliable, and can easily modify relative navigation algorithms. The user can choose different simulation and test options on it. The GPS (LEA-4T) was selected as the navigation sensor and radio transceivers (AC4868) were selected as the data transfer device between satellites. LEA-4T gives raw information (Pseudo Measurement, Carrier Phase and Doppler shift) at a rate of 10 Hz and also gives ephemeris data for each locked GPS satellite. As soon as raw data and ephemeris data is available, the GPS receiver interrupts the microcontroller (S3C2410). Microcontroller receives these binary data and extracts, organize information for each visible satellite. The master satellite transmits its raw information and time tag to the deputy satellite. The deputy satellite calculates the time of transmit and other correction terms and then takes the single difference for the raw information for common satellites between the GPS receivers. An embedded Real Time operating system uC/OS-II is used and 1PPS (1 pulse per second) from GPS receiver is used to tick embedded operating system, hence all the time activities in navigation processor are synchronized with GPS time. Hardware is designed in such a way that the navigation system can be used for different formation flying configurations like, Leader-Follower control (decentralized approach), Master-Slave control (centralized approach), and distributive control etc. It can be connected to GPS Simulator and also to software base GPS simulator like GPSLab. GPSLab generate raw information and this information can be fed directly to the

microcontroller through serial port bypassing the GPS receiver. Software has the option either to run the simulation by capturing live data from the GPS receiver or to run the simulation from the recorded data. The second option is especially very important during the navigation algorithm development and when there is limited access to the GPS simulator. In this chapter the hardware design of relative navigation system and brief description of main components are given. Relative navigation software development, its structure and flow diagrams are also presented.

5.2 Main Components of Relative Navigation System:

The relative navigation system consists of three main functional blocks, GPS, navigation processor and radio modem. Here a brief description is given of these components and the reasons for their selection.

5.2.1 GPS - LEA-4T

A single frequency, 16 channel GPS receiver LEA-4T from U-Blox is selected for this project. It is very lightweight , low cost and consumes very low power. Its input antenna has 3MHz bandwidth and it can acquire signals while moving at 4g acceleration. The tracking loop bandwidth is 210 Hz/sec, is sufficient to track a GPS satellite, and to accommodate the satellite accelerating at up to 4g when thrusters are used. It provides raw measurements (pseudo range, carrier phase and Doppler shift) at a rate of 10 Hz. Ephemeris data are transmitted as soon as it locks to some GPS satellite. The navigation step is set to 1 sec, hence extra raw data can be used to improve data quality and for error correction. It generates 1PPS signal, which is used to synchronize all the time activities in the navigation processor with GPS time. It has one serial and one USB port and both are interfaced to the navigation processor for data transfer. At present only the serial port is being used for data transfer and the driver development for USB is left for future work.

5.2.2 Navigation Processor

For the microprocessor's selection many processors were studied, including the ARM7 based processors AT91SAM7S-XXX, PIC microcontrollers, Philiphs LPC21XX series, and ARM9t based Samsung's S3C2410 processor. Among them S3C2410 was considered the

best suited to the requirements. It was fast (203MHz), 3 serial ports, two USB device port, and one USB host port. It has 272 pins and is based on the 32 bit ARM9t RISC structure. Its interrupt handling and priority setting scheme is very efficient. Its on chip clock generator capability is excellent. It allows stack development for the tasks in supervisor mode and interrupt stacks in IRQ mode. This scheme helps to reduce multiple access and requirement of RAM. It allows SVC trap, which is used by the RTOS (Real Time Operating System) for task switching. A lot of literature and development help is available on Samsung's site. It supports in circuit programming facility. It has 8 input A/D channels and more than 24 digital I/Os. It supports the use of RTOS Kernel. It also has the option to tick one of its timers with IPPS signal from the GPS receiver. Many C compilers are available to develop software around ARM9 processor. One serial and host USB port are interfaced to the GPS receiver and other two serial ports are interfaced to the radio modem and interface computer. Embest IDE software with GNU C compiler is used here for software development and Online Flash Programmer is used for in-circuit flash programming.

5.2.3 Radio Modem:

A GPS is used as the navigation sensor and for the relative navigation algorithms, LEO satellites need to transfer their GPS raw information between themselves. Hence a wireless communication is required. A radio modem is used for this purpose. The selection of the radio modem is very important for real time data transfer and also to decide the formation flying configuration (centralized or decentralized). It is also important for the design of the relative navigation system, so that they can be used for different formation flying configurations. The AC4868 radio modem is used here. This has a small form factor, consumes very low power (250mW) [52] and can work in full duplex mode. At maximum power it has a 15 Km range and it can be programmed for any lower range by varying power options in EEPROM. It has 256 bytes FIFO, which reduces the chance of data losses. It uses frequency hopping techniques to reduce interference at a particular frequency. As the purpose of this test bed is to test satellite formation flying with different configurations, a radio modem should be able to support different satellite formation flying topologies. Due to frequency hopping, the AC4868 radio modem not only gives highly reliable data transfer in real time but also supports different formation flying topologies by using address mode and broadcast mode. These formation flying topologies, like data transfer from Master to one deputy, Master

to many deputy (decentralized), among the deputies (distributed), multi point to point (Centralized) are shown in the figure 5.1. It is also possible to form many independent formation flying groups within the large satellite constellation. Each group has its own radio communication network. Any communication within one group will be invisible to other groups and one satellite can be part of many groups and can communicate to different groups by changing channel no and system ID in real time.

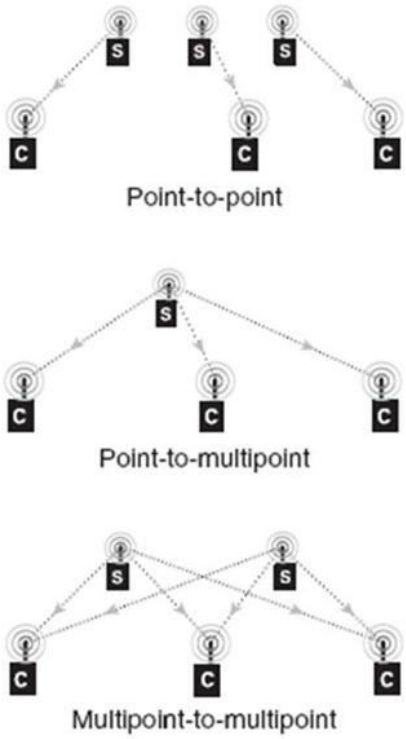


Figure (5.1): Different Formation Flying control strategies

5.3 Hardware Implementation:

The block representation of the relative navigation system design is presented in figure 5.2 The GPS receiver is interfaced to the navigation processor through the serial port UART1 and USB port. 1PPS signal is fed to timer 1, which is then used to tick the real time operating system. For debugging purposes the GPS receiver can be interfaced directly to the

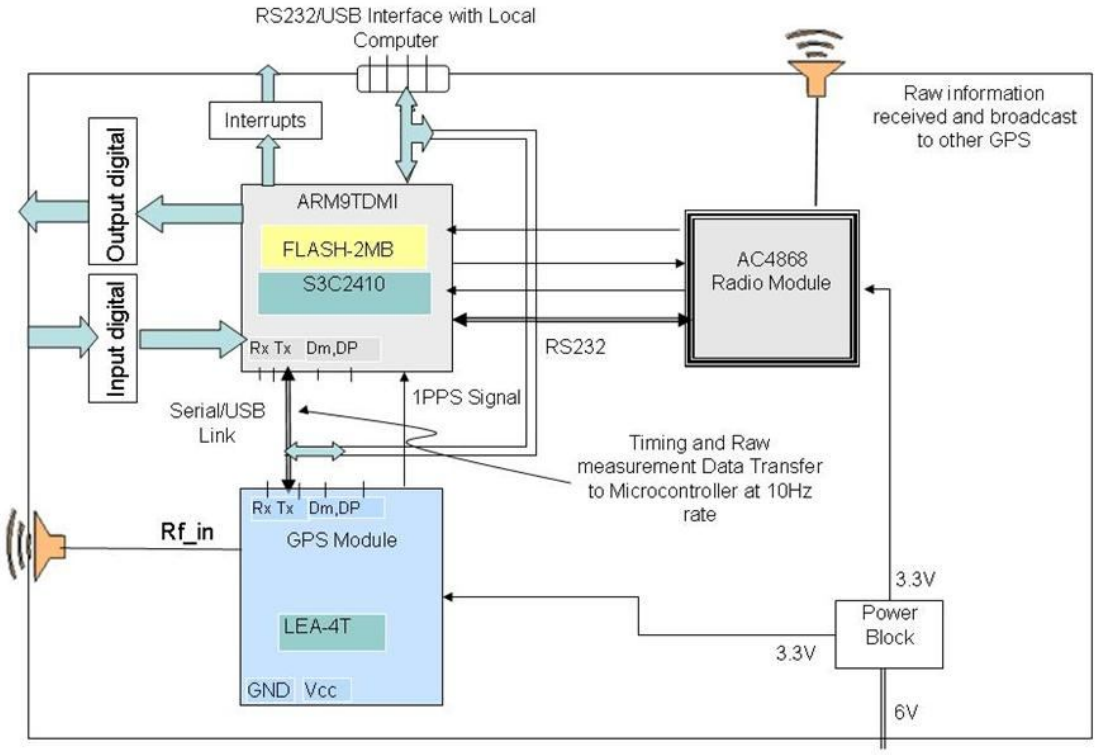


Figure 5.2 A Block diagram of Relative Navigation System design

interfacing PC through serial port by using jumper JP4-2 and JP4-3, and JP4-1 (figure 5.3) is used to interface USB with navigation processor. Similarly GPS receiver can be by passed and raw information could be fed to navigation processor directly from the interface computer running a GPS simulator. The radio modem is connected to serial port UART0 of the navigation processor. To control the programming and data transmit mode of AC4868, GPG15 of S3C2410 is connected to COMMAND/DATA pin of AC4868. GPG7 pin is used to reset the modem in real time. Resetting the transceiver loads the values stored in the EPROM. The third serial port UART2 of S3C2410 is connected to external interface computer. This port is used to configure the parameters of GPS receiver, radio modem and to choose different simulation options. This port is also used to log and display the real time navigation results in the external computer. The whole circuit uses 3.3V power supply. MAX3232 (A serial interface IC to connect 3.3V serial port to 12V PC serial port) is used to interface S3C2410 serial port to computer's serial port. Buffered eight digital input and 8 digital outputs are available to connect to other instruments. Three 5V tolerant interrupt lines are available for external events.

The processor module contains 2MB flash and 64MB SDRAM and is running at 203MHz processing speed. Flash can be programmed in the circuit by using the JTAG port, which helps to modify and load navigation program quickly. The navigation processor is receiving raw information at 115200 baud rate with a 10Hz update rate and sends raw information to the radio transceiver at 57600 baud rate with a 2Hz data transfer rate. The RF link is working at 28.8K baud rate. In case of malfunctioning of radio transceiver, the navigation processor can reset it by pulling its reset pin.

If the deputy satellite goes out of the range from the master satellite or does not receive any signal from the master satellite for a specified period of time, it will inform the navigation processor by pulling the high IN-RANGE pin. The navigation processor then can take the necessary action to resume communication with the master. The navigation processor can bring the radio transceiver to program mode in real time by pulling its command/Data line low. By programming the channel number and system ID parameter, the relative navigation system can be switched from one network to another network of satellites. Each transceiver can be programmed either as a deputy or server and its status can be changed at any time during the simulation. The GPS module is mounted in such a way that its RF (Radio Frequency) side does not face any digital circuit in order to avoid the interference of RF from the digital circuit. The radio transceiver's frequency band is 868-870MHz and the band of next harmonics is close to the GPS L1 frequency so any overlap of the GPS antenna cable and radio transceiver's antenna cables is avoided to reduce interference on the GPS signal. The GPS antenna and RF antenna is mounted 25cm away from each other. RF parts of both modules are placed at opposite sides of the PCB board. The block diagram of the PCB (Printed Circuit Board) layout is shown in figure 5.3. This block diagram shows the different jumper locations on the board. The schematic and PCB layout was designed by using Eagle software. Module libraries for GPS and Radio modem were developed as they were not standard IC modules.

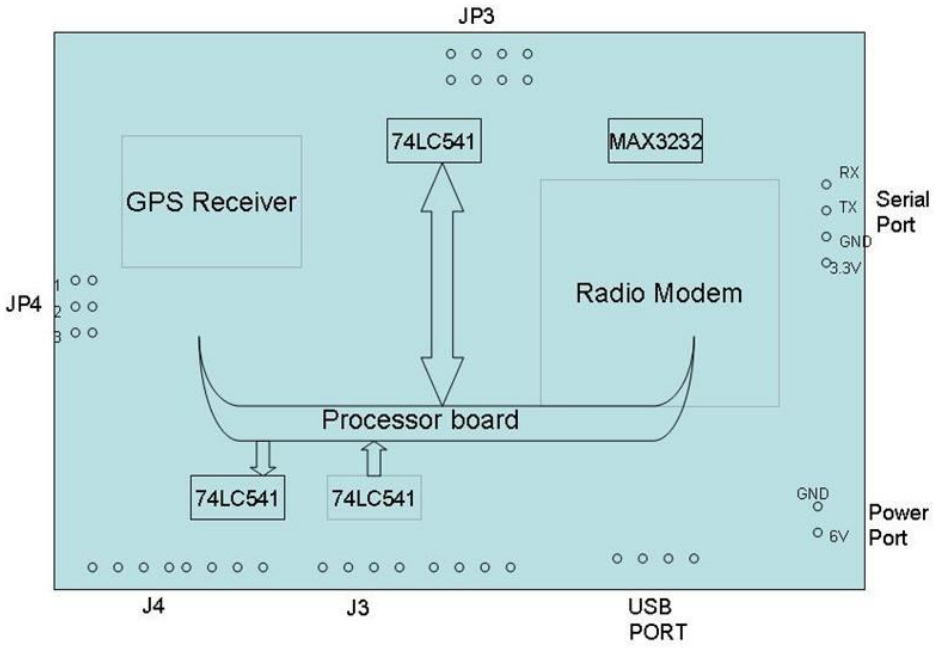


Figure 5.3: Relative navigation board block diagram and jumper locations

Care has been taken that GPS RF and radio transceiver's RF sides do not face each other. Proper ground layers are used under GPS and radio device to absorb high frequency interference. The GPS antenna is soldered directly to the GPS soldering pad in order to avoid any reflections due to impedance mismatching of the PCB tracks. The developed relative navigation system along with their casing is shown in figure 5.4 and 5.5. Figure 5.4 shows the GPS side, where the GPS module and radio modem can been seen. The figure 5.5 shows the complete relative navigation system with the casing. Two antennas (GPS and radio antenna) are mounted apart from each other to avoid any interference. All interfaces (serial, digital I/Os, antenna connector, power port programming port) are provided at the casing. The schematic drawn in Eagle is presented in Appendix B. In order to avoid any frequency interference and track reflections, the GPS antenna is soldered directly to the GPS soldering pad on the PCB.

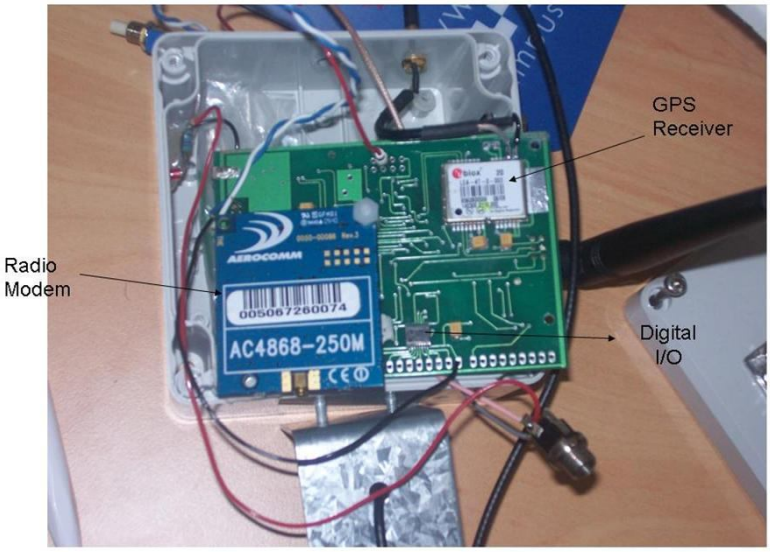


Figure 5.4: PCB Board of Relative Navigation System

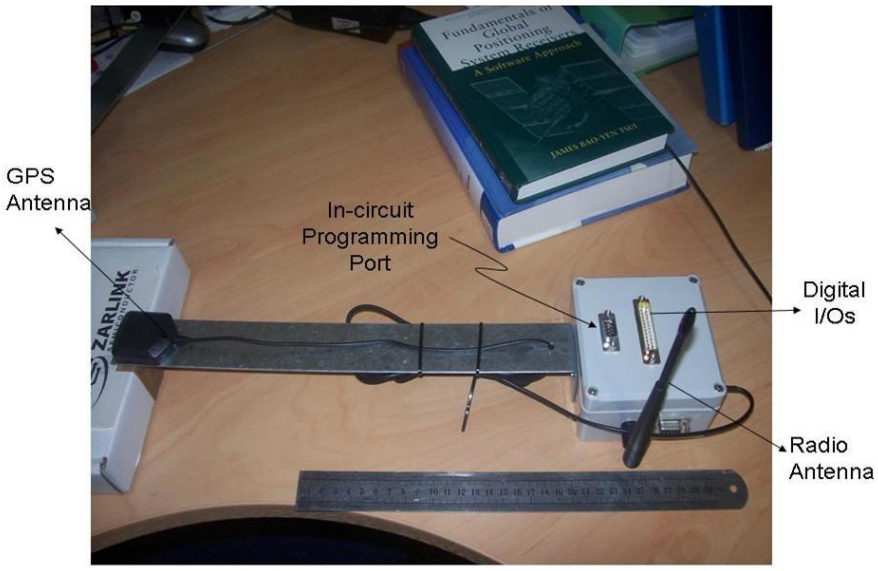


Figure 5.5: Complete Relative Navigation System

5.4 Power Management:

Power management is a very important factor of the satellite's onboard circuitry especially for nano satellites, where available power is very limited. This relative navigation system consumes very little power and many of its peripherals can be sent to sleep mode when

they are not in use. The transceiver at maximum range (15 Km) consumes 250mW power. This power can be reprogrammed in real time according to the range. In sleep mode it consumes only 24mW power. The GPS receiver consumes 117mW power at maximum. The navigation processor S3C2410 consumes at maximum 660mW power. Total maximum power consumption is around 1Watt. The S3C2410 has several power management schemes and can optimize the power according to the task requirements. The power management block can switch off unused peripherals or can slow down the clock to optimize power. In the present design USB host and device ports, port C, D and E, ADC port, SDI, SPI, I2S and NAND Flash controller are not in used and are switched off by disabling the clock to these peripherals in the CLKCON register.

5.5 Memory Management:

The processor module consists of 2MB flash memory and 64MB of SDRAM. Flash memory is used to store navigation code permanently and code written on it cannot be changed by the processor alone. For the navigation code 2MB space is sufficient and at present the navigation code including the RTOS is about 670KB. SDRAM has been divided into different portions and different portions have different uses as shown in figure 5.6. At boot time code written in the flash is loaded in the SDRAM downloadable memory. A non cacheable area of 48MB is used to store recorded raw information during the real time simulation. This recorded raw information is then uploaded to the external computer for further simulation or to repeat the simulation with recorded data.

A summary of the main features of the relative navigation system is as follows

- ★ A single frequency GPS based navigation system
- ★ 203MHz, 32bit ARM9 RISC processor,3.3 volt
- ★ 2MB Flash memory, 64 MB SDRAM

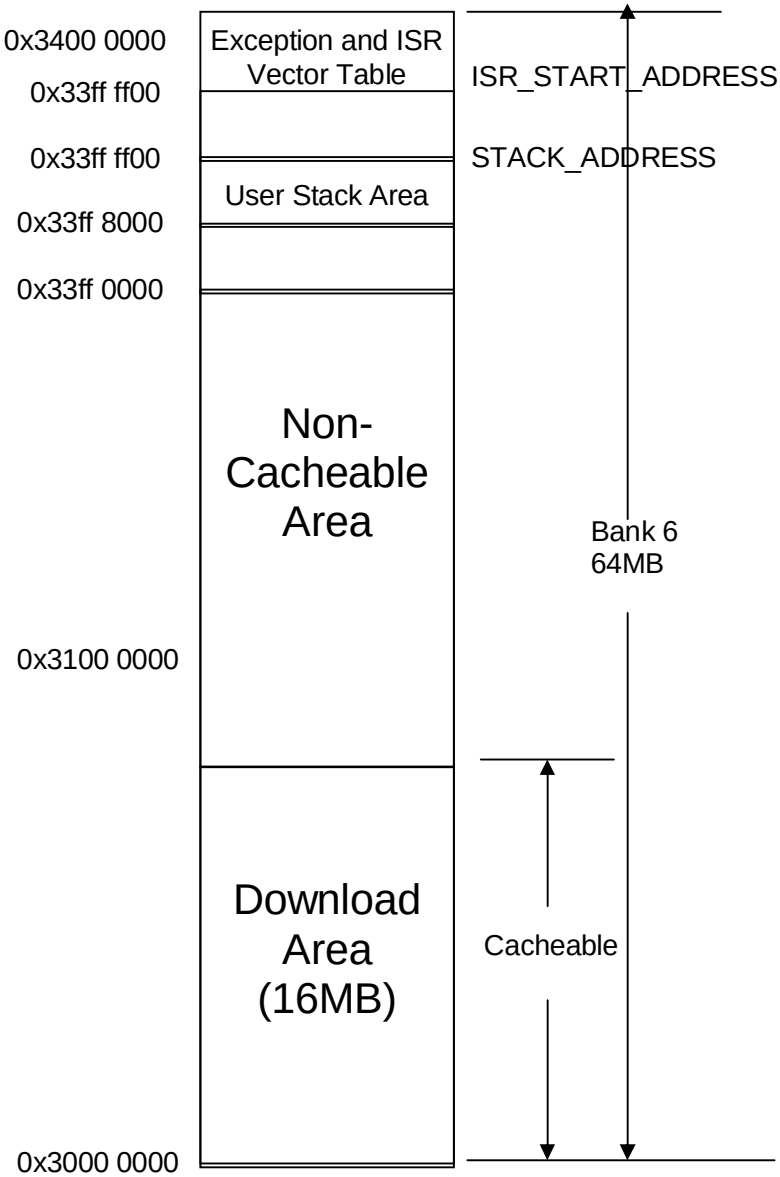


Figure 5.6: Navigation System Memory Map

- ★ uC/OS-II RTOS is used, which is robust, reliable, preemptive real time, multitasking operating system.
- ★ Power consumption is around 1Watt.
- ★ 8 Digital inputs and 8 digital outputs, three external interrupts
- ★ Maximum base line separation between satellites is 15Km.
- ★ It supports centralized, decentralized and distributed formation flying topology
- ★ Relative and absolute navigation system for nano satellites with thrusters having acceleration less than 4g

- ★ Compact, light weight and consume very less power.
- ★ All time activities are synchronized with GPS time
- ★ Relative navigation update rate is 1Hz.
- ★ In circuit programming
- ★ Crash halt recovery by using watchdog timer

5.6 Relative Navigation Software:

For a disciplined software management, development and deterministic context switching, it was decided to use real time operating system (RTOS). RTOS provides some valuable services to the applications, such as Time delays, System time, inter process communication management, task synchronization and system resources protection. uC/OS-II RTOS was selected from the different RTOS options available. It is a scalable, preemptive, multitasking real time kernel that is a safe and robust real time operating system and has been certified for the avionics product by the Federal Aviation Administration (FAA) (Jean J Labrosse [1]). It has very small footprints and it can run up to 64 tasks in parallel. It allows nesting of the interrupts and 255 interrupts can be nested. Lots of literature supports are available for porting to ARM processor. It is a very fast RTOS and the execution time for most of the services provided by uC/OS-II is both constant and deterministic. Its source code is available and there was no need of a license for its use in university research. Navigation processor S3C2410 fulfills all the requirements for uC/OS-II. Most of the code of the uC/OS-II is processor independent and few functions are needed to be written again in target processor's assembly language according to its architecture. The following functions were written to port uC/OS-II to S3C2410 processor, OSStartHighRdy(), OSCtxSw(), OSIntCtxSw(), OSTickISR() and also uC/OS-II data types were define to match with ANSI C compiler (Jean J Labrosse [1]) , (Deitel [53]) [54].

OSStartHighRdy( ) is used only once. It loads the stack pointer of the CPU with the top of the stack of highest priority task. Task level switching is accomplished by issuing software interrupt or Trap instruction. The switching between the tasks is called context switching and is the responsibility of the OSCtxSw( ) function. uC-OS-II is a preemptive operating system so as soon as any high priority task gets ready, the execution of the current task is suspended temporarily and OSCtxSw( ) saves all the processor registers of the running

task in current task stack, retrieves new task pointer and restore the value of CPU registers for new task. OSIntCtxSw( ) performs a context switch from an interrupt service routine (ISR). It finds the interrupt source and load appropriate interrupt handler routine. It also takes care of the nesting of the interrupts. Before calling OSIntCtxSw( ) from ISR, all the CPU registers of the interrupted task are saved in its stack. It only restores CPU registers of the new stack. At present three interrupts and 4 tasks are being used in this navigation code. OSCtxSw( ) OSIntCtxSw( ) subroutines are performing very well. OSTickISR( ) is a timer interrupt handler and provides a periodic time source to uC/OS-II to keep track of time delays, timeouts and other time related activities. Just like ISR, it also needs to store CPU registers of running task and perform necessary action and then resume the registers of High priority ready to run task. All these routines discussed above involves saving and restoring of CPU registers and hence are processor related, so their ports are written according to target processor's architecture by using its assembly language. The port of these routine for S3C2410 in ARM9 assembly language is given in Appendix C.

One of the important performance indexes for a RTOS is the context switch latency and recovery time. For the S3C2410 in supervisor mode, uC/OS-II uses 23 instructions and 22 memory access. Running at 203MHz and 203/2 MHz memory bus speed, it takes less than 4 μsec for context switching, which is excellent response time. This time remain constant and therefore response to any real time event will occur in deterministic manner. uC/OS-II is a multitasking operating system and 64 tasks can be run in parallel. RTOS uC/OS-II allocates priority to each task and as soon as the high priority task is ready to run the context switch routine OSCtxSw(), saves the context of the present task on its stack and gives the CPU control to the high priority ready task.

5.6.1 Initialization and loading of the RTOS:

After power on reset, the execution starts from the ResetHandler( ) subroutine, which disables all interrupts and sub interrupts, disables the watchdog timer and all code and data present in the flash ROM are copied to SDRAM so that code can run from SDRAM in order to speed up instruction loading. At this stage the navigation system displays a message on an external computer about "RTOS Boot successfully". Control is then passed to main( ) routine. First of all the system and bus clock values and ratios are programmed. Three UART port settings and baud rates are programmed. I/O ports are programmed according to their

assignment (either input or output). Interrupt vectors are initialized, DMA ports are initialized, timers are programmed and RTOS timing tick is programmed. OSInit( ) initializes uC/OS-II's variable and data structures. It also creates two task OS_TaskIdle( ) and OS_TaskStat( ). These tasks use CPU, when no other tasks are running on the CPU Jean J Labrosse [1].

Next the navigation system goes into interactive mode. Here the user can use different programming and simulation options through an external computer, like configure GPS, configure radio modem, run simulation, display mode, radio modem range test, radio modem EEPROM read etc. The details of these options are given in the next chapters. System time is set to zero, tasks are generated. OSStart( ) is called to start multitasking and give control to uC/OS-II. The job of OSStart( ) is to determine which of all the tasks created has highest priority and start executing this task. Now embedded program never return to main( ) and keeps on running tasks and interrupt. The flow chart presented in figure 5.7 showing flow of initialization, interactive flow, loading of RTOS and running the program selected through interactive selection. RTOS is loaded only for simulation and data display mode. Once RTOS is loaded the program never come back to the interactive options, unless the navigation system is reset.

5.6.2 GPS Programming Routine:

In this routine the GPS receiver is informed about the messages required and their protocol (UBX or Binary). GPS receiver's, baud rate, dynamic condition, time source, time period, positive or negative pulse of 1PPS signal are programmed. It also programs the GPS receiver for number of data messages required and their update rate. This routine receives the programming parameters from the external computer. To program the GPS receiver CFG class commands with different IDs are used. If the CFG command is successfully received by the receiver, it issues an acknowledgment ACK-ACK message and in case of failure to program receiver, it will issue an ACK-NAK message [55]. In both cases the navigation processor informs the external computer, which then takes the necessary action (may change that parameter values and try again). After programming the GPS receiver, control is given back to the OS, which waits for further command from the external computer.

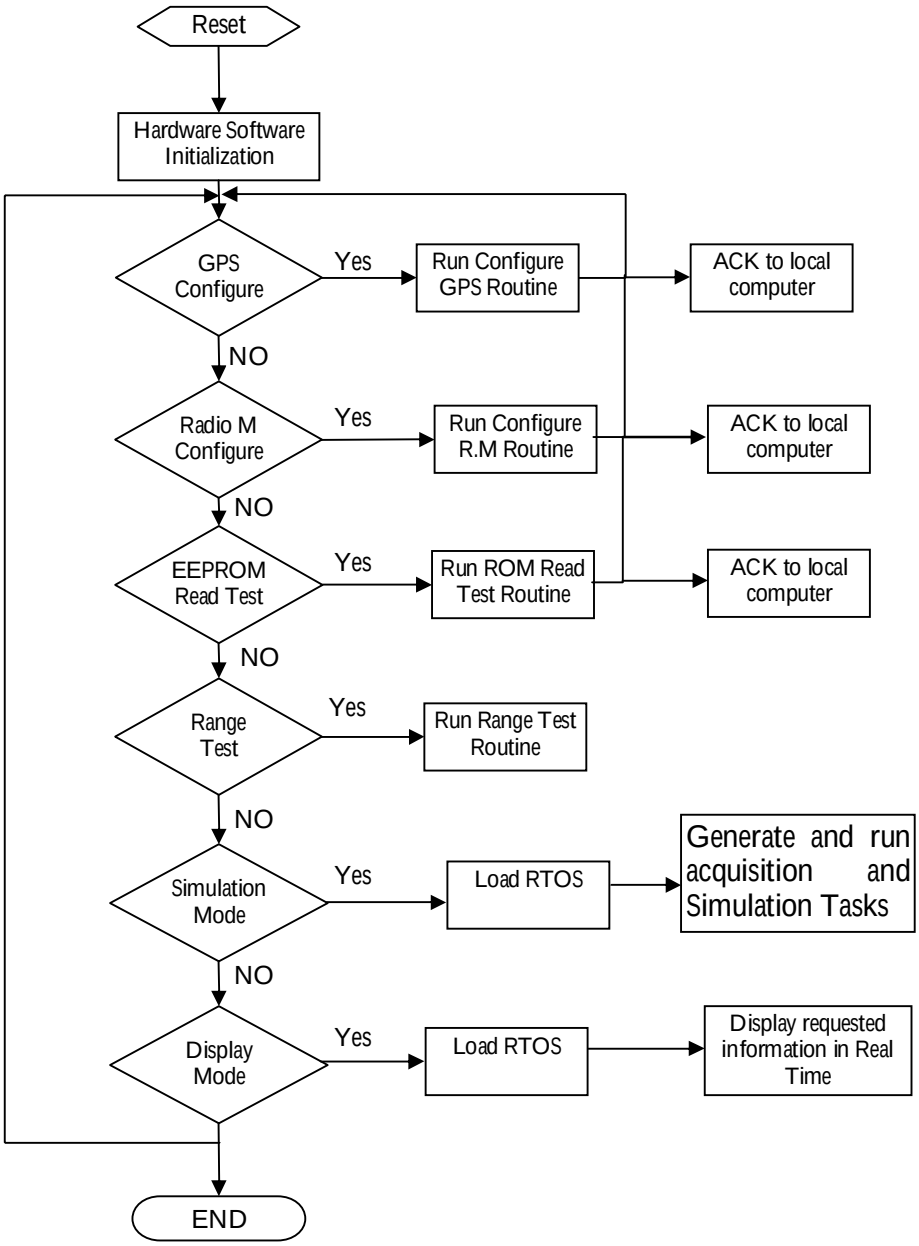


Figure: (5.7), Initialization and Interactive Flow in embedded navigation code.

5.6.3 Radio Modem Programming Routine:

This routine programs different EEPROM parameters of the radio modem according to the mission requirement. It programs transmit protocol (Unicast, broadcast), maximum power (range), Half Duplex/Full Duplex, network ID (network password), channel number, destination address (for address acknowledge mode). There are two ways to program the

transceiver, either by software command or by pulling Command/Data line (Hardware method). The software method is slow and not suitable for real time applications. To program a transceiver, therefore the hardware method is used here. In command mode by using Command/Data (Pin7), transceiver still can receive transmit data from other transceiver, while in command mode through software command, transceiver are able to receive or transmit any data. Command starts with 'CC' and on successful programming, transceiver acknowledge it by sending back the value of programmed parameter. All these parameters setting will revert back to the setting stored in EEPROM when the transceiver is reset. Control is given back to the OS, which waits for further command from the interface computer.

5.6.4 Data Acquisition and Sorting Task:

This task has the highest priority. Its main responsibility is to receive binary raw data, extract information, arrange them in structures/arrays and pass them to other tasks. Deputy satellites receive data from GPS receiver and also from master satellites, so in deputy satellite navigation system there are two data acquisition/sorting tasks. In the case of the decentralized approach the master satellite receives raw information and broadcasts it to other satellites in the formation and it does not receive any information from the other satellites, therefore in the decentralized approach the master navigation system will have only one data acquisition/sorting task. In this case of centralized approach both master and deputy satellites will have two data acquisition/sorting tasks. The task creation for centralized or decentralized approach is controlled by the GIS software.

All the data required for the relative and absolute navigation algorithm are present in RXM_RAW and RXW_EPH messages. The data acquisition from the serial port is interrupt based and DMA controllers are used to transfer data from UART FIFO buffer, which not only speeds up the transfer of data but also reduces the CPU overhead. As soon as 16 byte of data are available in serial port receive FIFO buffer an interrupt is generated. The data length of RXW_RAW is variable depending upon the visible GPS satellites, UART ISR calculates the data length followed and asks the DMA0 controller to transfer data from UART0 to some specified memory location without involving CPU. At completion of data transfer DMA0 post a message in mail box GPSMboxD. As soon as the message is posted in mail box, high priority data acquisition task DATA_ACQ( ) takes the control of CPU and pick the data from the memory and sort it accordingly. The RXM-RAW message contains raw data for all visible

satellites, their satellite number, GPS time and week number. The floating point data Carrier phase, pseudo phase and Doppler measurement are transmitted in double precision IEEE754 format. These data are extracted back to decimal format and then are written to predefined data structure. The pointer to this data structure is posted in a message queue GPSQ, where it will be available to other tasks. In case of master satellite, this structure is broadcast to other satellites. Similarly ephemeris data is received, extracted and written to data structure. This data structure is posted to message queue EPHQ (a pointer to message queue), where it will be available to other tasks. The ephemeris data contains satellite vehicle number, HOW, and three subframes. The details of satellite parameters in these subframes and their units are given in [56] Page 94.

5.6.5 Data Verification:

The navigation observation data fed to absolute and relative navigation tasks passes through different stages, like data acquisition and storing routine, conversion to integer and decimal data subroutines, their management and arrangement in structures and queues and then their transmission to other satellites in the formation. In order to verify that all these stages are working properly and data fed to navigation tasks are correct data, the following tests were conducted.

GPS receiver transmits data in binary format called UBX format. The format of this message is given in [55]. The binary data is transmitted in little endian mode and floating point data are transmitted by using IEEE754 format. The data contained in one of RXM-RAW message is presented Appendix D

The different data present in this message are sorted by the navigation processor and arranged in arrays in message queues. Pseudo and carrier measurements are present in 64 bit IEEE754 floating point format and Doppler measurement are present in 32 bit IEEE754 floating point. To verify correct data acquisition and sorting of the data, binary data packet and the final data to be used in navigation tasks are transmitted to external computer. The binary data is sorted manually in external computer. IEEE754 floating point data are converted to decimal presentation by using IEEE754 to decimal converter calculator presented by [57]. The converted integer and floating point values from the UBX binary packet are presented in Appendix D in the same order as present in the binary packet.

The last two bytes in binary data represent the checksum of data transmitted and is calculated by using fletcher algorithm[55]. Once a complete data set is received in navigation processor, its check sum is calculated by using fletcher algorithm and then compared with the checksum transmitted by the receiver. If both checksum are in accordance, then data received are valid date and further sorting algorithms are performed on it. After sorting, the data is arranged in arrays in a data structure and master satellite transmits these data structures to deputy satellite. Deputy satellite sends this data to navigation task and also sends it to external computers. The data are arranged and send to external computer in following order

TOW+WEEK NO+

Satellite NO1+Pseudo Measurement_1+Carrier Masurement_1+Doppler Measurement_1+

Satellite NO_2+Pseudo Measurement_2+Carrier Masurement_2+Doppler Measurement_2+

-
-
-
-

Satellite NO_n+Pseudo Measurement_n+Carrier Masurement_n+Doppler Measurement_n+

The sorted data received in external computer is presented in Appendix D. The data transmitted by the GPS receiver and data in integer and floating point format that are being used by the navigation algorithm are compared were found to perfectly match to each other. That's verifies that data transmitted by the GPS receiver are acquired and sorted in a the right way.

5.6.6 Debugging and Display Task:

This task is written for debugging purposes. It just captures the data structure pointer from GPSQ and EPHQ message queues. If it transmits captured data to the external computer, then it means the GPS and data acquisition task are working properly. Sometimes the GPS does not lock on to the GPS satellites due to weather conditions , hardware problems , radio interference or software errors. In that case GPS can be connected directly to the

external computer by using special “GPS RS232” cable. Hardware is designed in such a way that by connecting the special GPS serial cable, will disconnect the navigation processor and the GPS receiver. Now UCentre software from the U-Blox connects directly to the GPS receiver and can be used to check GPS receiver’s health, available GPS satellites and their signal strength and GPS raw data. This approach can also be used to verify the absolute navigation algorithm. One such run by using special “GPS RS232” serial cable is shown below in figure (5.8).

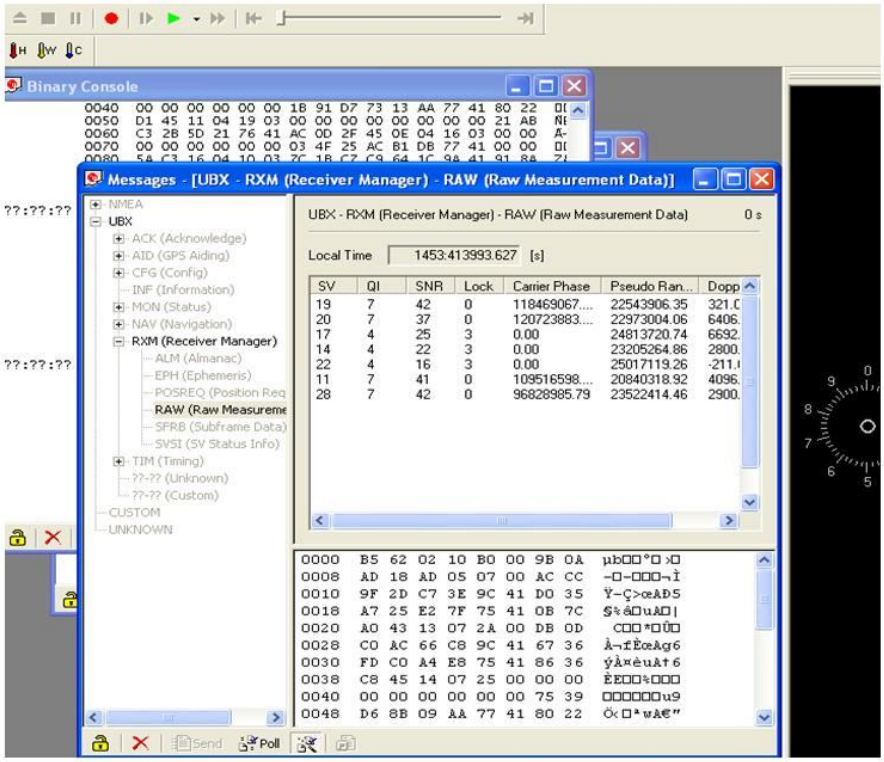


Figure 5.8: Debugging and testing of GPS receiver by using UCentre Software from UBlox

This figure shows GPS captured raw data. The lower right window shows data in UBX binary format, while the upper right window shows raw data in decimal and integer format. This window displays satellite vehicle number and their raw data (carrier phase, pseudo range, and Doppler shift). This information can also be used to verify captured data in navigation system.

Radio Modem Debugging: Some times it may happen that the radio modem does not respond. This can be due to navigation software error, wrong radio modem programming

(wrong address and ID), hardware problems or the radio modem itself can cause a problem. In such a case it is important to eliminate the error on the radio modem side. For this purpose the radio modem can be mounted on its adaptor board and connect with the laptop through the USB port. Load 'aerocomm.exe' software, its main interface window is shown in figure (5.9) [52]. The 'aerocomm.exe' is used to read, write, configure and test radio transceiver.

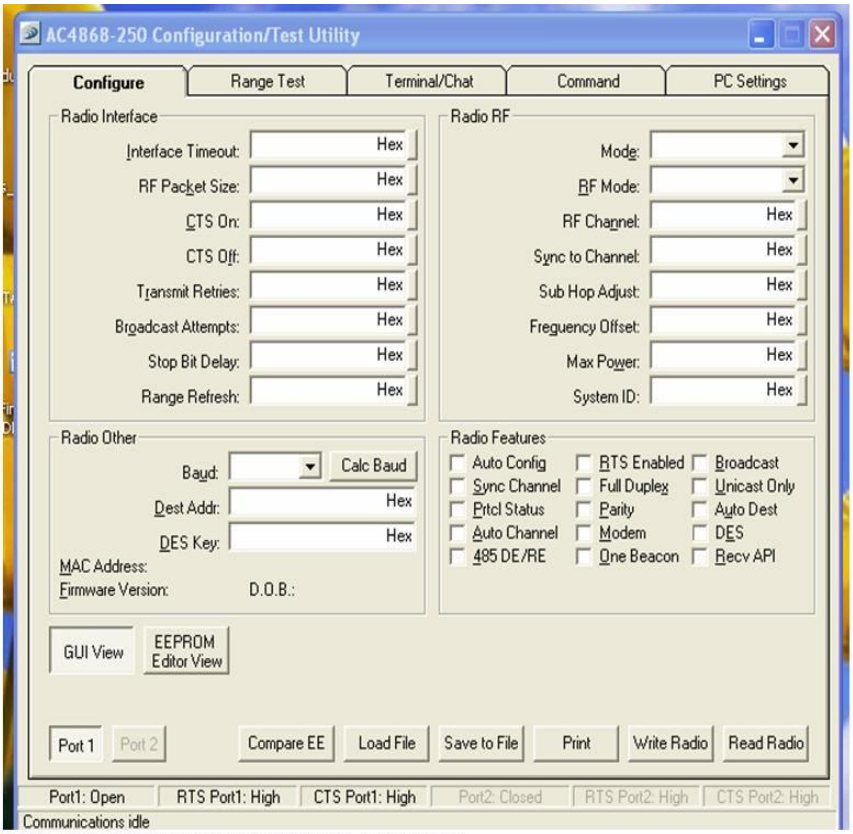


Figure 5.9: Radio Transceiver test software interface

Through this software, the health of the radio modem can be check, the EEPROM can be read to see any programming error and different tests can be performed to validate the proper functioning of the radio transceiver.

Once an error on the radio modem and GPS side is eliminated then debugging in navigation software can be started. This can be done by removing the transceiver module and using a simple RS232 cable to connect processor module with the laptop. This simple RS232 cable disconnects processor module and GPS module. Now download "sertest" program into processor module. This program sends messages to the external computer on serial port. The

reception of these messages in external computer verifies processor module is working properly. Once these entire tests have passed it will confirm that all hardware is working fine and the problem is in software code. The software can be debugged by commenting its relevant routine or task.

5.6.7 Validation of Relative Navigation Hardware and Software:

The relative navigation system consists of hardware and software. Different validation tests were run to verify the correctness of the hardware and software. Here a step by step analysis is presented. The hardware of the relative navigation system consists of a microprocessor, a GPS sensor, radio transceiver, one RS232 communication port and 8 digital I/O. Whenever power is ON, a red light on the box and a power LED on the processor module glow to indicate that power is there and the hardware is ON. During the boot process, the microprocessor configures different onboard peripherals and ports. After successful booting, it transmits a success message to the GIS software, which indicates that all the peripherals have been programmed, serial communication is working and the RTOS has been loaded successfully. At this stage the navigation program goes into an interactive mode that provides the options to configure the GPS and the radio transceiver. For the GPS configuration different programming parameters are set and downloaded through the GIS software via a serial link. The microprocessor receives these parameters and sends them to the GPS receiver according to UBX protocol. On successful programming the GPS receiver issues an acknowledge message and in case of failure the GPS does not issue an acknowledge message. These messages are transferred to the external computer to verify the process of GPS programming. These steps are shown in the video CD attached with the thesis in the folder named GPS_Programming. The interactive program file and Matlab function are presented in Appendix G and the whole Embedded code and Matlab GIS software is also attached in the CD.

Similarly configuration parameters for radio transceivers can be adjusted and downloaded from the GIS software to the radio transceiver via the microprocessor. The subroutine radio_modem( ) is responsible for programming the radio transceiver and 'Radio_Modem.m' is the Matlab function responsible for communicating with the radio

modem through the embedded microprocessor. After successful downloading the radio transceiver acknowledges back to the GIS software via the microprocessor. These functions, subroutines are added in the attached CD.

In-Range is a facility developed in this navigation system, which tells the deputy satellite, when it is in range from master satellite navigation system. To test this aspect of the navigation system, the master satellite is switched OFF and ON. As soon as the master navigation system is switched OFF, the deputy satellite sets the 'IN_RANGE_FLAGE' flag and also informs the GIS software that he is out of range and could not see the master satellite. By switching the master satellite navigation system ON, deputy satellite reset 'IN_RANGE_FLAGE' flag and also inform to the GIS software. This subroutine and corresponding Matlab routine is presented in Appendix G and video clips of this test is included in the attached CD in a folder named Radio_Modem_IN_Range.

Simulation option set by the SIM_MODE flag, is later used by the RTOS to create different sets of tasks. GPS data acquisition is interrupt based, as soon as data is available on the receive port an interrupt is generated to the RTOS. Interrupt handler ask the DMA to transfer data from serial port to a prespecified memory location, and gives the control back to the microprocessor. At the end of data DMA interrupt the RTOS to inform that new raw data is available. Control is then given to the 'DATA_ACQD' task, which has the highest priority. This task reads the memory buffer, sort the information, and check the health of the received data by verifying checksum. All the navigation data is received in binary format. These data are then converted back to integer and floating point data. Finally the data is arranged in message boxes and send to navigation code task. For the verification for all the hardware and software involved simulation mode, raw data in binary format and final arranged data is transmitted to GIS software. In binary data the length of data its checksum, no of visible satellite, raw information for each visible satellite, its time stamp, data update rate are calculated manually and then compared with data sorted and arranged by 'DATA_ACQD' task. Both data are always found perfectly matching without any data missing. A detail description of data verification is already given in section 5.6.5 and data comparison is presented in Appendix D. A video clip for this activity is included in the CD attached here in the folder named Full_test. In the simulation mode the deputy satellite also receive the master satellite's navigation data

through the radio transceiver. This data is also transmitted to GIS software. GIS software logs all this navigation data from master and deputy satellite.

Testing and verification of the operation of the GPS has been discussed in section 5.6.6. The video clips for these tests are given in the folder named GPS_Isolate in the CD.

5.6.8 Absolute Navigation Algorithm:

Up till this stage raw information from the GPS binary signals have been extracted. This information includes TOW, Week number, Satellite number, pseudo measurements, carrier measurements, Doppler measurements and ephemeris data. Raw information is updated at the rate of 8Hz, whereas ephemeris data is updated after 2 hours. Raw information is used in both absolute and relative navigation algorithm. At least 4 GPS satellites should be visible to run the navigation algorithm. As soon as 4 GPS satellites are locked, the event flag is set and the navigation task becomes active.

Before proceeding forward for absolute and relative navigation, the following is assumed

- ✓ Cycle slip and Integer ambiguity has been resolved
- ✓ It is supposed that the two satellites are close and the GPS signals used by them pass through the same medium. Therefore in single difference equation ionospheric error will be cancelled. For absolute navigation, ionospheric error can be estimated by using correction data in ephemeris data.
- ✓ GPS satellite's clock bias has been calculated by the using the data transmitted in ephemeris data.

The absolute states calculation algorithm uses Least Square Estimate method and its complete algorithm is given in [56]

5.6.9 Relative Navigation Algorithm:

This task calculates relative navigation states by using raw data and absolute positions of master and GPS satellites. For a decentralized formation topology, this task will be present in the deputy satellite's navigation system only. The easiest way to calculate relative states is to

use equation 3.36, but the drawback in this algorithm is that it depends on absolute position accuracy. The absolute position accuracy is not very good and that error will propagate in relative state estimation as well. The equation 3.36 however can be used for initial estimation for the following algorithm. Another approach was to use difference equations, which does not depend on absolute state's accuracy. This relative navigation algorithm is developed by using difference equation.

This task picks data structures from message queues, GPSQ and EPHQ of local receiver and GPSQ, EPHQ of remote receiver stored in local navigation system. Data with the same time tag and for the same GPS satellites are selected and single difference observations data are generated. From this relative navigation single difference and double difference equations can be used. Single difference gives uncorrelated equations and double difference gives correlated equations. It is easy to solve uncorrelated equations, while the algorithm for the solution of double difference becomes very complicated. Single difference and double difference techniques demonstrate similar performance for relative navigation solution [51], therefore in this navigation algorithm single difference algorithm is adopted. For single difference, satellite clock bias cancelled. It is supposed here that it is closely placed formation and therefore in single difference observation, any ionospheric bias will also be cancelled. The unknowns in the single difference equations are $(\Delta x, \Delta y, \Delta z, \Delta \delta t)$ where Δx , Δy , Δz are components of baseline vector and $\Delta \delta t$ is the differential receiver clock error. The number of single difference equations depends upon the visible GPS satellites common to both receiver. As there are four unknown parameters at a minimum four single difference equations are required. Where there are more than four single difference equations available the number of equations will become more than unknowns. In this case Least Square Estimate (LSE) or Kalman Filter techniques are used to solve these set of equations. LSE is easy to implement, but gives less accuracy as compared to Kalman filter techniques. For the implementation of Kalman filtering, the process model and dynamic model of the formation flying should be known. The single difference equations gives measurement model equations and many satellite formation flying dynamic models are available for different frame of reference. As all the observation and ephemeris data transmitted from GPS satellites are in ECEF frame, most of the algorithms already developed measures relative state vectors in ECEF frame and then transform them into the master's local frame of reference. The satellite formation flying model

for ECEF frame used in that algorithm is quite complicated and involves modeling error. The Jacobean matrix used in the state and covariance propagation scheme in Kalman Filtering algorithm is quite complicated and large (8×8). It affects the quick convergence of Extended Kalman Filter. The 2nd approach also uses Extended Kalman Filter to estimate relative states, but now measurement and process models are defined in a formation flying frame of reference (RSW). Hill's equations are used as the relative dynamic model, which is easy to implement. The Jacobean matrix is smaller (6×6) and simpler to calculate. This approach is easy to implement and it will give the relative states directly in master frame of reference. The relative states to be estimated are

$$x_k = \begin{bmatrix} \Delta r \\ \Delta \delta t \\ \Delta \dot{r} \\ \Delta \dot{\delta} t \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(5. 1)}$$

Where Δr is a relative separation, $\Delta \dot{r}$ is relative velocity, $\Delta \delta t$ is receiver's clock error and $\Delta \dot{\delta} t$ is receiver's clock drift rate. The measurement model and the relative dynamic models in the formation flying frame of reference are presented in the following section.

Measurement Model:

Measurement model include observation equations. The observation equations in ECEF frame are[11, 58]

$$\left.\begin{aligned} P(t) &= \rho(t) + c(\delta t - \delta t_{GPS}) + I + \varepsilon_p \\ \Phi(t) &= \rho(t) + c(\delta t - \delta t_{GPS}) + N\lambda - I + \varepsilon_\Phi \\ D(t) &= \dot{\rho}(t) + c(\delta f - \delta f_{GPS}) - \dot{I} + \varepsilon_D \end{aligned}\right\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(5. 2)}$$

$P(t)$, $\Phi(t)$, $D(t)$ are pseudo, carrier and Doppler observations. $\rho(t)$ is the actual range between the receiver and GPS satellite, $\dot{\rho}(t)$ is the range rate, c is speed of light, *I* is ionospheric path delay, $\dot{I}$ is ionospheric path delay rate, $N\lambda$ is unknown bias equal to an integer number of wavelengths, ε denote the other errors in the corresponding observation equations. Single Difference equations are [11, 58]

$$\left.\begin{aligned} \Delta P(t) &= \rho_m^j(t) - \rho_d^j(t) + c\Delta\delta t + \sqrt{2}\varepsilon_P \\ \Delta\Phi &= \rho_m^j(t) - \rho_d^j(t) + c\Delta\delta t + \sqrt{2}\varepsilon_\Phi + \Delta N\lambda \\ D(t) &= \dot{\Delta\rho}(t) + c\Delta\delta f + \sqrt{2}\varepsilon_D \end{aligned}\right\} \quad \text{..... (5. 3)}$$

Where j represent j^{th} GPS satellite. Subscript *m* and *d* represent master and deputy satellites, Δ showing the difference between two receiver's values for same GPS satellite at same time tag. The GPS satellites are far away from the LEO satellites, hence $\vec{\rho}_m^j(t)$ and $\vec{\rho}_d^j(t)$ vectors are supposed to be parallel. Now from the figure 5.10

$$\Delta\rho = \rho_m^j(t) - \rho_d^j(t) = |SD| = \vec{b} \cdot \frac{L\vec{O}S}{|L\vec{O}S|} \quad \text{..... (5. 4)}$$

Where $\vec{b}$ is a base vector and $L\vec{O}S$ is a Line of Sight vector along the line joining master satellite and GPS satellite and is

$$L\vec{O}S = \vec{r}_j - \vec{r}_m$$

As the formation flying frame of reference (RSW) is selected for this navigation algorithm, to define equation 5.3 in the formation flying frame of reference (RSW), the two vectors $\vec{b}$ and $L\vec{O}S$ should be defined in the RSW frame. Let the component of unknown relative vector $\vec{b}$ in formation flying frame RSW is $\dot{\vec{b}} = (\dot{x}_{rel}, \dot{y}_{rel}, \dot{z}_{rel})$ and $L\vec{O}S$ in RSW frame can be defined by using equation 3.36 as.

X-component of $\vec{LOS}$ vector will be

$$x_{LOS} = (\frac{\vec{r}_m}{|r_m|} \cdot \vec{r}_j) - r_m$$

Where $\vec{r}_m$ and $\vec{r}_j$ are defined in geocentric frame of reference (IJK).

Y-Component of $\vec{LOS}$ vector will be

$$y_{LOS} = (\frac{\vec{h}_m}{h_m} \times \frac{\vec{r}_m}{r_m}) \cdot \vec{r}_j$$

Z-Component of $\vec{LOS}$ vector will be

$$z_{LOS} = (\frac{\vec{h}_m}{h_m}) \cdot \vec{r}_j$$

Now $\Delta\rho$ can be written from equation 5.4

$$\Delta\rho = \rho_m^j(t) - \rho_d^j(t) = |SD| = (x_{rel}, y_{rel}, z_{rel}) \cdot \frac{(x_{LOS}, y_{LOS}, z_{LOS})}{|\vec{LOS}|}$$

Differentiate above equation

$$\Delta\dot{\rho} = \frac{d}{dt}(\rho_m^j(t) - \rho_d^j(t)) = \vec{b} \cdot \frac{\dot{\vec{LOS}}}{|\vec{LOS}|} + \dot{\vec{b}} \cdot \frac{\vec{LOS}}{|\vec{LOS}|} \qquad \text{…….... (5. 5)}$$

Where $\dot{\vec{LOS}}$ is velocity vector in formation flying frame (RSW), in the direction of $\vec{LOS}$ vector and can be calculated by using equation 3.61 to 3.63.

Now equation 5.3 is

$$\Delta P(t) = (x_{rel}, y_{rel}, z_{rel}) \cdot \frac{(x_{LOS}, y_{LOS}, z_{LOS})}{\left|\vec{LOS}\right|} + c\delta t + \sqrt{2}\varepsilon_P \quad \text{..... (5. 6)}$$

$$\Delta\Phi(t) = (x_{rel}, y_{rel}, z_{rel}) \cdot \frac{(x_{LOS}, y_{LOS}, z_{LOS})}{\left|\vec{LOS}\right|} + c\delta t + \sqrt{2}\varepsilon_\Phi \quad \text{..... (5. 7)}$$

In above equation, it is supposed that integer ambiguity has been resolved already

$$\Delta D(t) = \vec{b} \cdot \frac{\dot{r}_{LOS}}{|r_{LOS}|} + \dot{\vec{b}} \cdot \frac{r_{LOS}}{|r_{LOS}|} + c\Delta\delta f + \sqrt{2}\varepsilon_D \quad \text{..... (5. 8)}$$

This set of single difference equations gives the measurement equations in the formation flying frame of reference. Carrier phase and Doppler measurement equations are used because they give better accuracy, The GPS receiver used can track up to 16 GPS satellites in parallel, so the differential measurement equation vector may include as many as 16 sets of equations, so the maximum number of observation equations can be 32. The expected measurement function is

$$y_k' = \begin{bmatrix} (x_{rel}, y_{rel}, z_{rel}) \cdot \frac{(x_{LOS}, y_{LOS}, z_{LOS})}{\left|\vec{LOS}\right|} + c\delta t \\ \bullet \\ \bullet \\ \bullet \\ (x_{rel}, y_{rel}, z_{rel}) \cdot \frac{(x_{LOS}, y_{LOS}, z_{LOS})^N}{\left|\vec{LOS}\right|} + c\delta t \\ (x_{rel}, y_{rel}, z_{rel}) \cdot \frac{\dot{\vec{LOS}}}{\left|\vec{LOS}\right|} + (\dot{x}_{rel}, \dot{y}_{rel}, \dot{z}_{rel}) \cdot \frac{\vec{LOS}}{\left|\vec{LOS}\right|} + c\Delta\delta f \\ \bullet \\ \bullet \\ \bullet \\ (x_{rel}, y_{rel}, z_{rel}) \cdot \frac{\dot{\vec{LOS}}^N}{\left|\vec{LOS}\right|} + (\dot{x}_{rel}, \dot{y}_{rel}, \dot{z}_{rel}) \cdot \frac{\vec{LOS}^N}{\left|\vec{LOS}\right|} + c\Delta\delta f \end{bmatrix} + \begin{bmatrix} \sqrt{2}\varepsilon_\phi \\ \cdot \\ \cdot \\ \cdot \\ \sqrt{2}\varepsilon_\phi \\ \sqrt{2}\varepsilon_D \\ \cdot \\ \cdot \\ \sqrt{2}\varepsilon_D \end{bmatrix} \quad \text{.....(5. 9)}$$

Where $y_k^{'}$ is the observation column vector. N varies from 4 to 16, as the minimum number of visible GPS satellites has to be 4. The above equation can be written in matrix form as.

$$y_k = h_k(x_k) + v$$

Where $h_k(x_k)$ is observation matrix in RSW frame of reference in terms of unknown parameters $(\Delta r, \delta t, \Delta \dot{r}, \delta f)$ and v is error vector.

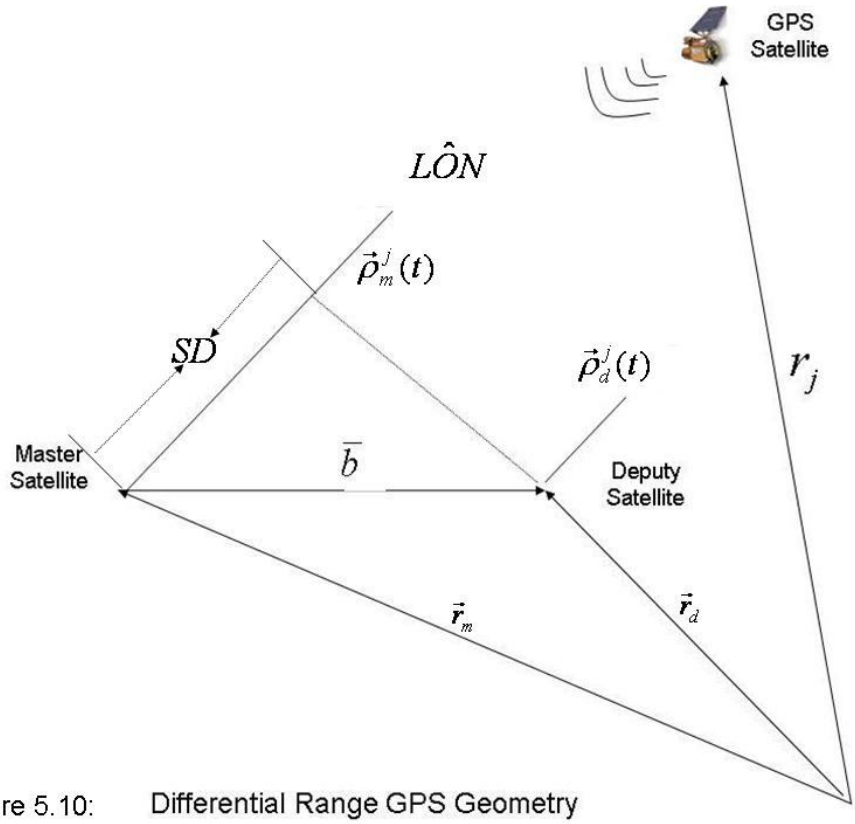


Figure 5.10: Differential Range GPS Geometry

Process Model:

Another requirement for Kalman filter algorithm is a process relative dynamic model in formation flying frame. Hill's equation gives dynamic of relative states of satellites in master frame of reference and can be used as process model in this algorithm. Although it is not accurate and have some assumptions and modeling errors associated with it. These assumptions and modeling errors is lumped together in error term *u*..

$$\ddot{x} = 2\omega\dot{y} + 3\omega^2 x + u_x$$
$$\ddot{y} = -2\omega\dot{x} + u_y \qquad \text{..... (5. 10)}$$
$$\ddot{z} = -\omega^2 z + u_z$$

Where u_x, u_y and u_z are modeling errors. The process model in matrix form is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 \\ 3\omega^2 & 0 & 0 & 0 & 2\omega & 0 \\ 0 & 0 & 0 & -2\omega & 0 & 0 \\ 0 & 0 & -\omega^2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + u \qquad \text{..... (5. 11)}$$

Where *u* is error column vector. Now both measurement and dynamic models in the formation flying frame of reference are known and are ready to be used in the Extended Kalman filter. A complete algorithm of the Extended Kalman Filter is given by Greg Welch and Gary Bishop [59, 60]. The advantage of this method is that it is quite simple and easy, it has a small no of calculations in the Jacobean matrix, it gives relative states directly in formation flying frame of reference and it will converge faster. Due to lack of time and unavailability of GPS simulator, this algorithm has not been implemented and tested in embedded code.

5.7 Summary:

In this chapter the design and development of a fast compact, low power relative navigation system for nano satellites in low earth orbit is presented. The goal has been to develop a relative navigation as it would actually be used on a satellite. By designing a compact, low mass (<400 gm) system with low power consumption (1 Watt), it has been demonstrated that a GPS based relative navigation system is a practical choice as the navigation sensor for a formation of nano satellites. Single frequency GPS receiver can be used in absolute and relative navigation systems. Hardware is designed intelligently and it provides different debugging and testing options during software development. This relative navigation system can operate whilst moving with an acceleration of up to 4g. A radio transceiver is used

to exchange raw information between the satellites. The range between the navigation systems can be up to 15 km. Hardware has been designed to support both the centralized and the decentralized approach. It consumes very little power (1Watt) and it can run from a 5V power supply. All timing activities are synchronized with the GPS clock. Due to in-circuit programming facility, the navigation code can easily be changed and downloaded to the navigation process. All the hardware has been tested and it is working fine. All the tests were performed indoors by capturing GPS data directly from the GPS satellites by putting GPS antenna outside through window. GPS receiver provides raw measurement and ephemeris data. Raw data acquisition rate is set to 8Hz, and the navigation update rate is set to 1Hz. Extra raw data can be used in error elimination algorithms. The navigation processor (S3C2410) is a 32 bit processor running at 203MHz speed. A preemptive, multitasking, embedded real time operating system uC/OS-II is used to achieve deterministic real time performance. The software code is written in embedded C. The navigation code is spread into different tasks and these tasks have different priority levels. All data capturing are interrupt base and DMA are used to transfer data. It reduces the overhead on the main processor and also speeds up the data transfer. The basic infrastructure for the relative navigation software has been completed and has been tested successfully. Due to the lack of time absolute and relative navigation tasks could not be implemented in embedded environment, although a novel approach has been presented in this chapter. A distinctive algorithm is developed for relative navigation. It uses Extended Kalman Filtering technique and gives relative states directly in formation flying frame (RSW). It eliminates the requirement of post coordinate transformation of the output of Extended Kalman Filter results. The hardware and software code structure is designed in such a way so that it can be used for both centralized and decentralized formation flying approach.

Chapter 6

Graphical Interface Software

6.1 Introduction:

Another important part of the RTHIL setup was to develop graphical interface software. The purpose of the software was to control and synchronize the simulation, to set different simulation parameters before any simulation starts. This software manages to program the GPS and radio modem according to the simulation requirements. It provides different debugging options. It also provides online data captures during simulation running. Through this graphical interface software, different simulation options can be selected. This Graphical Interface Software (GIS) is developed in Matlab. The detail of its development and its different options are discussed in this chapter.

6.2 Graphical Interfacing Software (GIS):

The GIS was developed in Matlab by using a graphical editor. It is used to program the different components of the relative navigation system through navigation processor (S3C2410). GIS running on the external computers connects to the UART2 port of the S3C2410. The main window is called Nav_interfacex, where x represent the software version (Figure 6.1). The main window has two child windows, one is called gps_program and the other one is called Radio_Modem_Programming. These windows can be invoked through main menu or from the icons given on the main window. When this interface is loaded from Matlab, it initializes COM1 of host computer at 115200 baud rate and display message of successful initialization. In case COM1 is occupied by some other software, it will display a warning message and user can manually select other port from the main menu. It is also possible to change port selection any time from COM1 to COM2 and vise versa by selecting Port option from main menu and then click initialize port option in serial port drop down menu. At the time of initialization, it also opens a data capture file “deputy.txt” and

"master.txt". It is possible to change the name of data capturing file by using Capture

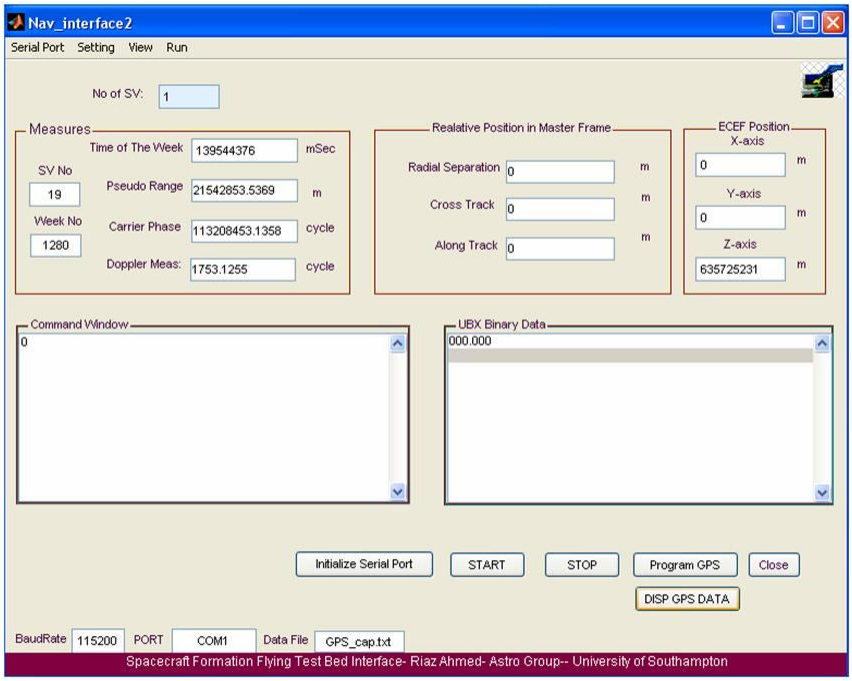


Figure 6.1: Main window of the GIS software

data option in setting drop down menu. All the data received are logged into these files. The port status can be check any time by clicking port status option in serial port menu. The selected port, its status and data capture file name is displayed at the bottom of main window. Communication status can be check any time by clicking communication test in Run menu. The Nav_interface2 window display raw measurement, absolute velocity, absolute position, relative position, satellite number for which data is being displayed and total number of GPS satellites visible to navigation system. It has data command window to send commands to the relative navigation system. This option will be used in future to send a set of commands to program and to set different parameters and simulation options to the navigation system. The other window "UBX Binary Data" is used to display binary data from the GPS and is used for debugging purposes. Data capturing can be stop and start at anytime. The main menu and their drop down options are shown in figure 6.3 and 6.4.

In order to run the simulation, either press START button or click RT Simulation in RUN menu. Under this option both relative navigation systems go into simulation mode. They acquire raw information from their GPS receivers, sort and arrange them in structures and pointer to these structures is then stored in message queues to be available to absolute and

relative navigation task. Master satellite also transmits raw information and time tag information to the deputy satellite. Deputy satellite arranges Master's raw data in structures and queues. Deputy satellite transmits its own and master's raw information to the external computer as well for validation and debugging purposes. Master's raw information is stored in mater.txt file and deputy satellite's raw information is stored in deputy.txt file. One such data capture in these files is presented in Appendix E. The setup for this real time simulation is presented below in figure 6.2. In this setup GPS receivers acquire true GPS signals from GPS satellites. GPS antennas were placed outside the building through window.

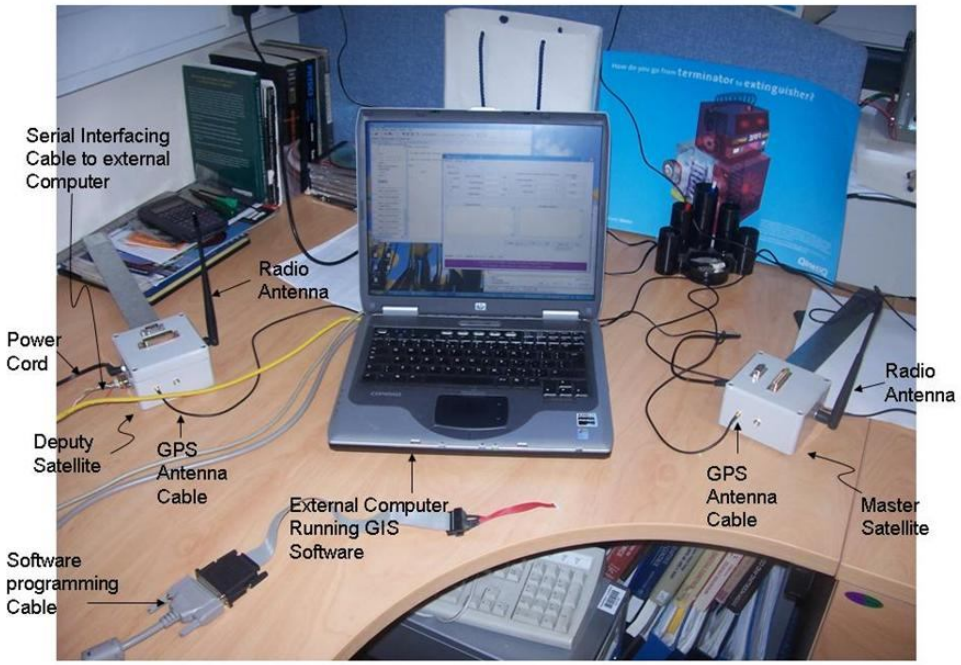


Figure 6.2 A small real time simulation setup with master and deputy relative navigation systems capturing data from the GPS satellites and exchanging them

In order to stop the communication or data capturing, just press the button "STOP". Click 'StopDataCap' in settings drop down menu to close both data capturing files. Once data capture files are closed, they can be open in any word editor for further processing

To see the GPS data, like raw measurements, absolute state in ECEF frame, Timing data, clock drift and clock bias, click 'GPS DATA' in view menu or click 'Disp GPS Data' button. GPS data will start displaying on the main window, it may take more than 30 Sec to lock to some satellites and in some case it may take more time to lock to GPS. As soon as GPS receivers lock to GPS satellites, a valid data will displayed in main window. Absolute state and clock data will only be updated after 4 or more than 4 GPS satellites are locked.

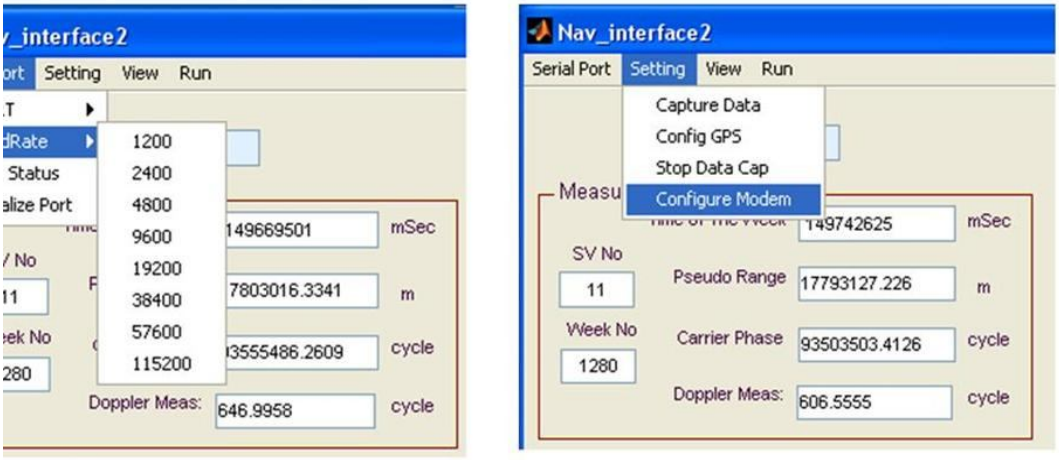


Figure 6.3: Different options under main menu Setting and sub menu Baud Rate

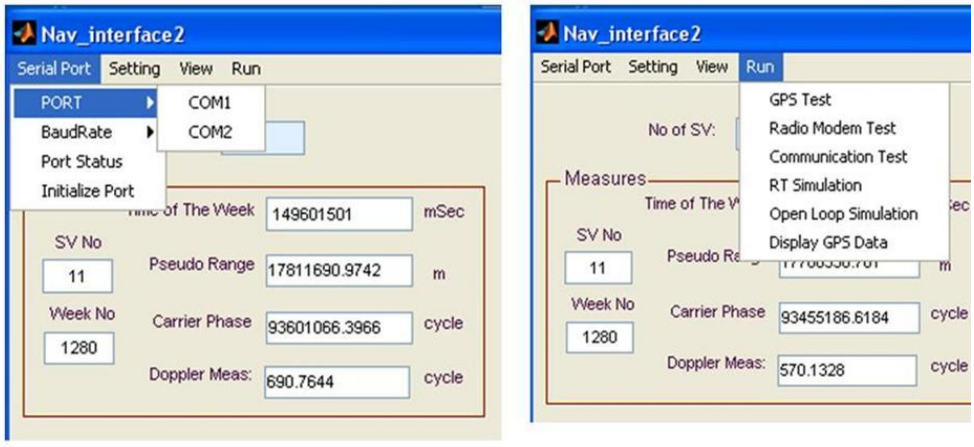


Figure 6.4: Different options under Serial port and Run options in Main menu of Graphic interface software's

Different test and simulation options can be run through Run menu., like GPS test, Radio modem test, communication test between navigation system and local computer, run real time simulation, display GPS data. Simulation can be Start and Stop any time during the simulation running. In order to configure GPS, press 'Program GPS' button or click 'config GPS' option

from settings menu. A new gps programming window will be opened (figure 6.5). First select, is it Master navigation system or Deputy. Set Different GPS receiver's parameter according to the test requirement and select apply to program GPS. GPS timing signal frequency, cycle duty, negative or positive pulse can be set here. Also timing activities can be synchronized either with GPS time or with UTC. Different Dynamic platform options are

Stationary
Pedestrian
Automotive
Airborne <1g
Airborne <2g
Airborne <4g

Navigation update rate and raw measurement rate can be set up to 4Hz and 10 Hz respectively. Navigation rate cannot be more than 4Hz or not more than raw measurement rate. Press OK will close the window. These parameters are transmitted to deputy satellite, which decide either these parameters are for deputy or master relative navigation system. Navigation processor programs these parameters step by step to the gps receiver. On the successful programming GPS receiver acknowledge by sending 0x05 0x01 bytes and 0x05 0x00[55] for unsuccessful programming. Navigation processor then informs the external computer bout successful or unsuccessful programming.

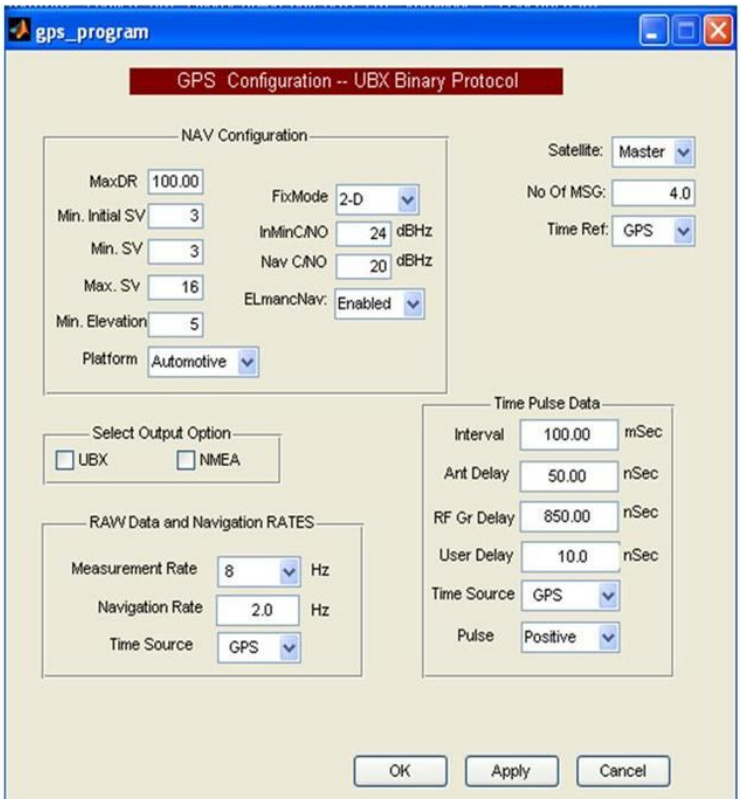


Figure 6.5: GPS Programming Window developed in Matlab

To open the Radio modem configuration window, click 'configure modem option' in the setting menu on main window. A new window will pop up (figure 6.6). First of all select either the master or deputy satellite. Some parameters are only valid for the deputy satellite and not available for the master satellite. Select RF communication protocol (broadcast, unicast). Write the values for maximum power, network ID (network password), channel number, destination address (for address acknowledge mode). Decide deputy's out of range response time in sec (Beacon Period- Range Refresh). Press 'Apply' button to write these parameters to the radio modem's EEPROM.

Press the 'In RANGE TEST' will identify whether the master satellite is in range of the deputy satellite or not. This test is valid only for the deputy satellite. If the master satellite's relative navigation system is on and is in range of deputy satellite, the deputy satellite will set the IN_RANGE flag to 0 and also transmit it to external computer for debugging purposes. If the master satellite is out of range or switched off, the deputy satellite will set IN_RANGE flag to 1 and also transmit this flag to the external computer. Now switch off the the master's radio modem. Deputy satellite waits for (Beacon Period-Range Refresh) sec, and then will set

IN_RANGE flag low or high. As soon as the master satellite comes in range of deputy satellite, it will set IN_RANGE flag back to 0.

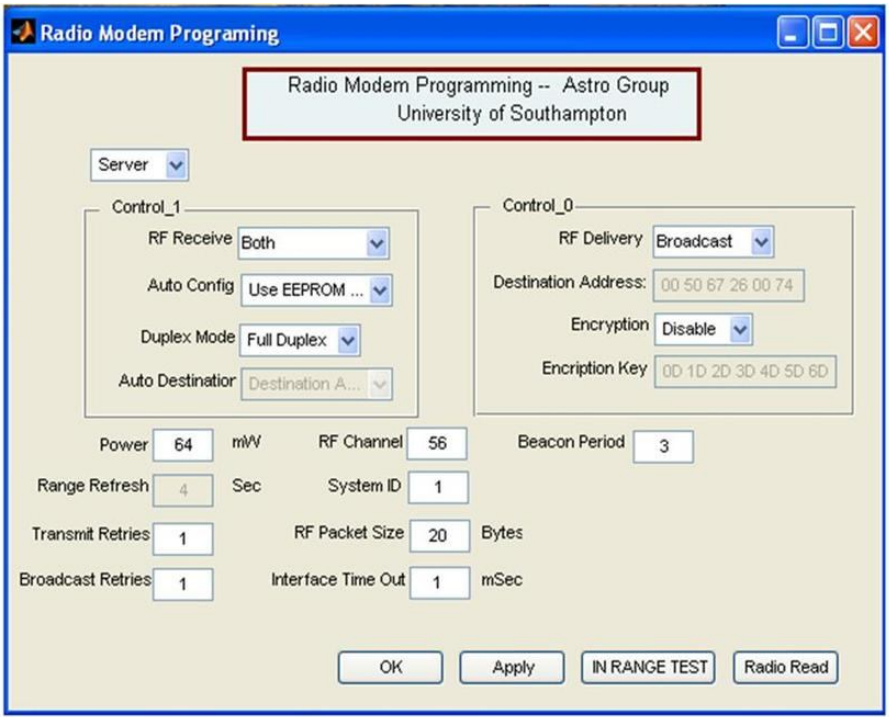


Figure 6.6: Radio Modem Programming Window developed in Matlab

The IN RANGE TEST has been run successfully and deputy satellite responded accordingly as master satellite is in range or out of range (by switching master satellite ON and OFF). 'Radio Read' button will read the written EEPROM's values to verify written parameters. Press 'OK' will close the window.

By pressing the close button in the main window all open windows will close and will prompt for verification of the window close action. If data capture files are open at this stage, they are also closed after prompting.

Section 4

RTHIL Setups, Recommendations and Conclusion

Chapter 7

RTHIL Setups, Recommendations and Conclusion

7.1 Full hardware in the close loop Real Time Simulation Setup:

The purpose of this project was to develop a real time hardware in the loop satellite formation flying test bed for nano satellites in LEO orbit. In this project

- A decentralized approach was selected for the software development.
- The simulation step time for all software on this test bed is selected as 1 sec.
- Matlab/SIMULINK and Real Time Windows Target is used to run real time simulation of satellite formation flying model.
- A single frequency GPS is used as relative navigation sensor.
- Radio signals are used to transfer information among the satellites.
- GPS time is used as reference time in navigation software code.

The real time hardware in the loop SFF test bed initially planned is given in figure 7.1. The test bed was divided into four sections. The section 1 consisted of a network of real time computers running a real time satellite formation flying model. This model is accurate, efficient and can be used for circular and elliptical orbits. It can also be used in centralized and decentralized mode and can include regular and stochastic perturbations, and be accommodated in closed or open loop. The effect of J2 perturbations and atmospheric drag are considered. It was then validated by running offline simulations for different formation flying scenarios and comparing the results with other already existing satellite formation flying models. It was then downloaded to a network of two single board real time computers to validate their performance in a real time simulation. This network of real time computers, record current states of each satellite, their relative states and their simulation time during each simulation step. This information is then used to verify relative navigation algorithm in close loop or in open loop. This section has been completed successfully.

Section 2 consisted of the development of relative navigation system's hardware and software. A compact, low power and fast relative navigation system was developed. Its main building blocks are a navigation processor (S3C2410), a single frequency GPS receiver (LEA-

4T), Radio transceiver (AC4868), 2MB Flash, 64 MB SDRAM. The selection of components, its hardware design, and development were the part of this section. An intelligent relative navigation system for this test bed has been developed and has been presented in chapter 5. Real time operating system uC/OS-II is used in the navigation code to guarantee the real time response, efficiency and reliability of the relative navigation code. The main software setup and tasks for this relative navigation code have been developed and verified. The navigation acquire GPS data, sorts and arranges the GPS data in structures and queues and the master satellite transmits raw information and time to the deputy satellite. Deputy navigation processor arranges raw data both from master satellite and from its own navigation system into message queues and pass on these data to navigation tasks. The navigation algorithm was described in chapter 5, but due to time constraints these navigation algorithms have not been implemented in navigation code. Navigation code provides many debugging options during software development. Due to its in-circuit programming facility, it is easy and fast to implement any changes in navigation code. The selection of components and hardware of the relative navigation system is designed in such a way that it supports both centralized and decentralized formation flying strategies. The design supports a formation of more than two satellites although all demonstrations and software developments have been done with two satellites due to finance constraints. Finally the graphical interface software (GIS) was developed to control and manage the flow of real time simulation. It records simulation data in real time from the deputy satellite and sets initial hardware and software parameters according to the requirements on both the deputy and the master navigation system. It can run different tests for debugging purposes. This software can also be used to synchronize all the activities spread across different computers and microprocessor.

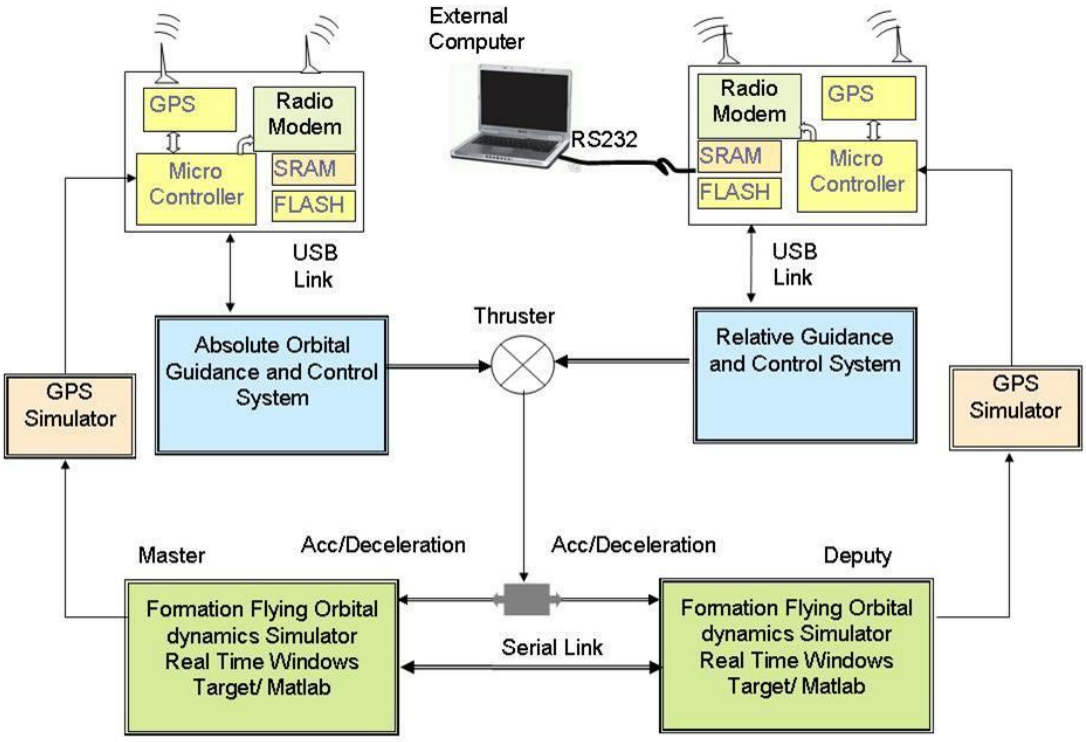


Figure 7.1: Hardware in the Close Loop Real Time Simulation Setup

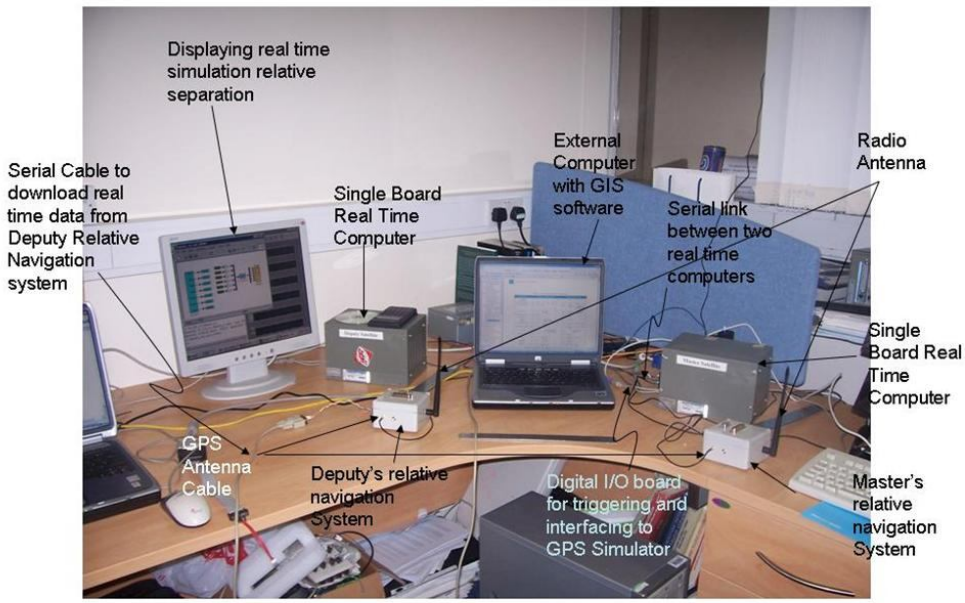
A complete setup for the real time hardware in the loop SFF test bed completed in this project is shown in figure 7.2. This picture shows two single board real time computers, running the master and deputy satellite's dynamics. Two relative navigation systems with GPS and radio antenna mounted on them. The serial link has been used between the real time computers to exchange their information in real time. Both are equipped with digital I/O cards that can be used for data exchange. GIS software triggers the simulation on the two real time computers running satellite formation flying dynamics. The absolute states and relative states at each simulation step are stored in deputy satellite. Deputy satellite not only stores, but also displays this information in real time. In the navigation embedded software two relative navigation systems are capturing GPS signals through RF interface, sorting and extracting raw information and the master satellite is transmitting this raw information to the deputy satellite for the calculation of relative states. For actual real time hardware in the loop simulation, these relative navigation systems would get their GPS signals from a GPS simulator (either hardware or software). To complete this test bed for closed loop hardware in the loop simulation, the development of guidance and control algorithm and implementation of navigation algorithms are required. Its different possible open loop uses are described in the following chapter.

At present the relative navigation system is synchronized to GPS time, but real time computers, running satellite formation flying models are not synchronized to the GPS time. This may affect the synchronization of the formation flying simulation and relative navigation algorithm in long run. The simulation and navigation update rate is set to 1 sec.

This test bed setup is useful to develop and test relative guidance and control algorithms. It can be used for tuning the controller for better performance. It can test the hardware components and technology in the loop. It can be used to study fuel consumption estimate. This test bed is designed to work only with GPS based relative navigation system. The setup of real time computer is ready to interface through their serial port (COM2). The RF output of GPS simulator (hardware based) can be connected directly to the antenna connector on the receiver.

There are limitations associated with this test bed. These are

- Only a GPS based relative navigation sensor can be used in this test bed.
- Only formations in LEO orbits can be tested on this test bed.
- In the RTSFF model only J2 and air drags were considered. To make the model more precise, other perturbations, like solar radiations, 3rd body interaction, and zonal harmonics can also be included in the RTSFF model.
- In the present test bed, the master satellite has only two serial ports, so it can transfer master satellite's absolute states to only two deputy satellite. In order to increase the number of satellites in the formation, it would be necessary to add more serial port cards into the master satellite's real time computer.
- The navigation processor used in the relative navigation system is a fixed point processor and for higher accuracies a floating point processor can be used.
- For full autonomous close loop real time hardware in the loop simulation, the attitude model of each satellite can be used in parallel to the RTSFF model.



A real time simulation test bed for spacecraft formation flying

Figure 7.2:

7.2 Open Loop Simulation Test Bed:

In this real time simulation setup, only the relative navigation system and external computer are used. The external computer will download prerecorded GPS raw measurement and ephemeris data with known relative states, on to the SDRAM (memory) of the navigation system. Nearly 47Mbytes of space on SDRAM is reserved for this purpose. In this setup relative navigation will pickup data from the SDRAM and will calculate absolute and relative states. The relative states calculated by the navigation system will be transmitted and logged in the external computer. These relative states are compared with the already known relative states to estimate the accuracy of the relative navigation algorithm. This setup can be used for rapid relative navigation software development and testing. In this setup the GPS receivers will not be used. This test bed setup is shown in figure 7.3

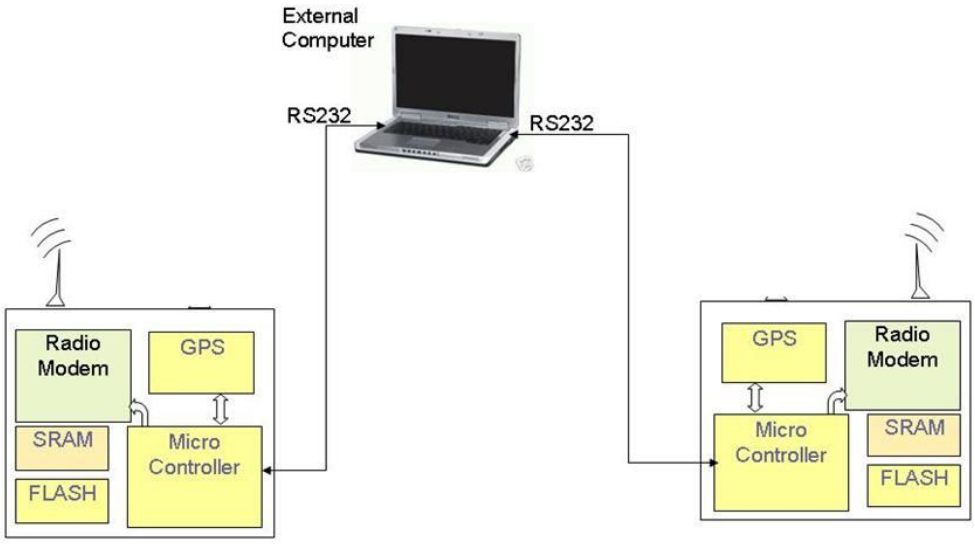


Figure 7.3 Open Loop Simulation Test Bed

This setup is used to initially assess the accuracy of relative navigation algorithms, to find a better set of initialization matrices for the Kalman filter. The user can repeat the simulation with the same input GPS raw data to see the affect of different parameters and algorithms on relative state accuracy. It can also be used to testify the accuracy of the relative navigation controller. For this purpose a set of GPS raw data from a RTHIL simulation is required.

7.3 Open Loop Simulation Test Bed with Hardware GPS Simulator in Loop:

In this test bed the Formation Flying Dynamic model, the GPS Simulator and relative navigation systems are connected in open loop. The tests can not be run without a GPS simulator. In the open loop simulation setup, guidance and control algorithms, thrusters, feedback hardware are not used. The block diagram for this setup is shown in figure 7.4. This simulation test bed needs at least a two channel GPS simulator. At the start of the simulation, the external computer issues trigger signals to both real time computers, running RTSFF model and also issues trigger signal to deputy relative navigation system. Deputy satellite then triggers the master relative navigation system to start simulation. As the simulation starts on

real time computers, they fed current absolute states (velocity and position) to the GPS simulator and also record that data in Matlab workspace. GPS simulator generates standard navigation messages as approved by the Navstar GPS [61]

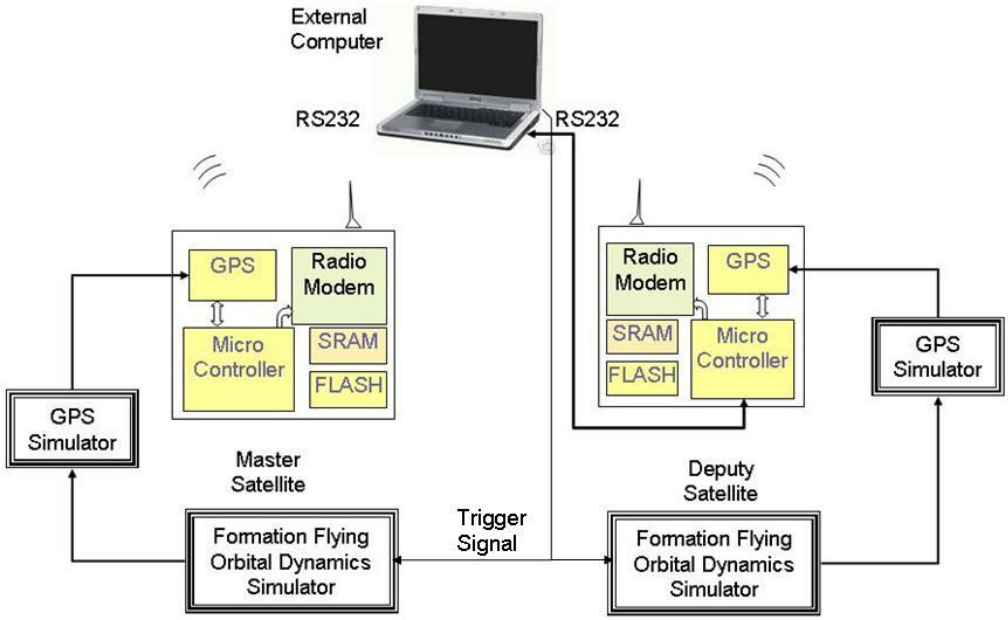


Figure 7.4: Open Loop Simulation Test Bed

GPS receivers acquire GPS signal, sort raw and ephemeris information, arrange them in message queues and transmit this information to other satellite to be used in relative navigation algorithm. These relative states are then compared with the relative states recorded by the satellite formation flying model. Software GPS simulator can also be used in place of hardware GPS simulator. This setup allows rerunning the real time simulation for the same input and formation flying scenario. This test bed is used for the verifications of relative navigation algorithm and identifies the effect of different parameters, gains, initialization matrix on relative states accuracy. All the relative states are logged into external computer.

7.4 Closed Loop Simulation Test Bed with Software GPS Simulator in Loop:

This is again a complete close loop real time simulation setup, but instead of a hardware GPS simulator, a software GPS simulator is used, like GPSLab by Accord. The block diagram for

this simulation setup is shown in figure 7.5. This software simulator uses current states of master and deputy satellites, measurement/error model and Almanac data to generate raw measurements. These raw measurements are transmitted on a serial port and fed directly to the navigation processor (S3C2410). The Master navigation system broadcasts its raw measurement data to the deputy satellite and the deputy navigation system calculates relative states. Relative guidance and control algorithms will generate error signals to the thrusters. Feedback thrust is fed to deputy satellite's dynamic model, through some analogue to digital converter. Updated states from master and deputy satellites are fed to the GPS simulator, which generates new set of raw measurements. Instead of actual thrusters, their transfer function can be used. This setup can be used for rapid initial controller development and testing. This setup gives less accurate results as compared to the setup discussed in section 6.3. As you will add more transfer functions of the hardware components in the loop, the results will be less accurate. The end results will carry the errors due to modeling errors in transfer functions. At the moment GPSLab software simulator is one channel simulator i.e. it takes input from one dynamic model and generates only one set of raw measurement. To get two sets of raw measurements on the basis of two different dynamic models, two GPS simulator should run at the same time on one computer. In their next version Accord is going to add multiple inputs dynamic model and multiple output sets of raw measurements.

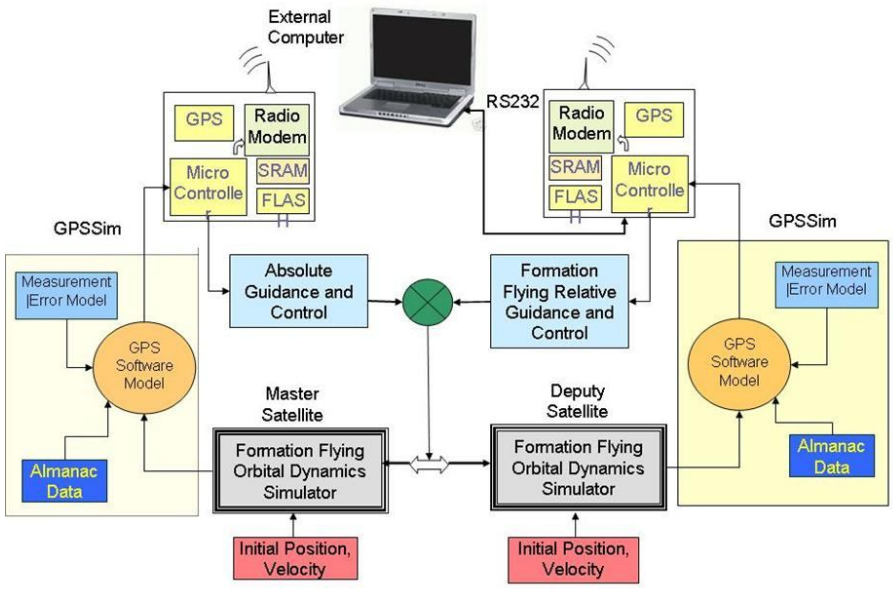


Figure 7.5: Close Loop Simulation Test Bed with Software GPS Simulator

7.5 Future Work/Recommendations:

The following recommendations are made to continue the research described in this thesis.

1. In the present RTHIL setup, all the timing activities on the navigation processor are synchronized with the GPS time, but the timing activities on real time computers running satellite formation flying models are not synchronized with the GPS 1PPS signal. Any timing mismatching in the long run will produce unpredictable results. So it is recommended to synchronize the real time computers time activities with GPS time by feeding GPS 1PPS to the real time computers and real time operating systems on these computers and use this 1PPS as timer tick.
2. At present the hardware design supports USB interface between GPS receiver and navigation processor (S3C2410) and also between external computer and navigation processor, the software device driver is only written for UART ports. As the data transfer on USB ports are much faster than on UART port, it is recommended as future work to write down device driver for USB and use USB interface to transfer data among GPS receiver, navigation processor and external computer.

3. Absolute and relative navigation algorithms developed in chapter 5, are not yet programmed in embedded C. It is recommended as future work to write these algorithms in embedded C and download them on navigation system. Tune initial matrices and gain for better convergence. Test these algorithms by running open loop simulation test.
4. Develop a guidance and control algorithm to complete the loop.
5. In the gravitational perturbation, only zonal harmonics were considered, which are a function of latitude. To make model more precise tesseral harmonics should be added as well. Third body interactions (lunar, solar) also effect each satellite.
6. Extend this setup for more than two satellites formation and experiment for centralized formation flying approach. In this case each radio modem on deputy satellites will broadcast data at different frequency to the master satellite. Similarly master navigation system will receive these data by hopping to different frequency channel.
7. Use data cache to store queues of local and remote raw information for their quick access.

In chapter 3 a real time formation flying model is developed. In chapter 5 a model of GPS constellation with one LEO satellite is developed. In chapter 5 absolute and relative navigation algorithms are developed. If these models and algorithms are combined together and GPS receiver's measurement and error models are added with them, it will give non real time software based close loop simulation setup. The schematic for this model is shown in figure 7.6. The SFF model will calculate absolute states of each satellite and relative states of deputy satellite. Absolute states will be passing on to the GPS software simulator. The GPS constellation model will calculate current true range of LEO satellites from each GPS satellite visible to the LEO satellites. Now add error model, white noise and true range to the measurement model will give raw measurements. Absolute and relative navigation algorithm will use this raw information and other navigation data to calculate absolute states of each LEO satellites and relative states of deputy satellite.

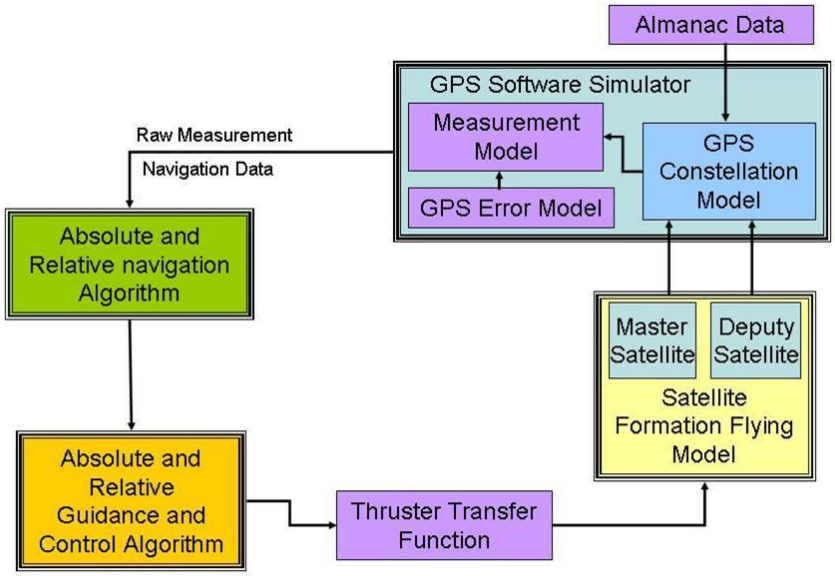


Figure 7.6: Non real time software based close loop simulation setup

Absolute and relative guidance algorithm will compare these absolute states and relative states with the reference states. On the basis of this comparison, control algorithm will generate error signal to the thruster model. Feedback thruster along three body axes will be passing on to the satellite formation flying model, which will update absolute and relative states. Simulation will continue in this manner. This non real time simulation setup will be a handy tool for quick relative navigation algorithm and relative controller development and assessment.

7.6 Conclusions:

A Real time hardware in the loop simulation test bed for LEO nano satellite formation flying has been developed. The purpose of this test bed is to test and estimate the performance of different hardware technologies, software algorithms and control algorithms. This RTHIL test bed is designed in a modular way, so that any part can be replaced by its hardware or software components. In this test bed different components are connected to each other in the same hierarchy as they are on the flight system. Although this test bed is designed to simulate different satellite formation flying configuration, a setup for decentralized approach is adopted for this project. The main components of this test bed are

- RTHIL satellite formation flying Model
- Single frequency GPS based relative navigation system
- RTOS based navigation software
- Graphical Interface Software

For the RTHIL setup different satellite formation flying models and equations were studied. All these models have some limitations associated with them and the assumptions used in their derivation introduce errors. These models do not fulfill RTHIL requirements. A RTHIL satellite formation flying model and algorithm was developed in terms of unit vectors. This RTHIL SFF model can be expanding easily and can simulate a formation consisting of two or more satellites. Initially these unit vectors were defined in a geocentric frame, but unit vectors can be defined in any earth centered inertial frame of reference without affecting them. By using this concept and RTHIL model another model is developed by defining unit vectors in perifocal coordinate system. This gave a difference model in terms of the difference of orbital elements. This difference model is then simplified for closely placed satellites (equation 3.49). This difference model is a very handy tool for the quick study of satellite formation flying. It gives the relative position of the deputy satellites in the master frame of reference.

The most troublesome perturbations for LEO satellites are non sphericity of earth and atmospheric drag. The affect of other perturbations, like third body interaction, solar radiation are very small and are ignored and left for future work. A gravity model proposed by Vinti [35] in geocentric frame has been used in this research. Only zonal harmonics are considered in this model. The expansion of a Vinti series is considered for up to J2 terms only. Rest of the terms can be added in future versions. Atmospheric drag perturbation in geocentric frame given in equation (3.69) is used. Analytical atmospheric model (equation 3.72), which consider the gravity and temperature variation along the altitude is used. For COWPOKE and the difference model, the variation of parameter method is used to include the perturbation affect of J2 and atmospheric drag.

These models were programmed in Matlab/Simulink and offline simulations were run for different formation flying scenario. The simulation results were compared, analyzed and RTHIL model was found to be quite satisfactory and robust for all scenarios. The Difference Model Short Form (DMSF) gives very reliable results. The error analysis shows that the DMSF model results were better than the COWPOKE model, especially when along track separation

and eccentricity increases. The RTHIL model has been downloaded onto a network of two single board real time computers. Data is transferred among them by using serial communication. The decentralized approach is used to demonstrate real time simulation. Simulation results for RTHIL model prove its suitability in RTHIL setups.

Different relative navigation sensors were studied by analysing the requirements of a LEO nano satellite. GPS was selected as the best navigation sensor, since it is small, low mass, compact, consumes low power and it can be used for absolute and relative navigation. The radio modem, AC4868 is selected for information exchange between satellites and S3C2410 and a 32bit RISC processor is used as a navigation sensor. Digital I/Os and serial interface are provided to connect it with other external devices. The PCB schematic and board layout is designed in Eagle software. **µC/OS-II** real time operating system is used for relative navigation development. An efficient, deterministic software structure for navigation system is developed. A novel approach for relative navigation in master frame of reference is presented. Algorithms for absolute and relative navigation are developed and its implementation in embedded microcontroller environment is left as future work. This navigation system provides in circuit programming facility, hence it is easy to modify and reprogram new code without opening the case.

The graphical user interface is a very important part of the real time hardware in the loop simulation setup. Graphical interface software (GIS) for this project has been developed in Matlab. This is used to initialize hardware components, to set different simulation options and to start, stop simulation. This software is also used to display simulation data in real time. It logs real time simulation results in a file, which can be used later for post processing. It can also be used to debug and test different parts of the navigation system.

There are a number of possible uses of the RTHIL test bed. It can be used either in open loop or in closed loop. The closed loop RTHIL setup is used to test the relative and absolute controller, whereas open loop RTHIL setups are used to test some software or the hardware part of the main system. Once different hardware components and software algorithms are tested then the closed loop RTHIL simulation can be run to test them in a complete closed loop simulation setup. Closed loop RTHIL simulation results are more reliable and have more preference over open loop results.

In summary in this research a very versatile, low cost, open ended, satellite formation flying test bed has been developed. The hardware and software has been designed in a modular

way so any hardware component can be replaced by its software transfer function and visa versa. Due to its low cost, its flexible and modular design structure, it can be used for a quick preliminary verification of different controller, navigation algorithms, and hardware technology. It can also be used to plan a formation flying mission, and to estimate fuel consumption.

References

1. Labrosse J. J, *uC/OS-II Real Time Kernel, Second Edition.* CMP Books, 2002.
2. Clohessy W.H. and Wiltshire R.S, *Terminal Guidance for Satellite Rendezvous.* Aerospace Sciences, 1960. **Vol 27**, p. 653.
3. Euler E. A.., and Shulman Y., *Second order solution to the Elliptical Rendezous Problem.* AIAA Journal, Volume 5, 1967: p. pp. 1033-1035.
4. Werlwas, R.W., *A New First Order Solution to the Relative Motion Problem with applications to Intercept and Rendezous.* PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, 1968.
5. Melton, R.G., *Relative Motion of Satellites in Elliptical Orbits,.* Advanced in the Austronautical Sciences, 1997. **Vol. 97, No 2.**: p. pp. 2075-2094.
6. Melton, R.G., *Time Explicit Representation of Relative Motion Between Elliptcal Orbits.* Journal of Guidance Control and Dynamics, 2000. **Vol. 23, No 4.**: p. pp. 604-610.
7. Schweighart S. Sedwick R, *A Perturbative Analysis of Geopotential Disturbances for satellite Cluster Formation Flying.* , IEEE, 2001.
8. Schweighart S, *Development and Analysis of a High Fidelity Linearized J2 Model for Satellite Formation Flying.* June 2001.
9. Alfriend K. T., Schaub H., and Gim D. W, *Gravitational perturbations, Nonlinearity and circular orbit Assumption Effects on Formation Flying of Satellites.* AAS Paper 00-258, 20-21 March 2000.
10. Sabol. C., McLaughlin C. A., and Luu K.K, *Meet the Cluster Orbits With Perturbations of Keplerian Elements (COWPOKE) Equations.* AAS/AIAA Space flight Mechanics Meeting, 9-13 Feb. 2003. **AAS 03-138**.
11. Oliver M., Takuji E., E Glenn L. E., and Sunney L., *A real time kinematic GPS sensor for Spacecraft relative Navigation* Aerospace Science and Technology, 30 July, 2002. **6**: p. 435-449.
12. Fortescue P., Stark J. and Swinerd G., *Spacecraft Systems Engineering , Third Edition, John Wiley & sons Inc.* 2003.
13. Zink M., *Defination of the TerraSAR-L Cartwheel Constellation.* ESA TS-SW-ESA-SY-0002, ESTECH, Noordwijk, 2003.

14. Chien S. E., Knight.R., Rabideau G., Sherwood R., Hansen E., Ortiviz A., Wilklow C., Wichman S., *Onboard Autonomy on the Three Corner Sat Mission.* Proceeedings of the 2001 International Symposium on Artificial Intelligence, Robotics and Automation for the Space, Montreal, Canada, June 2001.

15. Bleeker J. A. M. and ohers, *X-Ray Evolving Universe Spectroscopy - The XEUS Mission Summery.* SP-1242, ESA Publication Division, ESTEC, The Netherlands, , May 2000. **ISBN No. 92-9092-564-7**.

16. Kiraly Z., E.B., Busse F., Prof. Twiggs R. J., Prof. How J.P., *The orion Microsatellite: A Demonstartion of Formation Flying in Orbit.* 13th AIAA/USU Conference on Small Satellite, 1999. **SSC99-VI-8**.

17. Persson S., Jacobsson B., Gill E., *PRISMA- Demonstration Mission for Advanced Rendezvous and Formation Flying Technologies and Sensors.* 56th International Austronautical Congress, Japan, October 2005(Paper-IAC-05-B56B07).

18. Buss, F. D., *Precise Formation State Estimation in Low Earth Orbit using Carrier Differential GPS.* PhD Thesis, Stanford University, March 2003.

19. Twiggs R. and How J., *Orion: A microsatellite test bed for formation flying.* 12th AIAA/USU Conference on Small Satellites, 1998. **SSC98-XII-6**.

20. How J., Twiggs R., and Weidow D., *A Low Cost Demonstration of Formation Flying in Space using GPS.* 1998. **, AIAA-98-4398**.

21. Naffin D J. and Sukhatme G. S., *A test bed for Autonomous Formation Flying, Robotic Embedded System Laboratory.* Institute for Robotics and Intelligent Systems Technical Report IRIS-02-412**,** University of Southern California, 2002.

22. Miller D., and Saenz-Otero A., *SPHERES, A Test bed for Long Duration Satellite Formation Flying in Micro Gravity Conditions.* department of Aeronautics and Astronautics MIT, May 2002(AAS 00-110).

23. Ahmed, A., *Terrestrial Planet Finder Mission (TPF), Formation Flying Technology, Formation Control Testbed (FCT).* TPF Expo, JPL, 2003.

24. Leitner J. *A Hardware in the loop Testbed for Spacecraft Formation Flying Applications*. in *NASA Goddart Space Flight Center*. 2001.

25. Vallado D. A., and McClain W. D., *Fundamentals of Astrodynamics and Applications* Space Technology Series, McGraw ill, 1997.

26. Catlin K. A, *Modeling Formation Flight with J2 Perturbation using the COWPOKE Equations.* Dec. 2003.
27. Catlin K. A, *Satellite Formation Flying Using the Perturbed COWPOKE Equations.* AIAA Regional Student Conference, 2004.
28. Balaji S. K and Tatnall A, *Precise Modelling of Relative Motion for Formation Flying Spacecraft.* International Astronautical Fedration, 2003.
29. Balaji S. K & Tatnall A, *System Design Issues of Formation Flying Spacecraft.* IEEE Aerospace conference, 2003.
30. Balaji S. K & Tatnall A, *Relative Trajectory Analysis of Dissimilar Formation Flying Spacecraft.* Advanced in Astronautics Sciences, volume 114, 2003.
31. Ahmed R. Tatnall A. Balaji S. K, *Closed Form Analytical Solutions For Spacecraft Formation Flying Trajectory Analysis.* 4th Workshop on Satellite Constellations and Formation Flying, Feb 2005.
32. Chobotove V. A, *Orbital Mechanics second edition.* American Institute of Aeronautics and Astronautics Inc, 1996.
33. Bate R. R. Muller D.D. & White J. E, *Fundamentals of Astrodynamics.* New York Dover Publication Inc, 1971.
34. GRACE Gravity Model EIGEN-GRACE01S, G., July 2003.
35. Vinti J. P. Der G. J. & Bonavito N. L, *Orbital and Celestial Mechanics.* Progress in Astronautics and Aeronautics, 1998. **volume 177**.
36. Regan F. J., *Re entry Vehicle Dynamics, New York,American Institute of Aeronautics and Astronautics, Inc.* 2002.
37. Thomas L. Liu. Michael K. Louise D. M. Alec D. G. and Brian E. G, *Theoratical aspects of nanoparticle Electric Propulsion.* 2006.
38. Sidi M. J., *Spacecraft Dynamics and Control, A practical Engineering Approach.* Cambridge University Press, 1997.
39. Roy A. E, *Orbital Motion, 3rd edtion, Student text.* 1991. **Adam Hilger, Bristol**
40. SIMULINK User Guide, T.M, Mathworks.
41. Han H., S.A., *Geometry and Control of Satellite Formations.* Proceedings of the American Control Conference, Chicago, Illinoise, June, 2000.

42. Ahmed R. Tatnall A. Balaji S K, *A Comparison of Spacecraft Formation Flying Models and use in Real Time Hardware in the Loop Simulations.* 4th Workshop on Satellite Constellations and Formation Flying, Feb 2005.
43. JUKI-745E Petium with LCD/CRT & Ethernet ISA Bus SBC ver. 2.x, U.s.M., 1999.
44. Real Time Windows Target User Guide, The Mathworks.
45. Busse F. D, How J. P., and Simpson J., *Demonstration of Adaptive Extended Kalman Filter for Low Earth Orbit Formation Estimation Using CDGPS*.
46. Rizos C., *Principles and Practice of GPS Surveying Version 1.1.* 1999.
47. Jennifer L. Ruiz & Charles H. Frey, *Geosynchronous Satellite use of GPS.* ION GNSS 18th International Technical Meeting of the Satellite Division, Long Beach CA, Sep, 2005.
48. http://www.navcen.uscg.gov/gps/current/current.alm..
49. http://www.murata.com/articles/ta0641.pdf.
50. Egemen I, Palmer P., amd Hashida Y., *Precise relative orbit determination of low earth orbit Formation Flights using GPS Pseudo range and carrier phase Measurement*., Recent Advances in Space Technologies, June, 2007
51. Busse F. D. How J. P, *Real Time Experimental Demonstration of Precise decentralized relative navigation for formation flying spacecraft.* AIAA, 2002.
52. Aerocomm, *Developer Kit User's Manual, Version 3.3.* 2004.
53. Deitel & Deitel, *C How to Program* Prentice Hall, 2001. **3rd Edition**.
54. Labrosse J. J., *uC/OS-II and ARM Processor, Application Note AN-1014 Rev C. .* 2006.
55. U-BLOX, *ANTARIS Protocol Specification Manual* ( ).
56. Dr. Bao J., *Fundamentals of Global positioning System Receiver, software approach.* John Wiley and Sons, Inc, 2000.
57. http://babbage.cs.qc.edu/IEEE-754/32bit.html.
58. Hofmann-Wellenhof & Bernhard, *Global positioning system : theory and practice* Wien : Springer Verlag, 1993.
59. Welch G. and Bishop G., *An Introduction to the Kalman Filter.* University of North Carolina at Chapel Hill, July 2006.
60. Mohinder S. G., Lawrence. R. W., Angus P. A, *Global Positioning Systems, Inertial Navigation, and Integration.* Jan,2002(John Wiley & Sons, Inc).
61. *Navstar GPS Space Segment/Navigation User Interfaces, Revision D.* March, 2006. **IS-GPS-200**,GPS joint program office, El Segundo, CA.

62. Folta D. and Hawkins, A., 'Results of NASA's first Autonomous Formation Flying Experiment: Earth- Observing-1 (EO-1)', AIAA/AAS Astrodynamics Specialist Conference, Monterey, California, 2002, AIAA 2002-4743.

63. Moreira A., Krieger G., Hajnsek I., Riegger S. and Settelmeyer E.,.'TanDEM-X: A TerraSAR-X Add-On Satellite for Single PSAR interferometry', Geoscience and Remote Sensing Symposium, 2004, IEEE international volume 2.

64. Aung M., Ahmed A., Wette M., Scharf D., Tien J., Purcell G., Regehr M., Landin B., 'An Overview of Formation Flying Technology Development for the Terrestrial Plant Finder Mission', Aerospace Conference 2004,

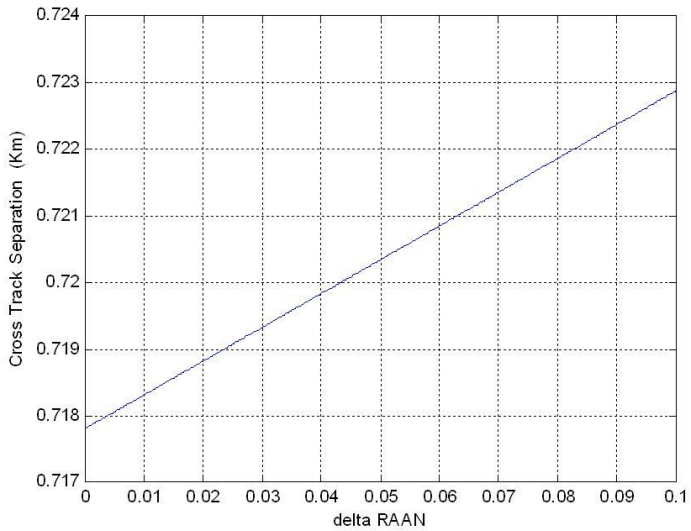
65. Roberts P. C. E., Bowling T. S., Hobbs S. E., 'MUSTANG: A Technology Demonstrator for Formation Flying and Distributed Systems Technologies in Space', Space Research Center, Cranfield University 2002.

66. Roberts J. A., and Hobbs S. E., 'Satellite Formation Flying For An Interferometry Mission', PhD Thesis, Cranfield University 2005.

Appendix A

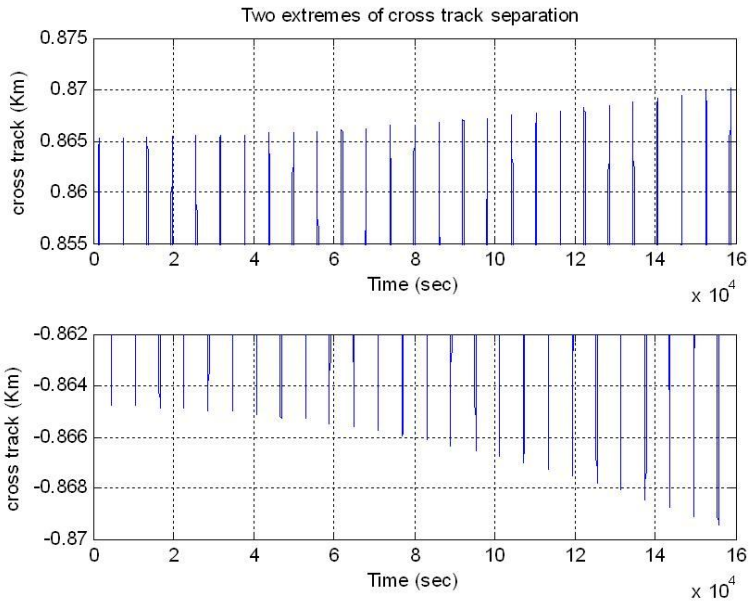
The plot A.1 is showing the variations in cross track separation as $\delta\Omega$ varies from 0 to 0.1 rad. For variations up to 0.02 in $\delta\Omega$, the corresponding error in cross track separation is about 1 meter.

Variation in Cross track separation DMSF model with variations in RAAN



Figure(A.1): Variation in cross track separation as $\delta\Omega$ varies

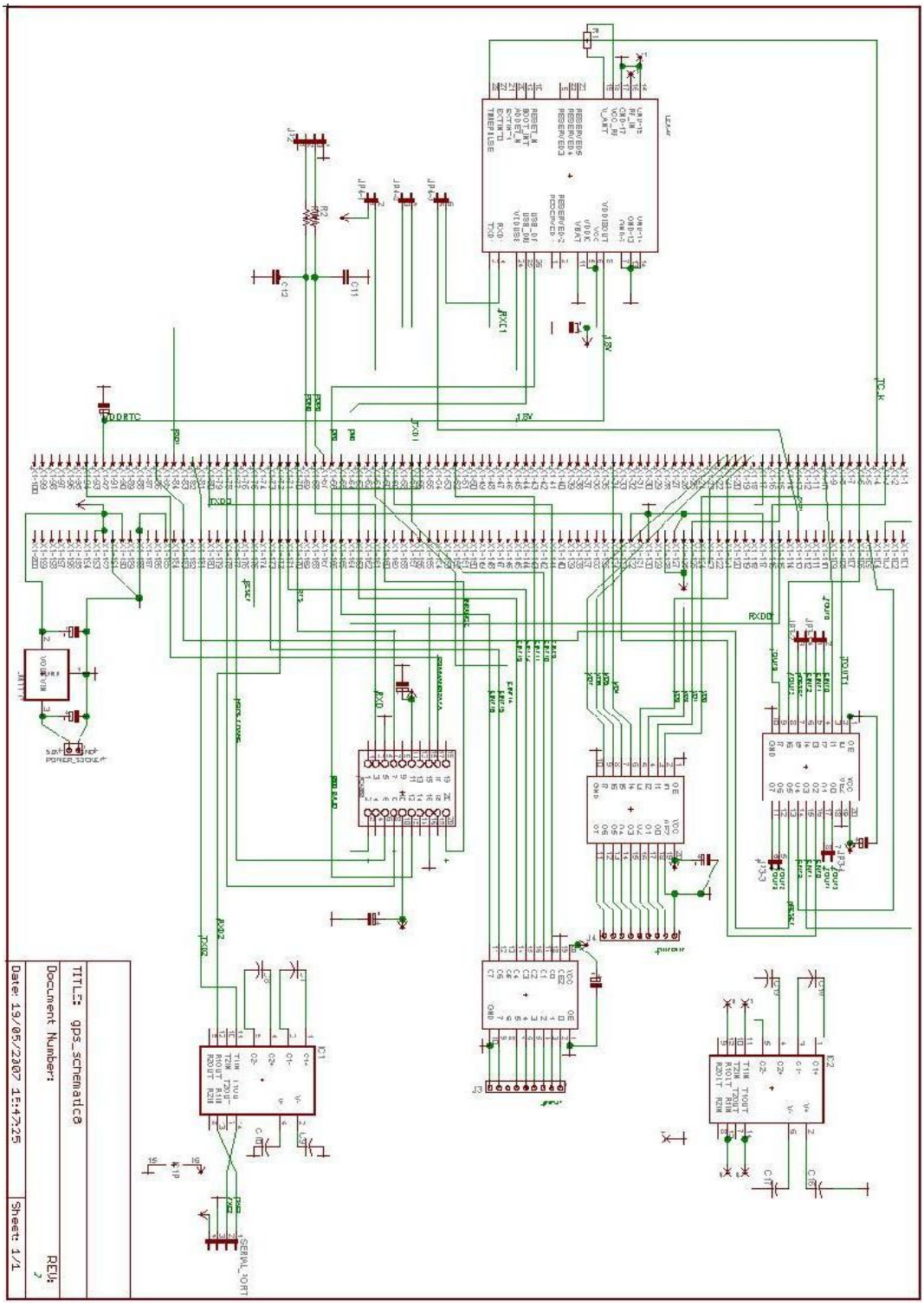
Zoomed view showing increase in amplitude of cross track separation for the case 2 simulation presented in chapter 3



Figure(A.2): Two extremes of cross track separation showing increase in the separation

Appendix B

This is the circuit diagram for relative navigation system. Details can be seen in the attached CD



Appendix C

Microprocessor based RTOS port subroutines

OSCtxSw:

```
  Stmfd     sp!,{lr}
  Stmfd    sp!,{r0-r12,lr}
  mrs         r4,cpsr
  stmfd    sp!,{r4}
  mrs         r4,spsr
  stmfd    sp!,{r4}
 ldr      r4,=OSTCBCur
 ldr      r5,[r4]
 str      sp,[r5]
 dr       r6,=OSTCBHighRdy
 ldr      r6,[r6]
 str      r6,[r4]
 ldr      r7,=OSPrioCur
 ldr      r8,=OSPrioHighRdy
 ldr      r8,[r8]
 str      r8,[r7]
 ldr      sp,[r6]
 Ldmfd   sp!,{r4}
  Msr spsr_cxfs,r4
 Ldmfd  sp!,{r4}
  msr     cpsr_cxfs,r4
 ldmfd   sp!,{r0-r12,lr,PC}
```

IntCtxSw:

```
      mov r1,#0
      str r1,[r0]
      ldmfd sp!,{r0-r3,r12,lr}
      stmfd sp!,{r0-r3}
      mov r1,sp
      add sp,sp,#16
      sub r2,lr,#4
      mrs r3,spsr
     msr     cpsr_c,#(NOINT|SVCMODE)
      stmfd sp!,{r2}
      stmfd sp!,{r4-r12,lr}
      mov r4,r1
      mov r5,r3
      ldmfd r4!,{r0-r3}
      stmfd sp!,{r0-r3}
      stmfd sp!,{r5}
      mrs r4,spsr
      stmfd sp!,{r4}
```

```
ldr r4,=OSPrioCur
ldr r5,=OSPrioHighRdy
ldrb r5,[r5]
strb r5,[r4]
ldr r4,=OSTCBCur
ldr r5,[r4]
str sp,[r5]
ldr r6,=OSTCBHighRdy
ldr r6,[r6]
ldr sp,[r6]
str r6,[r4]
ldmfd sp!,{r4}
msr SPSR_cxsf,r4
ldmfd sp!,{r4}
msr CPSR_cxsf,r4
ldmfd sp!,{r0-r12,lr,pc}
```

OSStartHighRdy:

```
 bl      OSTaskSwHook
ldr      r4,=OSRunning
 mov     r5,#1
strb     r5,[r4]
ldr      r4,=OSTCBHighRdy
ldr       r4,[r4]
ldr       sp,[r4]
ldmfd  sp!,{r4}
 msr     spsr_cxsf,r4
ldmfd    sp!,{r4}
 msr     cpsr_cxsf,r4
ldmfd    sp!,{r0-r12,lr,PC}
```

Appendix D

Data verification analysis

UBX binary packet containing raw information

B5 62 02 10 80 00 F9 7C D4 16 AD 05 05 00 6F 82 87 81 B6 F3 9A 41 52 89 00 32 E3 83 74 41 48 9D A3 45 06 07 23 00 00 00 00 00 00 00 00 00 15 16 64 C5 A4 4B 75 41 E4 21 4B 44 0C 04 12 03 74 37 4C 1A 92 BB 96 41 13 4F 25 E3 33 05 75 41 22 FD AC 45 07 07 2B 00 9A 14 A9 9F 55 3B 9C 41 88 43 7D 75 41 97 B7 1B 45 01 07 2A 00 39 54 CC AC A3 F4 99 41 C8 5C 0B 6C 43 0B 76 41 18 3C E6 43 0E 07 23 00 48 BE

The extracted information from above binary data are

B5 62 02 10 = ;Message Header

80 00 = 128 ; Total no of bytes following

F9 7C D4 16=383024377 ; Time of the Week (TOW)

AD 05= 1453 ; Week No

05 = 05 ; No of Satellites following

6F 82 87 81 B6 F3 9A 41 =113044896.38233350 ; Carrier Measurement of Satellite_1

52 89 00 32 E3 83 74 41 =21511731.125130960 ; Pseudo Measurement of Satellite_1

48 9D A3 45=5235.6601562500000 ; Doppler Measurement of Satellite_1

06=6 ; Satellite No_1

07 23 00

00 00 00 00 00 00 00 00 =0 ; Carrier Measurement of Satellite_2

15 16 64 C5 A4 4B 75 41 =22329932.336935120 ; Pseudo Measurement of Satellite_2

E4 21 4B 44 =812.52954101562500 ; Doppler Measurement of Satellite_2

0C=12 ; Satellite No_2

04 12 03

74 37 4C 1A 92 BB 96 41 =95347846.574430280 ; Carrier Measurement of Satellite_3

13 4F 25 E3 33 05 75 41 =22041406.196608614 ; Pseudo Measurement of Satellite_3

22 FD AC 45 =5535.6416015625000 ; Doppler Measurement of Satellite_3

07 =07 ; Satellite NO_3

07 2B 00

9A 14 A9 9F 55 3B 9C 41 =118412647.91511765 ; Carrier M4asurement of Satellite_4

89 9C 45 88 43 7D 75 41 =22533176.516994986 ; Pseudo Measurement of Satellite_4

97 B7 1B 45 =2491.4743652343750 ; Doppler Measurement of Satellite_4

01 =01 ; Satellite NO_4

07 2A 00

39 54 CC AC A3 F4 99 41=108865771.19954003 ; Carrier M4asurement of Satellite_5

C8 5C 0B 6C 43 0B 76 41 =23114806.752774030 ; Pseudo Measurement of Satellite_5

18 3C E6 43 =460.46948242187500 ; Doppler Measurement of Satellite_5

0E =14 ; Satellite NO_5

07 23 00

48 BE =Checksum

Data Transmitted to the external computer

TOW	+383024377
WEEK NO	1453
Satellite No _1	+6
Pseudo Range_1	+21511731.125130
Carrier Measurement_1	+113044896.382333
Doppler Shift_1	+5235.660156
Satellite No_2	+12
Pseudo Range_2	+22329932.336935
Carrier Measurement_2	0.0
Doppler Shift_2	+812.529541
Satellite No_3	7
Pseudo Range_3	+22041406.196608
Carrier Measurement_3	+95347846.574430
Doppler Shift_3	+5535.641601
Satellite No_4	1
Pseudo Range_4	+22533176.516994
Carrier Measurement_4	+118412647.915117

Doppler Shift_4	+2491.474365
Satellite No_5	14
Pseudo Range_5	+23114806.752774
Carrier Measurement_5	+108865771.199540
Doppler Shift_5	+460.469482

Appendix E

A dump of raw measurements data in external computer:

```
+164350498              %Time of the Week
+1475                   %Week No.
+17                     %Satellite No.
+22467844.782696        %Pseudo Measurement
+116493912.458854       %Carrier Phase Measurement
+395.530517             %Doppler shift Measurement
+14                     %Satellite No.
+21308249.176906        %Pseudo Measurement
+111975612.601607       %Carrier Phase Measurement
+5413.85937             %Doppler shift Measurement
+9                      %Satellite No.
+18798453.315611        %Pseudo Measurement
+128719477.619729       %Carrier Phase Measurement
+1543.78613             %Doppler shift Measurement

+164350750
+1475
+17
+22467825.717846
+116493812.739601
+394.251922
+14
+21307989.517617
+111974248.407363
+5412.956054
+9
+18798379.81672
+128719089.33272
+1541.205810
+164350998
+1475
+17
+22467807.177581
+116493714.697273
+395.804443
+14
+21307734.2630
+111972905.979262
+5412.499511
+9
+18798306.418861
+128718706.717766
+1540.930297
+164351250
```

```
+1475
+17
+22467788.78133
+116493615.203865
+395.318664
+14
+21307474.343931
+111971542.21880
+5413.922363
+9
+18798232.596093
+128718318.357155
+1541.926147
+164351498
+1475
+17
+22467769.607105
+116493517.189079
+395.391662
+14
+21307218.977311
+111970199.643355
+5413.571777
+9
+18798159.924816
+128717936.85717
+1542.331054
+164351750
+1475
+17
+22467750.638939
+116493417.546943
+395.972045
+14
+21306959.283340
+111968835.526229
+5414.15625
+9
+18798085.604079
+128717547.614937
+1541.648925
+164351998
+1475
+17
+22467731.757231
+116493319.521141
+394.466918
+14
```

```
+21306704.38290
+111967493.120162
+5412.135742
```

Appendix F

```
%  A input file for Satellite Formation Flying Model
%  This input file generates input data for three Models
%     RTHIL Model, COWPOKE Model, and Hill's Model
%  It is simulation Input File for Case 1
%  Master and Deputy Satellites are moving in similar
%  orbital plane with different orbital eccentricity
%  A File developed by RIAZ AHMED, Astronautics Group
%    University of Southampton
clear all;
format long;

%initial cond for hill's equation
Re=6378.1363;        %Radius of earth (Km)
%rmin=7178.1361;              %Perigee distance
j2=0.0010826269;
mu=398600.4415;             %Gravitational parameter of earth
(Km^3/sec^2)
Mu=398600.4415;             %Gravitational parameter of earth
(Km^3/sec^2)
we=0.00417807462229;        %Earth rotational rate deg/sec


pert=1;             % A flag to switch ON and switch OFF perturbation

em=0.00001;
ed=0.000056;

%*************************************
am=6823.00029074;
ad=6823.00029074;
rmin1=am*(1-em);
rmin2=ad*(1-ed);


%*************************************
n1=sqrt(mu/am^3);               %angular rate of satellite cluster
(mean motion)
n2=sqrt(mu/ad^3);               %angular rate of satellite cluster
(mean motion)


im=87.001418402*pi/180;
%i2=0.785+0.01;
id=87.011418402*pi/180;

Im=87.001418402*pi/180;
Id=87.011418402*pi/180;
%Id=pi/2+0.01;

OMGm=44.941186437*pi/180;
OMGd=45.041186437*pi/180;
%OMGd=4.712+0.01;
Omega1=OMGm;
```

```
Omega2=OMGd;

omegam=0.001421029*pi/180;
omegad=0.011421029*pi/180;
omega1=omegam;
omega2=omegad;

    %Em_0=acos((em+cos(thetam0))/(1+em*cos(thetam0)));
    %Mm_0=Em_0-em*sin(Em_0);
    %Mm_0=0.000045020*pi/180;
    %Md_0=0.006762950*pi/180;
    %M01=1.751;
    %M0=1.751;   %rad

    thetam0=pi/2;
    thetad0=pi/2-0.001;
    Em_0=acos((em+cos(thetam0))/(1+em*cos(thetam0)));
    Ed_0=acos((ed+cos(thetad0))/(1+ed*cos(thetad0)));
    Mm_0=Em_0-em*sin(Em_0);
    Md_0=Ed_0-ed*sin(Ed_0);
 %*************************************************************
 %Realtive velocity of satellite at time t  (Km/sec)

 %Relative position of satellite at time t (Km)

 %*************************************************************
Ci_PQW=[cos(OMGm)*cos(omegam)-sin(omegam)*cos(Im)*sin(OMGm), -
cos(OMGm)*sin(omegam)-cos(omegam)*cos(Im)*sin(OMGm),
sin(OMGm)*sin(Im);sin(OMGm)*cos(omegam)+cos(Im)*sin(omegam)*cos(OMGm),
-sin(OMGm)*sin(omegam)+cos(omegam)*cos(Im)*cos(OMGm), -
sin(Im)*cos(OMGm);sin(Im)*sin(omegam), sin(Im)*cos(omegam), cos(Im)];
Loc_to_Geo=[cos(OMGm)*cos(omegam+thetam0)-
sin(thetam0)*cos(Im)*sin(OMGm), -cos(OMGm)*sin(thetam0)-
cos(thetam0)*cos(Im)*sin(OMGm),
sin(OMGm)*sin(Im);sin(OMGm)*cos(thetam0)+cos(Im)*sin(thetam0)*cos(OMGm)
, -sin(OMGm)*sin(thetam0)+cos(thetam0)*cos(Im)*cos(OMGm), -
sin(Im)*cos(OMGm);sin(Im)*sin(thetam0), sin(Im)*cos(thetam0), cos(Im)];
rm=am*(1-em^2)/(1+em*cos(thetam0));
Rm_vector=[rm,0,0];
Vo_0=sqrt(Mu/(am*(1-em^2)))*[-sin(thetam0); (cos(thetam0)+em); 0];% in
PQR coordinate frame
Ro_0=[rm*cos(thetam0) rm*sin(thetam0) 0]';

Rgm_0=[(cos(OMGm)*cos(omegam)-
sin(omegam)*cos(Im)*sin(OMGm))*(rm*cos(thetam0))+(-
cos(OMGm)*sin(omegam)-cos(omegam)*cos(Im)*sin(OMGm))*(rm*sin(thetam0)),

(sin(OMGm)*cos(omegam)+cos(Im)*sin(omegam)*cos(OMGm))*(rm*cos(thetam0))
+(-
sin(OMGm)*sin(omegam)+cos(omegam)*cos(Im)*cos(OMGm))*(rm*sin(thetam0)),

(sin(Im)*sin(omegam))*(rm*cos(thetam0))+(sin(Im)*cos(omegam))*(rm*sin(t
hetam0))];

Vgm_0=Ci_PQW*Vo_0;
```

```
% Transformation matrix from deputy polar Frame(PQW) of reference to
Geocentric frame of reference.
Tr_to_geo=[cos(OMGd)*cos(omegad)-sin(omegad)*cos(Id)*sin(OMGd), -
cos(OMGd)*sin(omegad)-cos(omegad)*cos(Id)*sin(OMGd), sin(OMGd)*sin(Id);
    sin(OMGd)*cos(omegad)+cos(Id)*sin(omegad)*cos(OMGd), -
sin(OMGd)*sin(omegad)+cos(omegad)*cos(Id)*cos(OMGd), -
sin(Id)*cos(OMGd);
    sin(Id)*sin(omegad), sin(Id)*cos(omegad), cos(Id)];

% Transformation matrix from Geocentric to Local RSW master frame of
% reference
Geo_to_RSW=[cos(omegam+thetam0)*cos(OMGm)-
sin(omegam+thetam0)*cos(Im)*sin(OMGm),
cos(omegam+thetam0)*sin(OMGm)+sin(omegam+thetam0)*cos(Im)*cos(OMGm),sin
(omegam+thetam0)*sin(Im);-sin(omegam+thetam0)*cos(OMGm)-
cos(omegam+thetam0)*cos(Im)*sin(OMGm), -
sin(omegam+thetam0)*sin(OMGm)+cos(omegam+thetam0)*cos(Im)*cos(OMGm),cos
(omegam+thetam0)*sin(Im);sin(Im)*sin(OMGm),-sin(Im)*cos(OMGm),cos(Im)];

rd=ad*(1-ed^2)/(1+ed*cos(thetad0));
Vd_0=sqrt(Mu/(ad*(1-ed^2)))*[-sin(thetad0); (cos(thetad0)+ed); 0];% in
PQR coordinate frame
Rd_0=[rd*cos(thetad0) rd*sin(thetad0) 0]';
Rgd_0=Tr_to_geo*Rd_0;
Vgd_0=Tr_to_geo*Vd_0;

 %Relative position of satellite at time t (Km)
 rel_position=Geo_to_RSW*Rgd_0-Rm_vector';
 rotation=sqrt(Mu/am^3);
 x1=rel_position(1);
 y1=rel_position(2);
 z1=rel_position(3);
 velocity_m_RSW=Geo_to_RSW*Vgm_0;
 velocity_d_RSW=Geo_to_RSW*Vgd_0;
 rot_correction_d=[-
rotation*rel_position(2);rotation*(rel_position(1)+rm);0];
 rot_correction_m=[0;rotation*rm;0];

 %Realtive velocity of satellite at time t  (Km/sec)
 rel_velocity_d=velocity_d_RSW-rot_correction_m;
 rel_velocity_m=velocity_m_RSW-rot_correction_m;

 rel_velocity=-rel_velocity_m+rel_velocity_d;
 xdot1=rel_velocity(1);
 ydot1=rel_velocity(2);
 zdot1=rel_velocity(3);


rd1=sqrt(Rgd_0(1)^2+Rgd_0(2)^2+Rgd_0(3)^2);
Time_P=(sqrt(Mu)/(2*pi))*(1/(sqrt(am))^3);

Vxgd=Vgd_0(1)+rotation*(Rgd_0(3)-Rgd_0(2));
Vygd=Vgd_0(2)+rotation*(Rgd_0(1)-Rgd_0(3));
Vzgd=Vgd_0(3)+rotation*(Rgd_0(2)-Rgd_0(1));
```

```
   %*********************************
   %data for COWPOKE Equation
   da=ad-am;    %Km
   de=(ed-em);
   di=(id-im);      %rad
   dOMG0=(OMGd-OMGm); %rad
   domega0=(omegad-omegam);   %rad
   dM0=(Md_0-Mm_0);  %rad
   p1=am*(1-em^2);
   p2=ad*(1-ed^2);
 %*************************

 %*********************************
 %data forCOWPOKE Equation
 dM1=(sqrt(Mu/am^3)*(1+.75*(Re^2*j2*sqrt(1+em^2)/(am*(1-em^2))^2)*(2-
3*(sin(im))^2)));
 dM2=(sqrt(Mu/(am+da)^3)*(1+.75*(Re^2*j2*sqrt(1+(em+de)^2)/((am+da)*(1-
(em+de)^2))^2)*(2-3*(sin((im+di)))^2)));
 %dOmega1=-1.5*sqrt(mu/a1^3)*(Re^2*j2*cos(im)/(a1*(1-em^2))^2);
 %dOmega2=-1.5*sqrt(mu/(a1+da)^3)*(Re^2*j2*cos((im+di))/((a1+da)*(1-
(em+de)^2))^2);
 domega1=.75*sqrt(mu/am^3)*(Re^2*j2*(4-5*(sin(im))^2)/(am*(1-em^2))^2);
 domega2=.75*sqrt(mu/(am+da)^3)*(Re^2*j2*(4-
5*(sin((im+di))^2))/((am+da)*(1-(em+de)^2))^2);
 domegaxy1=[sqrt(Mu/am^3)]*[1+(3.0/4.0)*(2-
3*(sin(im))^2)*Re*Re*j2*sqrt(1-em^2)/((am*(1-em^2))^2)];
 domegaxy2=[sqrt(Mu/(am+da)^3)]*[1+(3.0/4.0)*(2-
3*(sin((im+di)))^2)*Re*Re*j2*sqrt(1-(em+de)^2)/(((am+da)*(1-
(em+de)^2))^2)];
 delta_th=dM0+2*(em+de)*sin(Mm_0+dM0)-2*em*sin(Mm_0);

 %*********************************
 %j2 effect on right ascention of ascending node
 Odot1=(-3*n1*j2*Re^2)*cos(im)/(2*(am*(1-em^2))^2);
 Odot2=(-3*n2*j2*Re^2)*cos(id)/(2*(ad*(1-ed^2))^2);
 %j2 effect on argument of perigee
 wdot1=(3*n1*j2*Re^2)*(4-5*(sin(im))^2)/(4*am^2*(1-em^2)^2);
 wdot2=(3*n2*j2*Re^2)*(4-5*(sin(id))^2)/(4*ad^2*(1-ed^2)^2);
 %j2 effect on mean anomaly
 Mdot1=(-3*n1*j2*sqrt(1-em^2)*Re^2)*(3*(sin(im))^2-2)/(4*am^2*(1-
em^2)^2);
 Mdot2=(-3*n2*j2*sqrt(1-ed^2)*Re^2)*(3*(sin(id))^2-2)/(4*ad^2*(1-
ed^2)^2);
 wxy1=n1+Mdot1;
 wz1=n1+Mdot1+wdot1;
 wxy2=n2+Mdot2;
 wz2=n2+Mdot2+wdot2;

 %effects for differnt inclination
 dOdot1=Odot1-Odot1;
 dwdot1=wdot1-wdot1;
 dMdot1=Mdot1-Mdot1;

 dOdot2=Odot2-Odot1;
```

```
 dwdot2=wdot2-wdot1;
 dMdot2=Mdot2-Mdot1;


 %Drag Perturbation Data


 b=3.139*10^-7;                % 1/m
 Mol_mass=28.9664              % Kg/Kmole
 R=287.05;                     % J/Kg.K
 g0=9.806;                     % m/sec^2

 zi1=(rmin1-Re)*1000;          % m     -- apogee for Master
 zi2=(rmin2-Re)*1000;          % m     -- apogee for Deputy
 Tmi1=1499.2;                  % K--- Temperature at zi1
 Tmi2=1487.4;                  % K--- Temperature at zi2


 Lz1=1.7/1000;                 % K/m
 Lz2=2.6/1000;                 % K/m
 %z=5600;
 m1=1000;                           %Mass of Master Satellite in Kg
 m2=1000;                           %Mass of Deputy Satellite in Kg
 A1=0.000001;                       %Cross sectional Area
 A2=0.000001;                       %Cross sectional Area
 Cd1=2.0000;                        %Drag Coefficient of Master
Satellite
 Cd2=2.001;                         %Drag Coefficient of Deputy
Satellite
 BC1= m1/(Cd1*A1);                  %Ballastic coefficient
 BC2= m2/(Cd2*A2);                  %Ballastic coefficient
 RoE=(7.292115854530769e-005);      %(Rad/Sec)Earth on complete
rotation in 23h 56min 4.0905Sec
 den1=9.754478849948109e-013;
 den2=5.845748533997969e-012;
 %p=den*[[(Lz*(z-zi1)/Tmi)+1]^-[(g0/(R*Lz))*[(R*Lz/g0)+1+b*[(Tmi/Lz)-
zi1]]]]*exp(g0*b*(z-zi1)/(R*Lz))%atmospheric density

%Realtive velocity of satellite at time t  (Km/sec)

disp('      omegam,             omegad,             thetam0,
thetad0            Um')
%[omegam*180/pi,omegad*180/pi,thetam0*180/pi,thetad0*180/pi,(omegam*180
/pi+thetam0*180/pi)]
[omegam,omegad,thetam0*180/pi,thetad0*180/pi,(omegam+thetam0)*180/pi]
disp('      am            ad');
[am,ad]
```

A Matlab function that initializes serial port, and parallel port:

```
function initializeSPort_Callback(hObject, eventdata, handles)
% hObject    handle to initializeSPort (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global baudrate;
global  PORT
global  s
global  prt
global  fid;
%baudrate=115200;
if strcmp(s.status,'closed')
   fopen(s);
end
fprintf(1,'Port is %s \n',PORT)
%s=serial(PORT,'BaudRate',baudrate,'Terminator','CR');
%fopen(s)
set(s,'BaudRate',baudrate)
brate=num2str(baudrate);
%fprintf(1,'serial port is open now and baud rate is %d %s\n',baudrate,PORT)
msgbox(['Serial port  ' PORT ' is initialized successfully at ' brate], ' Initialized')
%fclose(s)
%stop_serial_Callback(hObject, eventdata, handles)
%disp_Callback(hObject, eventdata, handles)
set(handles.edit_baud,'String',baudrate)
set(handles.edit_port,'String',PORT)
set(handles.main_win,'Name','Nav_Interface3')

dio = digitalio('parallel', 1); %assigns parallel port
addline(dio, 0:7, 'out'); % lines are assigned
putvalue(dio,[0 1 1 1 1 1 1 1]); %Logical value, check to see if button 1 is up or down, if
        portval =1 then up, if portval =0 then down.
delete(dio);


/*****************************************************************************
;*      A part of rt_stub.c file used to trigger the real time simulation
;*       on real time computers
;*       EXECUTE
;*       execute the model
;*       Input: tid = task id
;*       Output:    step size
;*****************************************************************************/

double __cdecl Execute(int tid)
{ int error=ERR_OK;
```

```
static unsigned char fg1;
static short fg=0;
/*************************************************************************
          Waiting for trigger

*************************************************************************/
fg1=(_inp(0x379))&0x40;

if((fg1!=64)&&(fg==0))     //If input at PA0 is low, keep real timer resetting
{
          rt_InitTimingEngine(S);
          if ((rt_InitTimingEngine(S)) != NULL)
          return(ERR_SIMNOTINIT);
          fg=0;
          return(0);
}


/*************************************************************************/
/* TID==0 means the base rate */
if (tid==0)
 {fg=1;
/* first process any messages and upload any buffer data */
#ifdef EXT_MODE
  rt_MsgServerWork(S);
  if (!IsSharedBufferFull())     // we can upload only if buffer is not full
    rt_UploadServerWork(S);
#endif // EXT_MODE
/* if MULTITASKING is defined, this is the multitasking base rate */
#ifdef MULTITASKING
  ssSetSolverStopTime(S,rt_UpdateDiscreteEvents(S));
  MdlOutputs(FIRST_TID);
#ifdef EXT_MODE
  UploadCheckTrigger();
  UploadBufAddTimePoint(S,FIRST_TID);
#endif // EXT_MODE
   MdlUpdate(FIRST_TID);
  if (ssGetSampleTime(S,0) == CONTINUOUS_SAMPLE_TIME)
    rt_UpdateContinuousStates(S);
  else
    rt_UpdateDiscreteTaskTime(S,0);
#if FIRST_TID == 1
    rt_UpdateDiscreteTaskTime(S,1);
#endif
/* if MULTITASKING is not defined, this is the whole model */

#else // MULTITASKING
  ssSetSolverStopTime(S,rt_GetNextSampleHit());
```

```
  MdlOutputs(0);
#ifdef EXT_MODE
  {
   int i;
   UploadCheckTrigger();
   for (i=0; i<NUMST; i++)
     if (ssIsSampleHit(S,i,0)) UploadBufAddTimePoint(S,i);
  }
#endif / / EXT_MODE
  MdlUpdate(0);
  rt_UpdateDiscreteTaskSampleHits(S);
  if (ssGetSampleTime(S,0) == CONTINUOUS_SAMPLE_TIME)
    rt_UpdateContinuousStates(S);
#endif / / MULTITASKING
/* test if the execution should be stopped */
  if  (ssGetStopRequested(S)  || (    ssGetT(S)  >=  ssGetTFinal(S)  &&  ssGetTFinal(S)  !=
          RUN_FOREVER ) )
   {
    StopTimers();
    MdlTerminate();

#ifdef EXT_MODE
    UploadPrepareForFinalFlush();
    rt_UploadServerWork(S);
    SendShutdownMsg();
#endif / / EXT_MODE
   }
/* check for end of the trigger */
#ifdef EXT_MODE
  UploadCheckEndTrigger();
#endif
 }
/* TID!=0 means multitasking subrate */
#ifdef MULTITASKING
else
 {
  MdlOutputs(tid);
#ifdef EXT_MODE
  UploadBufAddTimePoint(S,tid);
#endif / / EXT_MODE
  MdlUpdate(tid);
  rt_UpdateDiscreteTaskTime(S,tid);
 }
#endif / / MULTITASKING
return(ssGetSampleTime(S,0));
}

* ****************************************************************************/
```

```
 *     C-Sfunction for serial interface between two real time computers
 *     Written by RIAZ  Ahmed
 *     Astronautics Group
 *     School of Engineering
 *     University of Southampton
 * ************************************************************************/

#define S_FUNCTION_NAME  serial_test_3
#define S_FUNCTION_LEVEL 2
#define PORT 0x3F8

#include "simstruc.h"
#include <math.h>
#include <stdlib.h>
#include <conio.h>
#include <dos.h>
#include <stdio.h>
#include <string.h>

int byt;
char anal_rem[10];
char   anal[90];
double  xmgc,ymgc,zmgc;

static void mdlInitializeSizes(SimStruct *S)
{

   ssSetNumSFcnParams(S, 0);  /* Number of expected parameters */
   if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
      return;
   }

   ssSetNumContStates(S, 0);
   ssSetNumDiscStates(S, 0);
   if (!ssSetNumOutputPorts(S, 2)) return;
   ssSetOutputPortWidth(S, 0, 8);
   ssSetOutputPortWidth(S, 1, 8);
   ssSetNumSampleTimes(S, 1);
   ssSetNumRWork(S, 0);
   ssSetNumIWork(S, 0);
   ssSetNumPWork(S, 0);
   ssSetNumModes(S, 0);
   ssSetNumNonsampledZCs(S, 0);
   ssSetOptions(S, 0);
}
```

```
static void mdlInitializeSampleTimes(SimStruct *S)
{
   ssSetSampleTime(S, 0, 1);
   ssSetOffsetTime(S, 0, 0.0);
}

#define MDL_INITIALIZE_CONDITIONS   /* Change to #undef to remove function */
#if defined(MDL_INITIALIZE_CONDITIONS)
 static void mdlInitializeConditions(SimStruct *S)
 {

 }
#endif /* MDL_INITIALIZE_CONDITIONS */

#define MDL_START  /* Change to #undef to remove function */
#if defined(MDL_START)

 /*Function:mdlStart =======================================*/

 static void mdlStart(SimStruct *S)
 {

 }
#endif /*  MDL_START */

/*Function:mdlOutputs =======================================*/

static void mdlOutputs(SimStruct *S, int_T tid)
{
//real_T *u = (const real_T*) ssGetInputPortSignal(S,0);
real_T      *y = ssGetOutputPortSignal(S,0);
int i,len,j,ln,k,l;
double  dig;
char lowbyte,start;
char    data[15],data2[15];
double   dpar[20],Rmgg;
long    data1[10],dig_rem[10];
j=0;
i=0;
_outp(PORT+4,0x02);   //Set RTS pin in MCR
do{
lowbyte=_inp(PORT+5);
if(lowbyte & 1)
start=_inp(PORT+0);
/* start='s';   */
//i++;
}while(start != 's');
_outp(PORT+0,'a');
```

```
i=0;
do
{
while(!((_inp(PORT+5)& 0x01)));
anal[i]=_inp(PORT+0);
i++;
}while(anal[i-1] != 'e');
/***************************************/
i=0;
k=0;
j=0;
l=0;

 do
 {
   do
   {   if(anal[i]=='e')break;
       data[j]=anal[i];
       j++;
       i++;
    }while(anal[i]!='Q');
    i++;
   data[j]=0x0;
   data1[k]=atol(data);
       do
       {if(anal[i]=='e')break;
       data2[l]=anal[i];
       i++;
       l++;
       }while(l<5);
   data2[l]=0x0;
   dig_rem[k]=atol(data2);
   k++;
   j=0;
   l=0;
 }while((anal[i]!='e')&& k<7);


/***************************************/
_outp(PORT+4,0x00);    //Set RTS pin in MCR
dpar[0]=(data1[0])+(dig_rem[0]/100000.0);
if(data1[0]<0)dpar[0]=(data1[0])-(dig_rem[0]/100000.0);
xmgc=(dpar[0]/1000.0);
 dpar[1]=data1[1]+(dig_rem[1]/100000.0);
if(data1[1]<0)dpar[1]=data1[1]-(dig_rem[1]/100000.0);
ymgc=(dpar[1]/1000.0);
dpar[2]=data1[2]+(dig_rem[2]/100000.0);
```

```
if(data1[2]<0)dpar[2]=data1[2]-(dig_rem[2]/100000.0);
zmgc=(dpar[2]/1000.0);
dpar[3]=data1[3]+(dig_rem[3]/100000.0);
if(data1[3]<0)dpar[3]=data1[3]-(dig_rem[3]/100000.0);
dpar[3]=(dpar[3]/100000.0);
dpar[4]=data1[4]+(dig_rem[4]/100000.0);
if(data1[4]<0)dpar[4]=data1[4]-(dig_rem[4]/100000.0);
dpar[4]=(dpar[4]/100000.0);
dpar[5]=data1[5]+(dig_rem[5]/100000.0);
if(data1[5]<0)dpar[5]=data1[5]-(dig_rem[5]/100000.0);
dpar[5]=(dpar[5]/100000.0);


y[0]=xmgc;
y[1]=ymgc;
y[2]=zmgc;
y[3]=dpar[3];
y[4]=dpar[4];
y[5]=dpar[5];

y[6]=sqrt(pow(xmgc,2)+pow(ymgc,2)+pow(zmgc,2));
y[7]=(i-1);
y[8]=data1[0];
y[9]=data1[1];
y[10]=data1[2];
y[11]=data1[3];
y[12]=data1[4];
y[13]=data1[5];

}

#define MDL_UPDATE  /* Change to #undef to remove function */
#if defined(MDL_UPDATE)

  /*Function:mdlUpdate ====================================== */

  static void mdlUpdate(SimStruct *S, int_T tid)
  {
  }
#endif /* MDL_UPDATE */



#define MDL_DERIVATIVES  /* Change to #undef to remove function */
#if defined(MDL_DERIVATIVES)
  /*Function:mdlDerivatives ================================== */

  static void mdlDerivatives(SimStruct *S)
```

```
 {
 }
#endif /* MDL_DERIVATIVES */

/*Function:mdlTerminate ================================== */

static void mdlTerminate(SimStruct *S)
{
 _outp(PORT+4,0x02);   //Set RTS pin in MCR
}

/*==============================*
 * Required S-function trailer *
 *==============================*/

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```

Appendix G

```
/*********************************************************/
/*          File Name  interactive.c                     */
/*        Messages and Interactive Routines              */
/*        Deceloped by RIAZ  AHMED                       */
/*        ASTRO  GROUP                                   */
/*        UNIVERSITY OF SOUTHAMPTON                      */
/*********************************************************/
 #include"includes.h"                /* uC/OS interface */
#include"2410addr.h"
 void  start_message(void);
 void  interactive(void);
void  start_message(void)
{                int dly;
  int urt=0;
UART_trans("*=======================================\n",urt);
for(dly=0;dly<100000;dly++);
UART_trans("*========= Riaz Ahmed   Astro Group  ============*\n",urt);
for(dly=0;dly<100000;dly++);
UART_trans("*=======================================*\n",urt);
for(dly=0;dly<100000;dly++);
UART_trans("*========== University of Southampton ===========*\n",urt);
for(dly=0;dly<100000;dly++);
UART_trans("*===== Spcaecraft Formation Flying Test Bed ======*\n",urt);
for(dly=0;dly<10000;dly++);
UART_trans("*================uCOS-II Running..============*\n",urt);
for(dly=0;dly<10000;dly++);  }
void interactive(void)
 {
  char  opt;
int          cnt;
while(1)
{
      cnt=0;
      while(!(rUTRSTAT2 & 0x1));
      opt=RdURXH2();                  /* Receive option letter from GIS*/
      while(!(rUTRSTAT2 & 0x1));
      RdURXH2();                      /* Flush the receiver port   */
      for(cnt=0;cnt<4;cnt++){
      while(!(rUTRSTAT0 & 0x2));     /* Transmit receive option Letter to the master satellite   */
      WrUTXH0(opt);}
       /*  GPS_STOP();
       opt='S';
      for(cnt=0;cnt<10;cnt++){
      while(!(rUTRSTAT0 & 0x2));
      WrUTXH0('A');}
      */
        if(opt =='B' || opt =='b')       /* radio_mod_settings();     */
      {
      while(!(rUTRSTAT2 & 0x2));
      WrUTXH2('R');                   /* Handshaking message   */
```

```
while(!(rUTRSTAT2 & 0x2));
WrUTXH2(0x0D);
radio_modem();
}
else if(opt =='M' || opt =='m')     /* Radio Modem EEPROM Read */
{
while(!(rUTRSTAT2 & 0x2));
 WrUTXH2('R');                        /* Handshaking message   */
while(!(rUTRSTAT2 & 0x2));
WrUTXH2(0x0D);
ROM_ADDRESS_READ();
}
else if(opt =='N' || opt =='n')     /* Radio modem Range Test */
{
while(!(rUTRSTAT2 & 0x2));
WrUTXH2('R');                          /* Handshaking message   */
 while(!(rUTRSTAT2 & 0x2));
 WrUTXH2(0x0D);
IN_RANGE_TEST();
}
else if(opt =='C' || opt =='c')     /* GPS Programming Mode */
{
while(!(rUTRSTAT2 & 0x2));
WrUTXH2('G');                              /* Handshaking message   */
while(!(rUTRSTAT2 & 0x2));
WrUTXH2(0x0D);
gps_cnfg();
CNFG_SAVE();
}
else if(opt =='S' || opt =='s')     /*SimuLation Mode   */
{
while(!(rUTRSTAT2 & 0x2));
WrUTXH2('A');                              /* Handshaking message   */
while(!(rUTRSTAT2 & 0x2));
WrUTXH2(0x0D);
/*   GPS_FORCED();            */
while(!(rUTRSTAT2 & 0x2));
WrUTXH2('S');
while(!(rUTRSTAT2 & 0x2));
WrUTXH2(0x0D);
/*uart_printf("\n Simulation Mode \n");   */
 SIM_MODE=1;                        /* Set Simulation option   */
/*   GPS_START();*/
 /*   SimMesg();   */
/*     disp_mess();  */
break;
/*GPS_UBX();*/
}
else if(opt =='E' || opt =='e')
{
uart_printf("\n Good Bye \n");
/*  Power Down Options */
}
```

```
    else if(opt =='D' || opt =='d')            /*  Display Mode  */
    {
    while(!(rUTRSTAT2 & 0x2));
    WrUTXH2('Z');                             /* Handshaking message   */
     while(!(rUTRSTAT2 & 0x2));
     WrUTXH2(0x0D);
    DISP_MODE=1;                             /* Set display option and Flag   */
    /*       disp_mess();   */
     break;
    }
}  /*  End of While Loop  */
}
```

Matlab function that accepts messages from the navigation code and display acknowledge message

```
function sercallback(obj,event,handles)
%fprintf(1,'I am in calling function Sercallback \n')
%out=fscanf(s)
TOW=fgets(obj)
if strcmp(TOW(1),'G')
set(handles.figure1,'Name','GPS Programming       Programming Phase');
return;
end
if strcmp(TOW(1),'K')
%  set(handles.main_win,'Name','Nav_Interface2--- GPS PROGRAMMING
SUCCESS');
 msgbox('GPS Receiver has been programmed successfully','ACK SUCCESS')
return;
end
if strcmp(TOW(1),'a')
set(handles.figure1,'Name','GPS Programming       Comunication Link
Esrablished');
return;
end
```

IN RANGE_TEST C- subroutine:

```
/*********************************************************/
  /*    IN Range Test   */
 /*********************************************************/
void IN_RANGE_TEST(void)
  {
     INT16U  cnt;
     INT8U   IN_RNG;
     INT8U   str0[10];
     INT8U   Par;
     cnt=0;
  while(1)
   {
    if(cnt==10000)
     {  cnt=0;
```

```
     if((rGPGDAT&0x4000) && (IN_RANGE_FLAGE==0) )
/*       if((rGPGDAT&0x4000) )   */
       {IN_RNG=1;
        IN_RANGE_FLAGE=1;
       Par='I';
       itoa(IN_RNG,str0,&Par);
       }
   else  if(!(rGPGDAT&0x4000) && (IN_RANGE_FLAGE==1) )
/*       else   */
      {
       IN_RNG=0;
       Par='I';
        IN_RANGE_FLAGE=0;
       itoa(IN_RNG,str0,&Par);
       }
     }
      cnt++;
   }               /*   End of While  */

  }                  /*   END of Function  */
/*****************************************************************************/
```

Matlab function that receives message from the navigation system and display appropriate message:

```
function RadioReplycallback(obj,event,handles)
%fprintf(1,'I am in calling function Sercallback \n')
%out=fscanf(s)
TOW=fgets(obj);
firstch=TOW(1);

   if strcmp(firstch,'I')
       TOW(1)=' ';
       secondch=TOW(2);
       if strcmp(secondch,'0')
           TOW
     msgbox('      Deputy Satellite is inRange        ','In Range')
        else
        TOW
msgbox('  Deputy Satellite is out of Range   ','Out of range','warn')
       end
   else if  strcmp(firstch,'V')
               TOW(1)=' ';
        str2num(TOW)
       end
   end
   end
```

Differencing:

Differencing is one the effective and commonly used method to eliminate or at least reduces the biases and errors. There are three differencing processing techniques.

Single Difference: Single differencing can be between satellite, between receiver and between epochs. Differencing the observations between the satellites by a receiver will eliminate receiver clock error. Differencing the observations between receivers for a GPS satellite will eliminate satellite clock error and differencing between epochs generally eliminate integer ambiguity.

Double Difference: Differencing between satellite and then between receivers or vice versa will eliminate satellite and receiver clock errors. It also reduces orbital error.

Triple Difference: Differencing double difference observable between epochs. All errors eliminated during single and double differencing processing are also eliminated during triple differencing. When used for carrier beat phase, it eliminated initial integer ambiguity.