Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

A Simple and Precise Approach to Position and Velocity Estimation of Low Earth Orbit Satellites

Pierre D. Beaulne and Ishuwa C. Sikaneta

Defence R&D Canada – Ottawa
TECHNICAL MEMORANDUM
DRDC Ottawa TM 2005-250
December 2005

Canada

A Simple and Precise Approach to Position and Velocity Estimation of Low Earth Orbit Satellites

Pierre D. Beaulne and Ishuwa C. Sikaneta

Defence R&D Canada – Ottawa

Technical Memorandum

DRDC Ottawa TM 2005-250

December 2005

Abstract

This technical memorandum presents a procedure for calculating a satellite's state vector (position and velocity), at arbitrary times on its orbit and with a precision of up to 10cm, from a more sparse set of state vectors (measured or predicted) provided by the satellite itself or by its ground station. No orbit propagation is necessary. While very precise, the procedure's accuracy is limited to the accuracy of the input state vectors. The first part of the memo deals with time systems important in astrodynamic applications, different coordinate systems used to describe satellite state vectors, and the various coordinate transformations necessary for the task. The second part describes the use of Hermite polynomials to interpolate (upsample) new state vectors in the appropriately transformed coordinate system. In the final part, the precision of the procedure is verified by decimating an ERS-2 state vector set and using the sparse set to upsample new state vectors at the original ERS-2 rate. This interpolated set is then compared to the original measured set, and precisions of up to 10cm are observed. Finally, as an example, a set of RADARSAT-1 state vectors are upsampled to 10 Hz from their natural rate of $1/480$ Hz.

Résumé

Ce mémorandum présente une technique pour calculer le vecteur d'état (position et vitesse vectorielles) d'un satellite aux instants chosis sur son orbite à partir d'un ensemble plus clairsemé de vecteurs d'état (prédits ou mesurés) fournis par le satellite ou par sa station au sol. Des résultats, précis a 10 cm près, sont obtenus sans avoir besoin d'utiliser un modèle de propagation d'orbite. Tant précis que soient ces résultats, leur exactitude est limitée par celle des vecteurs d'état originaux. La première partie de ce rapport adresse les systèmes de temps importants en astrodynamique, les différents systèmes de coordonnées spatiales qui spécifient les vecteurs d'état, ainsi que les diverses transformations de coordonnées necéssaires pour accomplir cette tâche. La seconde partie explique comment utiliser les polynomiaux d'Hermite pour interpoler les vecteurs d'état, après les avoir transformés au système de coordonnées spatiales approprié. Ensuite, la précision de la procédure est vérifiée en décimant un ensemble de vecteurs d'état en provenance du satellite ERS-2 et en utilisant ensuite cet ensemble réduit pour interpoler de nouveaux vecteurs d'état aux mêmes temps que l'ensemble original. L'ensemble qui résulte de l'interpolation est ensuite comparé à l'ensemble original et des précisions à l'ordre de 10 cm ou mieux sont observées. Finalement, comme exemple de la technique, un ensemble de vecteurs d'état du satellite RADARSAT-1 sont sur-échantillonnés à un taux de 10 Hz á partir de leur taux naturel de $1/480$ Hz.

Executive summary

The calculation of the geographic position (latitude/longitude) of a target on the earth from Spaceborne Synthetic Aperture Radar (SAR) measurements requires that the satellite position and velocity be precisely known at each pulse transmission point. This can be accomplished by using an orbit propagator. However, using the state vectors available from the satellite platform (either predicted or measured and/or refined on the ground) is simpler, more direct and can produce very precise results. The accuracy of the results is however only as good as that of the input state vectors. This memo presents a procedure to perform interpolation (upsampling) of satellite state vectors at arbitrary times with 10cm precision.

Precise timekeeping is critical to this task; to that end the first part of this memo provides an overview of time systems important in astrodynamic applications. Calculations of the satellite geometry are best performed in an Earth Centred Earth Fixed (ECEF) coordinate system, while state vectors are often given in an Earth Centred Inertial (ECI) system. A discussion of the necessary coordinate systems, the conversions between them and the time systems needed to do so, follows. Finally, a procedure using Hermite polynomials to upsample the state vectors to the desired times is provided and its precision tested.

To verify the procedure's precision, a set of ERS-2 state vectors (provided every 30 seconds) is downsampled to the RADARSAT-1 rate (every 8 minutes) by keeping every 16th state vector. The Hermite polynomial interpolation procedure is then used on this sparse set to upsample it back to a rate of one per 30 seconds. The interpolated results are then compared to the original measured ERS-2 state vectors to check the precision of the upsampling. Finally, an example is provided on upsampling RADARSAT-1 state vectors (at $1/480$ Hz - every 8 minutes) to 10 Hz. The MATLAB© code used for coordinate transformations and Hermite polynomial interpolation is listed as an appendix.

Pierre D. Beaulne and Ishuwa C. Sikaneta; 2005; A Simple and Precise Approach to Position and Velocity Estimation of Low Earth Orbit Satellites;
DRDC Ottawa TM 2005-250; Defence R&D Canada – Ottawa.

Sommaire

Afin de calculer la position géographique (latitude/longitude) d'une cible terrestre à partir de données collectées par un Radar à Ouverture Synthétique (ROS) spatial, les vecteurs position et vitesse du satellite doivent être précisément connus à chaque point de transmission d'impulse. Un modèle de propagation d'orbite peut accomplir cette tâche. L'utilisation de vecteurs d'état fournis par la plateforme orbitale (qu'ils soient prédits ou mesurés et/ou traités au sol), par contre, permet une approche directe qui est plus simple et qui produit des résultats précis. L'exactitude des résultats, cependant, ne peut pas surpasser celle des vecteurs d'état originaux. Ce mémorandum présente une méthode pour effectuer une interpolation, précise à 10 cm près, de vecteurs d'état spatiaux à quelque temps arbitraire.

La précision et l'exactitude du chronométrage sont essentielles au succès de cette tâche ; à cette fin la première partie de ce rapport présente un aperçu des différents systèmes de temps utilisés en astrodynamique. Les calculs géométriques d'un satellite en orbite sont plus facilement effectués dans un système de coordonnées spatiales cartésien géocentrique, et qui est en rotation avec la terre. Les vecteurs d'état sont cependant souvent exprimés dans un sytème cartésien fixé par rapport aux étoiles (qui n'est donc pas en rotation avec la terre). Alors la seconde partie de ce rapport fournit une discussion des différents systèmes de coordonnées spatiales ainsi que les conversions nécessaires pour effectuer les transformations entre eux. Enfin, une technique de sur-échantillonnage de vecteurs d'état, utilisant une interpolation polynomiale d'Hermite, est présentée et sa précision evaluée.

Pour vérifier la précision de cette technique, un ensemble de vecteurs d'état de ERS-2 (un échantillon chaque 30 secondes) est décimé au taux d'échantillonnage de RADARSAT-1 (un échantillon chaque 8 minutes) en retenant seulement chaque seizième échantillon. L'ensemble ERS-2 décimé est ensuite échantillonné à son taux original avec la technique d'interpolation polynomiale Hermite. Les résultats de l'interpolation sont ensuite comparés à l'ensemble ERS-2 original pour vérifier la précision de cette approche. Enfin, un exemple de sur-échantillonnage de vecteurs d'état de RADARSAT-1 (de $1/480$ Hz à 10 Hz) est présenté. Les fonctions MATLAB© utilisées pour les transformations de coordonnées ainsi que pour l'interpolation d'Hermite sont incluses dans les annexes.

Pierre D. Beaulne and Ishuwa C. Sikaneta; 2005; A Simple and Precise Approach to Position and Velocity Estimation of Low Earth Orbit Satellites; DRDC Ottawa TM 2005-250; R & D pour la défense Canada – Ottawa.

Table of contents

List of tables

List of figures

1 Introduction

An accurate estimate of the position and velocity of a radar satellite is of great interest to, and is likely necessary for, applications such as bistatic SAR, repeat pass interferometry and precise geolocation and ortho-rectification of satellite SAR imagery; and this is by no means an exhaustive list.

This document outlines how to estimate, with up to 10cm precision, the position and velocity of a satellite at arbitrary times using sampled state vectors (a state vector represents the position and velocity components of a satellite at a particular time, expressed in Earth centered Cartesian coordinates). Although only tested on nearly circular Low Earth Orbit (LEO) satellites in this document, this same technique can in principle be applied to any satellite for which accurate state vectors are recorded or predicted. In this memorandum, we use state vectors from two LEO SAR satellites (both orbit at slightly less than 800 km altitude), the Canadian RADARSAT-1 and the European ERS-2, to demonstrate the procedure.

An orbit propagator can be used to estimate satellite positions with great accuracy [1, 2]. This approach, however, is more complicated and can impose an additional computational burden. Orbit propagation, if not done properly, can also introduce numerical innacuracies, as it involves multiple coordinate transformations as well as the solution of complex dynamical equations. As an alternative, it is simpler and more efficient to directly make use of state vectors, be they predicted by an orbit propagator or directly measured by a satellite or its ground station. This document proposes the use of high accuracy state vectors collected from measurement stations on the ground. These measurements are made every eight minutes for RADARSAT-1, and every 30 seconds ERS-2, and provide accurate position and velocity estimates of the satellite at precise sample times.

For RADARSAT-1, these state vectors are provided in the so-called Geographic Earth centred Inertial (GEI) coordinate system (also called Earth Centered Inertial - ECI). The method proposed for upsampling the sample points to a frequency adequate for SAR applications works more accurately for coordinates provided in the Earth Centred Earth Fixed (ECEF) system (also called Earth Centered Rotating - ECR). Thus a transformation of coordinate systems is required. The first part of this document discusses this coordinate transformation after presenting a short primer on time systems of interest in astrodynamic applications.

The second part of this document outlines a method to upsample the state vectors by using Hermite polynomials. To demonstrate the precision of the upsampling, a set of ERS-2 state vectors measured at $1/30$ Hz is downsampled to the Radarsat state vector sampling frequency of $1/480$ Hz. This is done by simply keeping every 16^{th} sample point. This depleted set is then upsampled to $1/30$ Hz using the Hermite polynomial approach, and the estimated vectors are compared with the measured vectors.

Finally, an example of upsampling RADARSAT-1 state vectors to 10 Hz is provided. This example demonstrates the required coordinate transformations as well as the Hermite polynomial interpolation using Matlab code. The code is listed in the appendix.

2 Time Scales for Astrodynamic Objects

2.1 Background

Time is a fundamental quantity in almost any branch of science (to say nothing of everyday life), but it is especially critical in any astrodynamics problem. To fully understand the coordinate transformations that follow, an overview of different time systems will be helpful.

To have a practical time system useful in scientific applications, a precise, repeatable time interval is required. In addition, it should be based on some physical phenomenon that can be readily measured. Time units that appear natural to man are all based on astronomical phenomena: the motion of the earth about the sun and itself, the motion of the moon about the earth, etc. Until relatively recently, the most accurate time measurements were derived from these fundamental cycles. However, these cycles can be problematic if a high degree of precision is required. There are ambiguities in reference points for rotation or revolution measurements. For example, the earth's rotation suffers from irregularities and drift due to precession and nutation (the same applies to all rotating bodies), meaning that the year is not precisely expressed as an integer number of days, or seconds. As such, familiar calendar measures of time are not very useful in astrodynamic computations. To solve these and other problems, a variety of time systems, scales and calendars have been developed [3, 4, 5, 6]. The most important of these are discussed below.

There are many different time scales, but they can be broadly grouped into 3 categories. The first includes time scales based on the rotation of the earth (accounting or not for anomalies), such as various solar and sidereal times, and Universal Time

(UT). The second, dynamical time, is the independent variable in the equations of motion of the observed celestial object. Elapsed time can be deduced from the mathematical description of the object's observed motion. The third and most precise group is atomic time, which is based on a specific electron transition in the Cesium-133 atom, emitting a photon of known wavelength. The SI second is defined as a fixed number of theses wavelengths. Only the first and third categories will be reviewed in any detail here.

2.1.1 Celestial Coordinates and the Vernal Equinox

To better understand reference points for various time astronomical scales, it is useful to review coordinates on the Celestial Sphere, a hypothetical sphere of infinite radius centred on the earth. The North and South Celestial poles are just the Earth poles extended into space (i.e. the earth's spin axis), while the celestial equatorial plane is the same as Earth's equatorial plane. The local celestial meridian is a great circle passing through the poles and a point directly over an observer.

A second important fundamental plane is the ecliptic, the path of the Earth's orbit about the sun (or alternatively, the sun's path through the sky as seen from Earth over the course of a year). The celestial equatorial and ecliptic planes are co-centered and oriented 23.5 degrees apart, owing to the tilt of the Earth's rotation axis. They intersect in a line called the line of equinoxes, which passes through the centre of the sphere and defines two points at opposite positions on the orbit (the vernal and autumnal equinoxes - at these points the length of night and day are equal at all locations on Earth).

Objects on the celestial sphere are located in a similar fashion to the familiar latitude/longitude system for positions on the Earth. A point's angular location above the celestial equator is called the declination (analogous to latitude on Earth). The quantity analogous to longitude is the hour angle, the angle (in the celestial equatorial plane) between the object and the local celestial meridian. Like the Greenwich meridian, however the local celestial meridian rotates. To define an 'hour angle' in inertial space, the angle, now called the right ascension (RA) is measured from the vernal equinox point, (symbol Υ, also called the first point of Aries, since the sun rose in the constellation Aries on the vernal equinox at the time of its naming). Note that Υ is not fixed; due to the precession of Earth's orbit, it will move along the celestial equator, completing a full cycle every 25900 years.

2.1.2 Solar Versus Sidereal Time

Solar time is based on the apparent revolution of the sun about the Earth. A solar day is the elapsed time between two successive transits of the sun through the local meridian (To simplify the following discussions, assume that the local meridian is Greenwich. Then, all calculations shown can be transformed to any local meridian by adding or subtracting its longitude).

Sidereal time is based on the revolution of the Earth with respect to distant, fixed stars. A sidereal day is the elapsed time between successive transits of a distant star through the meridian. The stars are many orders of magnitude further away than the sun, hence don't appear to move much over a year (although they do move). More precisely, sidereal time is measured from successive transits of the vernal equinox through the meridian; in other words, sidereal time is the hour angle of the vernal equinox.

Solar and sidereal time scales differ; between successive transits of Υ through the meridian, the Earth has moved along a substantial portion of its orbit. As such, it must rotate about 361 degrees before the sun transits over the meridian. Hence, the solar day is about 4 minutes (of UT) longer than the sidereal day. As the Earth rotates roughly an extra degree during a solar day, it will have rotated about an extra full revolution relative to sidereal during the course of a solar year (i.e. the sidereal year has one more day than the solar one). A combination of the above factors means that the solar and sidereal seconds are not of the same length.

2.2 Solar Time, Universal Time (UT) and Julian Date (JD)

As mentioned above, apparent solar time is the time interval between successive transits of the sun through the meridian. But this is not a smooth timescale. The Earth moves with variable speed along its elliptical orbit about the sun, according to Kepler's second law. Thus the sun's path through the sky exhibits irregular motion along the ecliptic over the year (as seen from Earth). In addition, the inclination of the ecliptic relative to the celestial equator causes the projection of the sun's motion (along the ecliptic) onto the celestial equator (on which its right ascension is measured) to exhibit sinusoidal motion (the sun's declination goes from -23.5 degrees at the winter solstice to +23.5 degrees at the summer solstice). The variation of the sun's apparent motion in right ascension makes it a poor choice for defining a precise time scale. Consequently, mean solar time was adopted in the late 19th century to replace apparent solar time.

To obtain a more uniform time scale, a Fictitious Mean Sun (FMS) was proposed[4, 6]. The Mean Sun takes the same time as the real sun to move from one vernal equinox to the next, but moves at a constant velocity along the celestial equator (i.e. it has a constant right ascension rate). The right ascension of this mean sun is given as a function of sidereal time. Mean solar time is defined as the hour angle of the mean sun plus 12 hours (to ensure that the sun transits the meridian at noon and not 0h). The difference between the mean and apparent solar times is calculable through the so called 'equation of time'[3, 4, 6]. The difference in apparent and mean solar times varies over the year from -14 to +16 minutes.

2.2.1 Universal Time (UT)

Universal Time (UT) was introduced in 1926 to replace the old Greenwich Mean Time (GMT) (at the time, several versions were in use, creating confusion). UT is equivalent to the Mean Solar Time at Greenwich (for most practical purposes). UT is considered a solar time in that it correctly predicts the position of the sun. However, it is not a true solar time in that the sun's observed position is not used to define it, since the achievable accuracy for such a measurement is insufficient. Instead UT is derived from sidereal time (by observations of distant stars) through a mathematical expression which accounts for the known path of the Earth along its orbit and permits the position of the fictitious mean sun to be calculated. Consequently, UT and sidereal time are not truly independent; they are two forms of the same scale, albeit with units of different lengths.

There are three different realizations of UT, and they must be distinguished for precise applications. UT0 is found as above by reducing the observations of stars from many different ground stations. The slight motion of the Earth's poles of rotation (polar motion) cause the geographic position of any place on Earth to vary by several metres, and different observatories will find a different value for UT0 at the same moment. UT1 is computed by correcting UT0 for the effect of this polar motion on the longitude of the observing site. UT1 is the same everywhere on Earth, but the length of a second is still irregular, as the rotational speed of the earth is not uniform. UT1 has an uncertainty of plus or minus 3 milliseconds per day. UT1 is the time scale most commonly adopted in astrodynamic applications. (The third realization, UT2 is now considered obsolete).

2.2.2 Julian Date (JD)

The Julian Day is just a count of the number of solar (UT) days elapsed since 12 noon Greenwich Mean Time (UT) on Monday, January 1, 4713 BC in the Julian calendar (that day is counted as Julian day zero). The Julian day system was intended to provide astronomers with a single system of dates that could be used when working with different calendars and to unify different historic chronologies. It is also very useful in computer applications, as all the time information (month, day, year, time of day) is contained in one variable. A Julian Day always begins at noon (for historical reasons - astronomers in the past were assured that all their observations on a particular night would have the same date - it's not that important today). The Julian Date is the Julian Day number (at 12:00 UT) plus the time of day (UT) since noon, as a fraction of the total day. There are many tables and algorithms to count or calculate the Julian date given a calendar date [4, 6, 5]. Of usual interest is the Modified Julian Date (MJD), which is just the Julian Date in question minus the Julian Date of some reference epoch (there are different choices of epoch available). Expressions such as that for the fictitious mean sun are usually functions of elapsed Julian centuries of 36525 Julian Days (i.e. T = MJD/36525 = (JD(now)-JD(ref. epoch))/36525).

2.3 Greenwich Mean Sidereal Time

As mentioned above, sidereal time is based on successive transits of the Vernal Equinox through the reference meridian. More specifically, it is defined as the hour angle of the vernal equinox. If Greenwich is taken as the reference meridian, we obtain Greenwich Mean Sidereal Time (GMST). Local Sidereal Time (LST) is obtained by adding the geographic longitude of the reference meridian to GMST. This quantity is needed to transform coordinates on a rotating Earth (regular geographic coordinates measuring longitude from Greenwich) to an inertial Earth centred system (where the x-axis points to the Vernal Equinox, and not Greenwich). Thus GMST is a measure of the offset between the Greenwich meridian and the Vernal Equinox. GMST can be calculated according to (2) .

2.4 Atomic Time, Universal Time Coordinated (UTC) and GPS Time

Atomic time (TAI, after the French Temps Atomique International) is the most precise time scale available and contains none of the irregularities of the Earth's motion. The definition of the SI second is based on atomic time. It is defined as a fixed number of wavelengths of the photon produced by a certain electron transition in the ground state of the Cesium-133 atom. This particular value was chosen to best match the most accurate time scales already in existence. Different real world atomic clocks will not fully agree (there are also other standards besides Cesium). TAI is actually defined by taking data from different locations and standards, and statistically weighting them. Known corrections for relativistic effects (atomic clocks are sensitive enough to feel relativistic effects - those at different altitudes or in motion will slow or speed up) are also included to obtain a mean TAI.

In the discussion of UT above, it was noted that UT1 correctly predicts the position of the sun. However, the length of a second derived from UT1 varies noticeably due to irregularities in Earth's rotation. Therefore, a new time scale was devised with the SI second as the base unit. This scale is called Universal Time Coordinated (UTC, sometimes called Zulu time (ZT)). It is a requirement that the difference abs(UTC-UT1) $<$ 0.9 seconds. UTC therefore offers both a constant unit of time and agreement with the position of the sun. For this reason, UTC is the basis of civil time keeping broadcasts worldwide (UTC is sometimes incorrectly called Greenwich Mean Time, but GMT was discontinued as an official time scale due to ambiguities in its usage).

Both TAI and UTC are based on the SI second; hence they flow at the same rate. Both scales coincided when UTC was officially introduced in January, 1972. However, since the UT1 second and the SI second differ in their lengths, UTC drifts with respect to UT1. To maintain the two within 0.9 seconds, leap seconds are added to UTC (at the end of January or June). This ensures that solar noon (averaged over the year) occurs at the same UTC. Due to the addition of leap seconds, UTC is presently 32 seconds behind TAI. The need for leap seconds is decided by the Bureau International de l'Heure (BIH) in Paris, in consultation with various other time laboratories.

GPS time is also based on TAI, and was synchronous with UTC at its introduction in 1980. It does not support leap seconds, and is therefore now 13 seconds ahead of UTC. Note that GPS time is reported in week number (since its introduction in 1980), and number of seconds into that week. UNIX time is also based on UTC. It counts SI seconds from the epoch January 1, 1970.

2.5 Dynamical Time

Dynamical time is mentioned here for completeness. As mentioned above, dynamical time is the independent variable in the equations of motion of the observed celestial object. Prior to the introduction of atomic clocks, Ephemeris Time (ET) was the was the closest available approximation to a uniform time scale for astrodynamic calculations. It was replaced by Terrestrial Time (TT) in 1984, where TT = TAI + 32.184 s. The offset from TAI was necessary to maintain continuity between TT and ET at the transition. Planetary motions are computed with Barycentric Dynamical Time (French acronym TDB), which is more uniform than TT as it accounts for relativistic corrections due to the Earth's motion through the Sun's gravitational field. Dynamical Times are not generally used for satellites in Earth orbit, and they are not used in this analysis.

3 Coordinate Systems and Transformations

This section describes the spatial coordinate systems used in this analysis and the various transformations necessary to convert from one to the other. We begin with the definitions of the two-coordinate systems of interest. These are both Earth centred-Cartesian coordinate systems. The first is the Earth Centered Inertial (ECI), or Geographic Equatorial Inertial (GEI) coordinate system (see figure 1). It is called an inertial coordinate system because the x-axis points, independently of the Earth's rotation, towards the fixed stars. The fixed stars chosen for the x-axis are actually only "quasi"-fixed since, technically, the axis points in the direction of the sun at the vernal equinox, and is thus subject to the Earth's precession. The vernal equinox, the first day of Spring in the northern hemisphere, is often referred to as the first point of Aires. Astronomically, the vector joining the earth and the sun on this day gives the direction in space of the vernal equinox. This direction is denoted using the symbol of the ram, Υ.

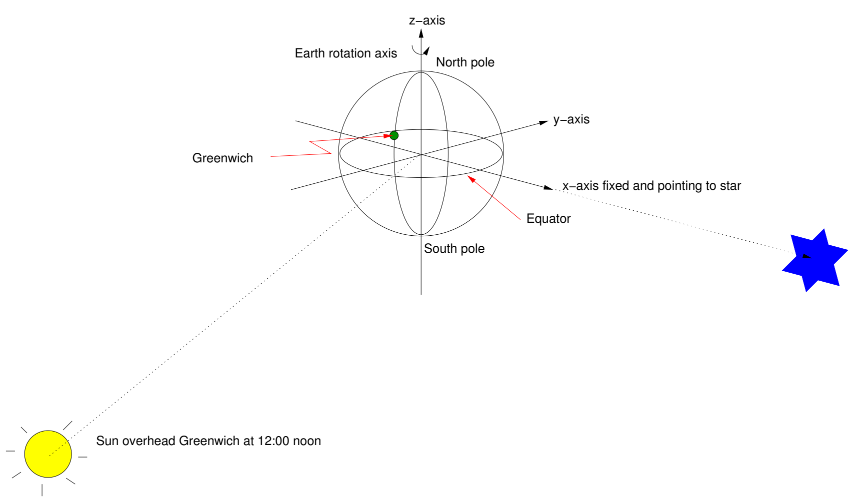

Figure 1: *Inertial Coordinate System. Fixed x-axis is in the direction of Υ, the direction of the vector between the earth and sun on the vernal equinox.*

The second coordinate system, see figure 2, chooses an x-axis as the vector between the centre of the Earth and the point on the equator due south of Greenwich. This coordinate system clearly rotates when viewed from off-planet, and is therefore denoted as an Earth Centered Rotating (ECR) or Earth-Centered Earth-Fixed (ECEF) system.

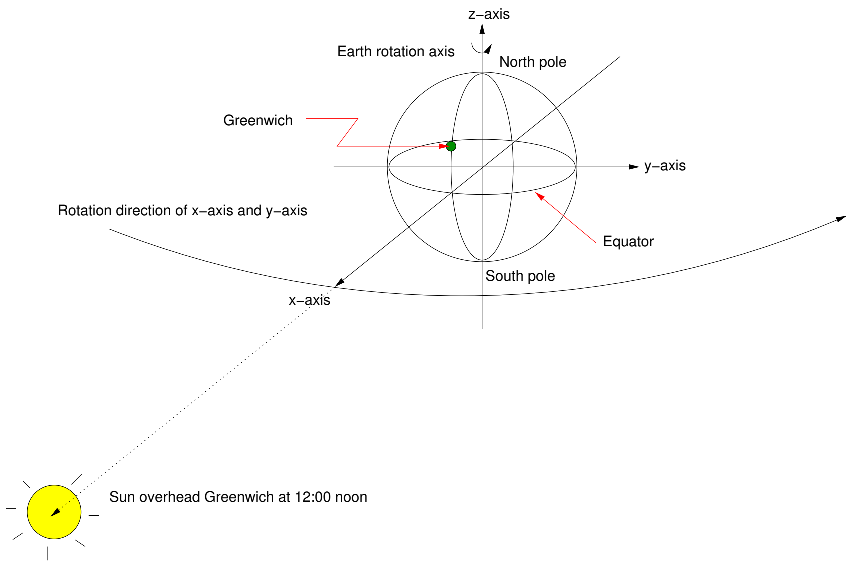

Figure 2: *Rotating Coordinate System. The x-axis is in the direction of the vector joining the centre of the earth and the point on the equator due south of Greenwich.*

3.1 Transforming Between Inertial and Rotating Coordinate Systems

Assume some point in time at which we wish to compute the coordinates of a point in the rotating coordinate system from the coordinates of the same point in the inertial coordinate system. The motion of the rotating coordinate system dictates that the transformation depends critically on a well defined and accurate time of interest. Therefore, assume that the position of a coordinate in the inertial system is given by the vector

$$\vec{p}_i(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \\ z_i(t) \end{bmatrix} \tag{1}$$

We must first compute the orientation of the rotating coordinate system relative to the inertial coordinate system at this point in time. Assuming that the time is given in units of Julian centuries, we can compute the angle between Greenwich and Υ, or the Greenwich angle in degrees as[3, 4]

$$\Upsilon_G(t) = \frac{360°/\text{day}}{86400 \text{ s/day}} \sum_{n=0}^{3} a_n t^n, \tag{2}$$

where

$$a_0 = 67310.54841 \tag{3}$$
$$a_1 = 876600 \cdot 3600 + 8640184.812866 \tag{4}$$
$$a_2 = 0.093104 \tag{5}$$
$$a_3 = -6.2 \cdot 10^{-6}. \tag{6}$$

Although in the a_n we have neglected to present the units, they are such that the summation in (2) has units of seconds. The actual units of a_1 are s/Julian century since the time variable is in Julian centuries. To compute the Julian century from a UT1 time, we use the formula[3, 4, 6]

$$t = t_{\mathrm{UT1}}/36525, \tag{7}$$

where t_{UT1} is given as decimal days since 00:00.0000 01-01-2000, the instant that began the year 2000.

To complete the transformation, we use the transformation matrix

$$\boldsymbol{A}(t) = \begin{bmatrix} \cos\Upsilon_G(t) & \sin\Upsilon_G(t) & 0 \\ -\sin\Upsilon_G(t) & \cos\Upsilon_G(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{8}$$

and compute the rotating coordinate system coordinates as

$$\vec{p}_r(t) = \boldsymbol{A}(t)\vec{p}_i(t) \tag{9}$$

To transform velocity vectors between the coordinate systems, we take the derivative of (9) with respect to time to get

$$\vec{v}_r(t) = \frac{\mathrm{d}\boldsymbol{A}(t)}{\mathrm{d}t}\vec{p}_i(t) + \boldsymbol{A}(t)\vec{v}_i(t), \tag{10}$$

where

$$\frac{\mathrm{d}\boldsymbol{A}(t)}{\mathrm{d}t} = \begin{bmatrix} -\Upsilon'_G(t)\sin\Upsilon_G(t) & \Upsilon'_G(t)\cos\Upsilon_G(t) & 0 \\ -\Upsilon'_G(t)\cos\Upsilon_G(t) & -\Upsilon'_G(t)\sin\Upsilon_G(t) & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{11}$$

and

$$\Upsilon'_G(t) = \frac{360°/\text{day}}{86400\ \text{s/day}} \frac{1}{36525\ \text{days/Julian century} \cdot 86400\ \text{s/day}} \sum_{n=1}^{3} n a_n t^n \tag{12}$$

in degrees per second.

3.2 Transforming Between Rotating Cartesian and Geodetic Coordinates on the Ground

The transformation from Earth Centered Rotating (ECR) coordinates to familiar latitude, longitude and height is a fairly simple one. Given the position vector $\vec{p}_r(t) = (X, Y, Z)^T$, on a spherical Earth the latitude and longitude, ϕ_{gc} and λ, are calculated directly as:

$$\begin{aligned} \phi_{gc} &= \tan^{-1} \frac{Z}{\sqrt{X^2+Y^2}}, \quad \text{and} \\ \lambda &= \tan^{-1} \frac{Y}{X}. \end{aligned} \tag{13}$$

The latitude subscript gc refers to geocentric latitude. Geodetic latitude (the every-day latitude we are used to) is defined as the angle between the equatorial plane and the normal to the Earth's surface at the point in question. Hence, it will differ from the geocentric latitude due to the polar flattening of the Earth, as shown in figure 3.

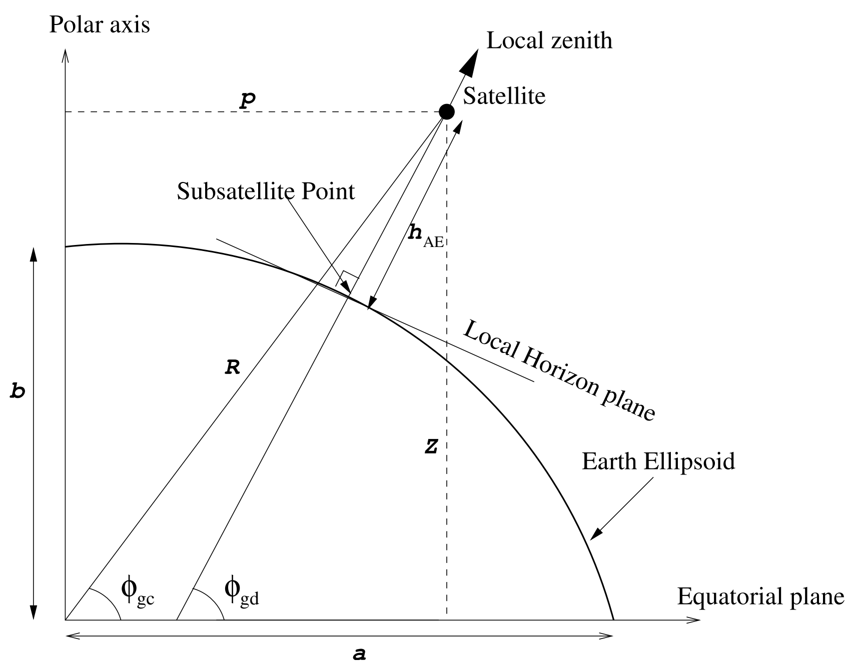

Figure 3: *Geocentric and geodetic latitude on the Earth ellipsoid. The earth's flattening is exagerated to illustrate the difference. The actual difference between the two varies with latitude and is at most 0.2 degrees*

Fortunately, given an Earth ellipsoid model, there are numerous algorithms to transform geocentric to geodetic latitude, ϕ_{gd}, and calculate the height above the ellipsoid, h_{AE} (with varying degrees of accuracy; the more accurate methods generally involve iteration). A single step transformation, which is not exact, but provides centimeter level accuracy for objects up to low earth orbit (LEO - $h < 1000km$) is sufficient for our purposes.[7]:

$$\begin{aligned}
\phi_{gd} &= \tan^{-1} \frac{Z + \varepsilon^2 b \sin^3 \theta}{p - e^2 a \cos^3 \theta}, \\
\lambda &= \tan^{-1} \frac{Y}{X}, \\
h_{ae} &= \frac{p}{\cos \phi_{gd}} - N(\phi).
\end{aligned} \tag{14}$$

where a and b are the semi-major and semi-minor axes (equatorial and polar Earth radii) respectively. The remaining quantities are as follows[3, 8]:

$$\begin{aligned}
f &= \frac{a-b}{a} \longrightarrow \text{Earth flattening factor}, \\
e^2 &= 2f - f^2 \longrightarrow \text{Earth ellipsoid eccentricity squared}, \\
N(\phi) &= \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \longrightarrow \text{Radius of curvature of the prime vertical}, \\
\varepsilon^2 &= \frac{a^2 - b^2}{b^2}, \\
p &= \sqrt{X^2 + Y^2}, \\
\theta &= \tan^{-1} \frac{Za}{pb}.
\end{aligned}$$

Note that h_{AE}, the height of the object above the ellipsoid, is not the same as the height above mean sea level used in everyday applications (eg. Digital Elevation Models); the two can differ by tens of meters. Heights above mean sea level are called orthonometric heights h_{ORTH}, and they represent a height above the geoid. The geoid represents the gravitational potential of mean sea level, and orthonometric heights are measured relative to it. As such, the geoid may be above or below the ellipsoid (depending on the Earth's mass distribution at that point). To convert an ellipsoid height to an orthonometric one, a geoid model is needed. The geoid height N_{geoid} (sometimes called geoid undulation), is given by $N = h_{AE} - h_{ORTH}$. So given a geoid model, the true height above mean sea level can be calculated

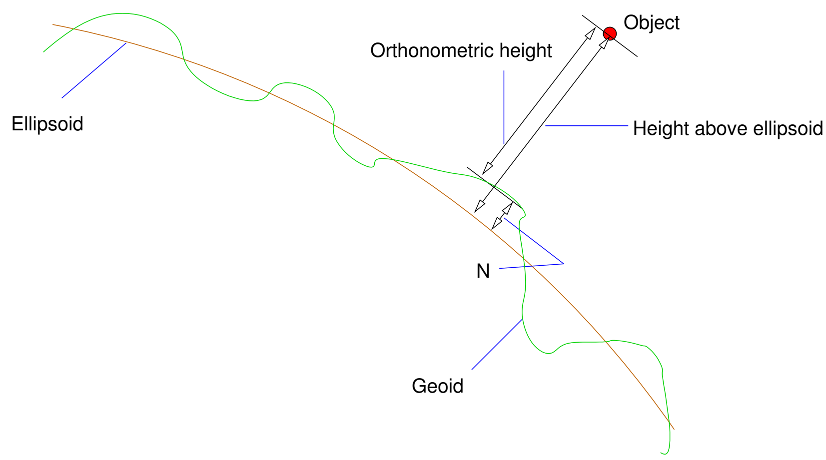

Figure 4: *Ellipsoid and Geoid.*

by subtracting the geoid undulation from the ellipsoid height (see figure 4). The reverse transformation back to rotating Cartesian coordinates is given by[3, 8, 5]:

$$\begin{aligned} X &= (N(\phi) + h_{AE})\cos\phi\cos\lambda, \\ Y &= (N(\phi) + h_{AE})\cos\phi\sin\lambda, \\ Z &= (N(\phi)(1-e^2) + h_{AE})\sin\phi. \end{aligned} \tag{15}$$

4 Interpolation

Often the radar satellite state vectors are provided in the inertial coordinate system. The following interpolation routine works best when coordinates are presented in the rotating coordinate system. Additionally, SAR processing routines are readily implemented using the rotating coordinate system since they attempt to create an image of the Earth's rotating surface. For the remainder of this document, therefore, it is assumed that the coordinates have been transformed from the inertial coordinate system to the rotating coordinate system using the procedure of section 3.

SAR processing depends critically on an accurate model of the imaging geometry as it evolves through time[9, 10, 11, 12]. The geometry from a space based radar presents an interesting problem because both the satellite and the earth's surface move with time. To complicate matters further, ERS-1 and RADARSAT-1 are not capable of measuring their positions and velocities (state vectors) at the pulse repetition frequency.

The positions and velocities of RADARSAT-1 and ERS-2 have to either be computed from an orbit propagator (predicted state vectors), or derived from measurements (measured state vectors). There are two sources of measured state vectors, those measured by sensors on board the satellite and those interpolated or refined from position measurements made by facilities on the ground. This analysis uses state vectors measured by equipment onboard the satellite and refined by post-processing on the ground. This provides the most accurate state vectors possible, although the procedure is applicable to state vectors from any source (i.e. the procedure will yield the same precision, but the accuracy will be limited to the accuracy of the input state vectors). The state vector of RADARSAT-1 is measured every 480 seconds, while that of ERS-2 is measured every 30 seconds. SAR image creation requires up-sampling (interpolation) of these sparse data points to the pulse repetition frequency.

This section proposes a simple interpolation scheme that makes use of Hermite polynomials. Note that the approach is precise, but only as accurate as the input state vectors used. Let the state vectors corresponding to n sample points be written as position row-vectors, $\vec{x}_r(t), i \in \{1, \ldots, n\}$, and velocity row-vectors, $\dot{\vec{x}}_r(t)$. These vectors have components in the three dimensional Cartesian coordinate system. By a Hermite polynomial interpolation, we seek to find a $(2n-1)^{\text{th}}$ degree polynomial that passes through the points $\vec{x}_r(t)$, possessing derivatives at these points given by $\dot{\vec{x}}_r(t)$. Let this polynomial be given by

$$p(t) = \sum_{k=0}^{2n-1} \vec{a}_k t^k. \tag{16}$$

Otherwise stated, the problem is to find the vector coefficients $\vec{a}_k$ that satisfy the criterion of passing through the sample points with the given time derivatives. Oftentimes, the derivatives at the sampling points are not known. Luckily, these quantities are measured with ERS-2 and RADARSAT-1. Substituting the time of each sample into the polynomial and equating the position and velocity vectors gives the matrix equation

$$\begin{bmatrix} 0^{2n-1} & \dots & 0^2 & 0^1 & 1 \\ (2n-1)\cdot 0^{2n-2} & \dots & 2\cdot 0^1 & 1\cdot 1 & 0 \\ 1^{2n-1} & \dots & 1^2 & 1^1 & 1^0 \\ (2n-1)\cdot 1^{2n-2} & \dots & 2\cdot 1^1 & 1\cdot 1^0 & 0 \\ 2^{2n-1} & \dots & 2^2 & 2^1 & 2^0 \\ (2n-1)\cdot 2^{2n-2} & \dots & 2\cdot 2^1 & 1\cdot 2^0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (n-1)^{2n-1} & \dots & (n-1)^2 & (n-1)^1 & (n-1)^0 \\ (2n-1)\cdot(n-1)^{2n-2} & \dots & 2\cdot(n-1)^1 & 1\cdot(n-1)^0 & 0 \end{bmatrix} \begin{bmatrix} \vec{a}_{2n-1} \\ \vdots \\ \vec{a}_3 \\ \vec{a}_2 \\ \vec{a}_1 \\ \vec{a}_0 \end{bmatrix} = \begin{bmatrix} \vec{x}_1 \\ \dot{\vec{x}}_1 \\ \vdots \\ \vec{x}_{n-1} \\ \dot{\vec{x}}_{n-1} \\ \vec{x}_n \\ \dot{\vec{x}}_n \end{bmatrix} \tag{17}$$

which can be written: $\boldsymbol{QA} = \boldsymbol{S}$. Now, so long as $\boldsymbol{A}$ is invertible, the equation can be solved to yield the polynomial coefficients. For example, if there are 4 anchor points (four samples), then

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 448 & 192 & 80 & 32 & 12 & 4 & 1 & 0 \\ 2187 & 729 & 243 & 81 & 27 & 9 & 3 & 1 \\ 5103 & 1458 & 405 & 108 & 27 & 6 & 1 & 0 \end{bmatrix} \tag{18}$$

, and

$$\boldsymbol{A}^{-1} = \frac{1}{99900} \begin{bmatrix} 10175 & 2775 & 24975 & 24975 & -24975 & 24975 & -10175 & 2775 \\ -119325 & -33300 & -249750 & -274725 & 274725 & -249750 & 94350 & -24975 \\ 556850 & 160950 & 924075 & 1173825 & -1148850 & 949050 & -332075 & 86025 \\ -1304250 & -399600 & -1498500 & -2422575 & 2247750 & -1698300 & 555000 & -141525 \\ 1564175 & 535575 & 899100 & 2397600 & -2022975 & 1423575 & -440300 & 111000 \\ -807525 & -366300 & 0 & -899100 & 674325 & -449550 & 133200 & -33300 \\ 0 & 99900 & 0 & 0 & 0 & 0 & 0 & 0 \\ 99900 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{19}$$

5 ERS Experiment

To test the Hermite polynomial interpolation scheme, use was made of the higher sampling frequency of ERS-2. By keeping only state vectors sampled every 480 seconds, the RADARSAT-1 sampling frequency was approximated. Then, by interpolating these low frequency state vectors using the Hermite polynomial approach, and comparing with the actual measurements (the deleted state vectors) the interpolation precision was assessed. Theoretically, interpolation requires only two anchor points (samples of the data at the low sampling frequency). However, precision of the interpolation improves with more anchor points, up to a certain limit. The error as a function of the number of anchor points is evident from figures 5 to 11. The figures show that the interpolation precision is compromised for seven or more anchor points, especially near the edges. In fact, in figure 11 the final few sample points are no longer anchored.

figure 7, which exhibits the smallest deviation between interpolated and sampled points, suggests that four anchor points are a good choice for interpolation. It shows that precisions on the order of 10 cm (or slightly better) are achievable; the maximum deviation at the edges is only about 15cm, and the RMS spread is less than 10cm. This explains the explicit calculation and presentation of $\boldsymbol{A}^{-1}$ in (19). Since the interpolation between anchor points one and two and anchor points three and four is less precise than between anchor points two and three, one should, for optimal precision, choose a pair of anchor points on each side of the desired interpolation instant. This is impossible at the beginning and the end of the orbit where reduced precision just has to be accepted. Figure 12, presents the overall interpolation error in position while figure 13 presents the overall interpolated error in velocity.

All computations were carried out in MATLAB© with the following two m-files (listed in the appendix): interState.tex and qmatrix.tex.

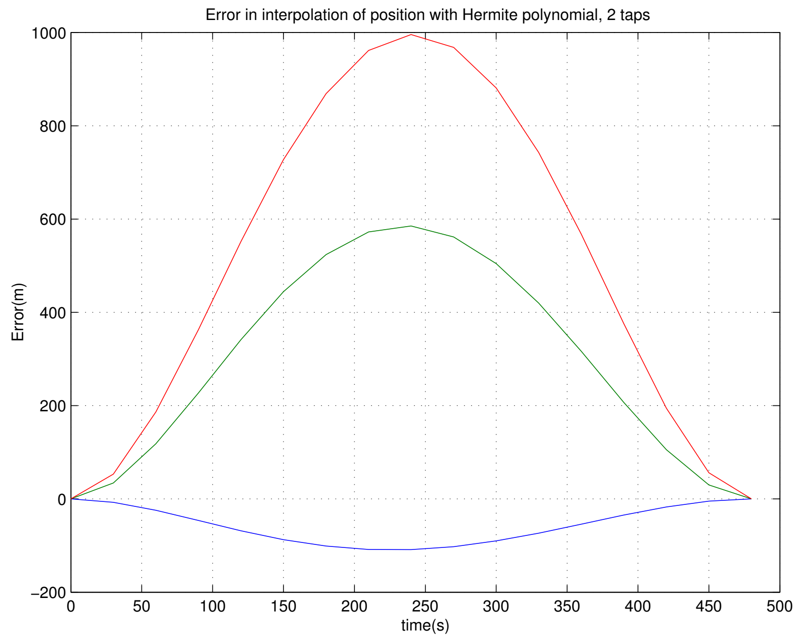

Figure 5: *Error between interpolated points and sampled points for a 2-point anchor.*

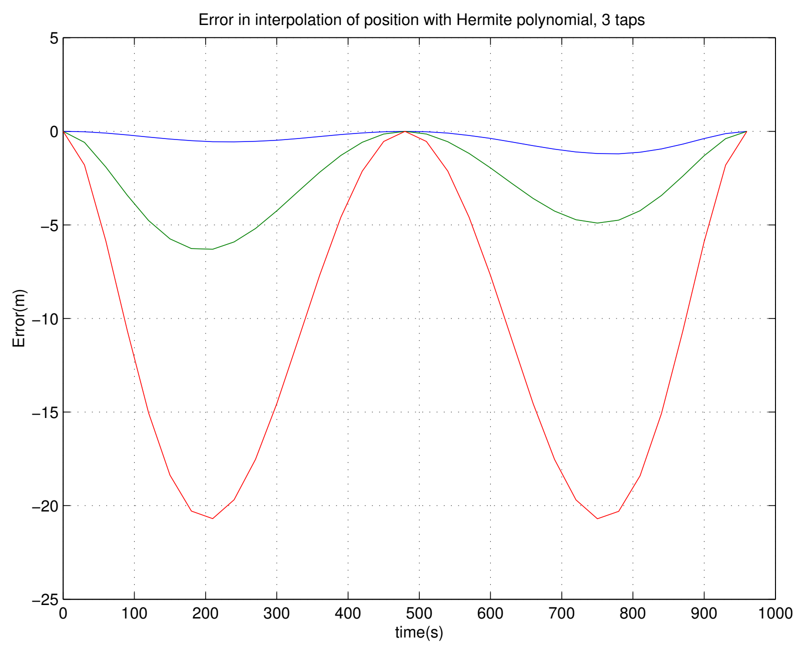

Figure 6: *Error between interpolated points and sampled points for a 3-point anchor.*

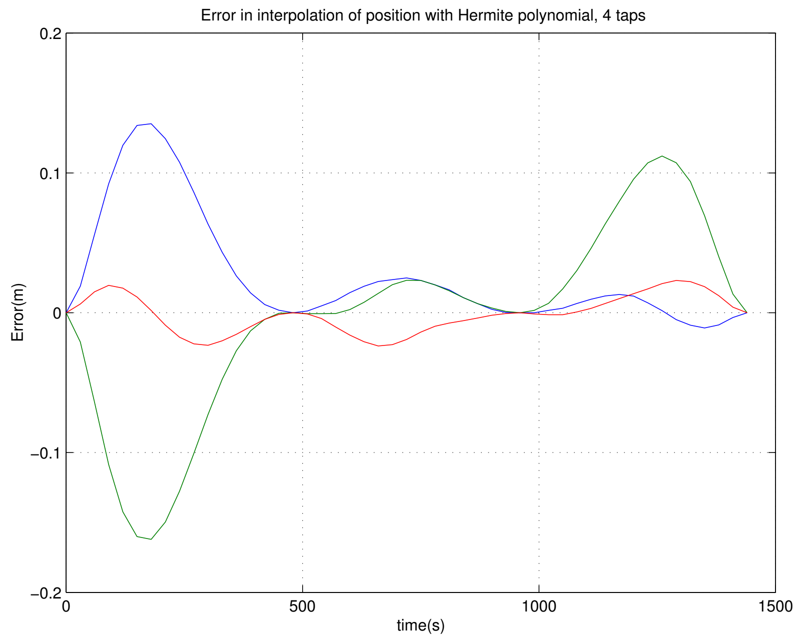

Figure 7: *Error between interpolated points and sampled points for a 4-point anchor.*

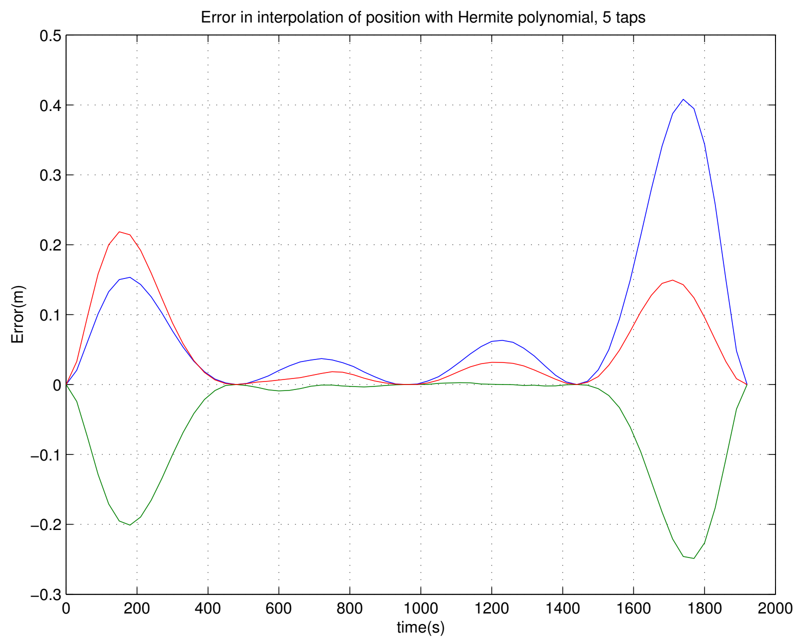

Figure 8: *Error between interpolated points and sampled points for a 5-point anchor.*

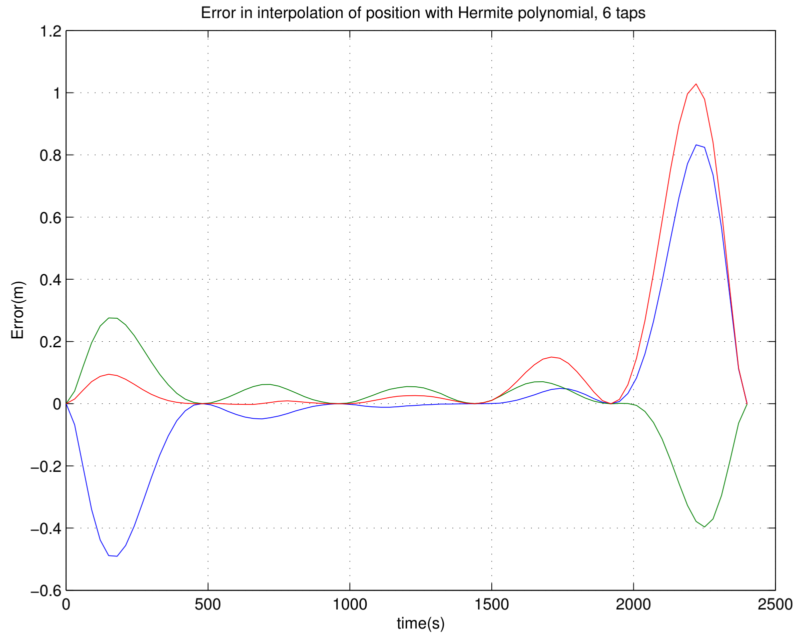

Figure 9: *Error between interpolated points and sampled points for a 6-point anchor.*

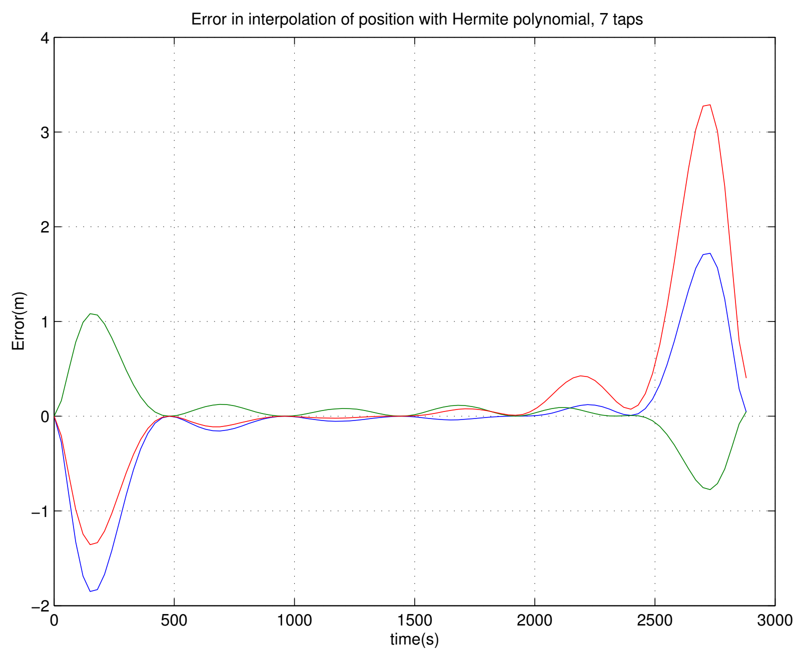

Figure 10: *Error between interpolated points and sampled points for a 7-point anchor.*

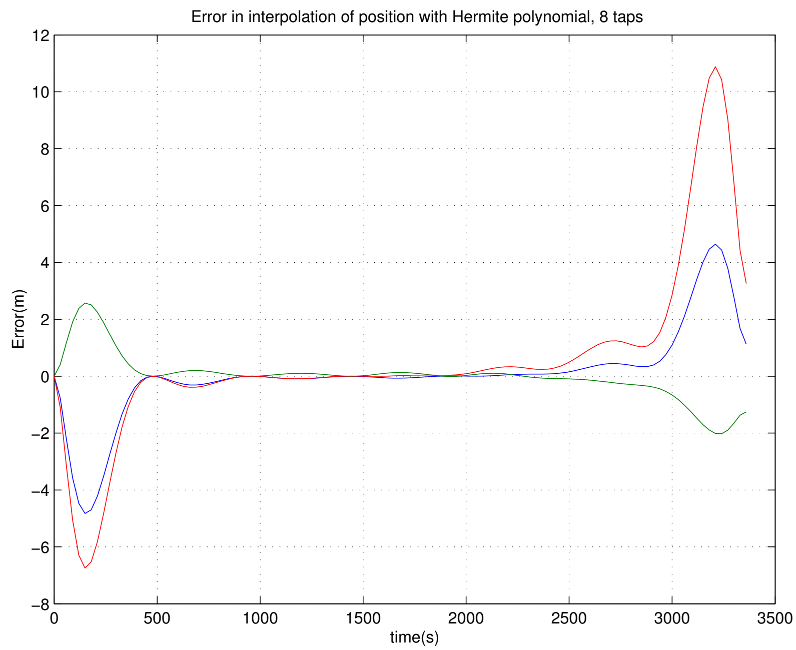

__Figure 11:__ Error between interpolated points and sampled points for a 8-point anchor.

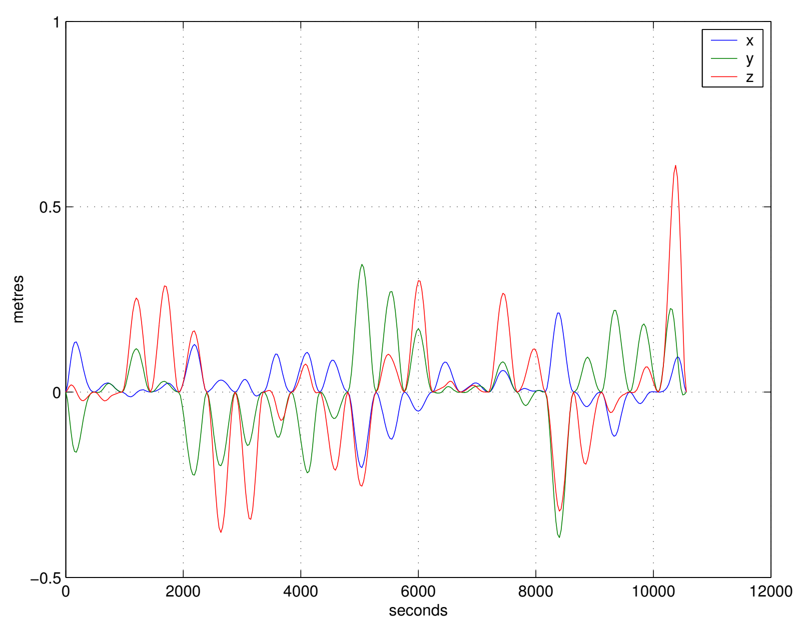

__Figure 12:__ Error between interpolated points and sampled points for whole orbit with moving 4-point anchor.

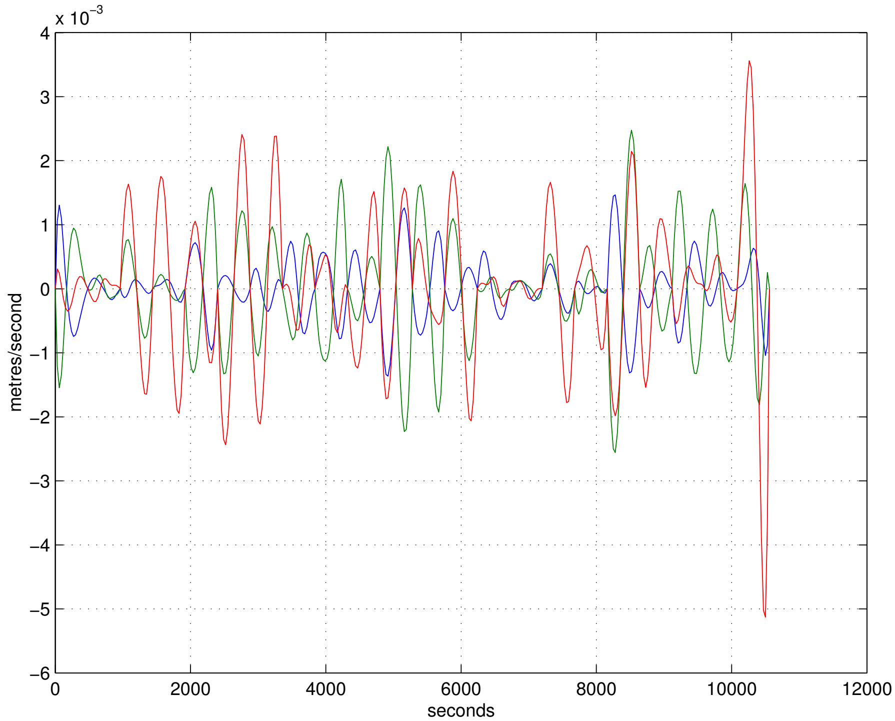

Figure 13: *Error between interpolated velocity and sampled velocity for whole orbit with moving 4-point anchor.*

6 Example of transformation and interpolation of RADARSAT-1 state vectors

Radarsat 1 provided the illumination source for a bi-static experimental trial conducted at CFB Petawawa in Spring 2004. During the data processing stage of the trial, there arose a requirement to estimate the satellite position with the greatest possible accuracy at any given moment. This section demonstrates estimation of the satellite position using the Hermite polynomial approach.

The procedure for estimating the satellite position at time t_0 proceeds as follows

1. Obtain a set of satellite state vectors that span the time of interest t_0
2. Transform these state vectors from the GEI coordinate system to the ECEF coordinate system
3. Use the Hermite polynomial approach to estimate the state vector at the desired time
4. Transform the ECEF state vector into geographic coordinates

The satellite state vectors for this experiment were acquired from the Canadian Space Agency. The requested data given the date of interest appears below

```
;###FILENAME:                    D4419600.ORB
;###SPACECRAFT_IDENTIFIER:       RADARSAT_1
;###FILE_CREATION_TIME:          2004-115-01:39:24.000
;###FILE_SOURCE:                 FD
;###FILE_DEST:              SDB
;###FILE_TYPE:                   FD_ORBIT_ANCILLIARY_DATA
;*********************************************************************
;
;    DEFINITIVE ORBIT
;    SEQUENCE                    08                              10
;
GENERATION_TIME = 2004−115−00:00:00.000          ;UTC
ORBIT_NUMBER = 44196
GREENWICH_ANGLE = 3.524057211156                 ;Radians
;
;
;
;
;
```

```
1 NOT AVAILABLE FOR DEFINITIVE ORBIT DATA
2 NOT AVAILABLE FOR DEFINITIVE ORBIT DATA
;
;
2004−113−23:22:16.342                          ;Equator Crossing Time (UTC)
 −3702003.54    6143766.40       1828.96      ;Position X,Y,Z (m)
   957065.74     564415.83    7372935.31      ;Velocity X,Y,Z (mm/s)
;
2004−113−23:30:16.342                          ;Time Tag (UTC)
 −2809585.85    5653642.57    3395020.06      ;Position X,Y,Z (m)
  2683920.63   −2564244.00    6469618.85      ;Velocity X,Y,Z (mm/s)
;
2004−113−23:38:16.342                          ;Time Tag (UTC)
 −1231253.44    3783309.57    5957367.73      ;Position X,Y,Z (m)
  3755139.53   −5065834.68    3984138.32      ;Velocity X,Y,Z (mm/s)
;
2004−113−23:46:16.342                          ;Time Tag (UTC)
   647464.65     990146.14    7063206.53      ;Position X,Y,Z (m)
  3909791.92   −6329777.17     527823.22      ;Velocity X,Y,Z (mm/s)
;
2004−113−23:54:16.342                          ;Time Tag (UTC)
  2368408.35   −2044349.12    6443381.50      ;Position X,Y,Z (m)
  3111423.87   −6050636.97   −3056830.79      ;Velocity X,Y,Z (mm/s)
;
2004−114−00:02:16.342                          ;Time Tag (UTC)
  3511833.08   −4580261.32    4248659.87      ;Position X,Y,Z (m)
  1553209.63   −4295052.56   −5897360.37      ;Velocity X,Y,Z (mm/s)
;
2004−114−00:10:16.342                          ;Time Tag (UTC)
  3798073.38   −5998138.70    1014415.15      ;Position X,Y,Z (m)
  −385449.63   −1489471.60   −7296849.33      ;Velocity X,Y,Z (mm/s)
;
2004−114−00:18:16.342                          ;Time Tag (UTC)
  3157291.88   −5952001.08   −2468203.36      ;Position X,Y,Z (m)
 −2228173.13    1676526.19   −6911058.90      ;Velocity X,Y,Z (mm/s)
;
2004−114−00:26:16.342                          ;Time Tag (UTC)
  1747348.31   −4456026.81   −5348568.96      ;Position X,Y,Z (m)
 −3523753.99    4425524.70   −4841748.92      ;Velocity X,Y,Z (mm/s)
;
2004−114−00:34:16.342                          ;Time Tag (UTC)
   −87017.44   −1878052.46   −6927623.80      ;Position X,Y,Z (m)
 −3960973.09    6092854.95   −1602532.10      ;Velocity X,Y,Z (mm/s)
;
2004−114−00:42:16.342                          ;Time Tag (UTC)
 −1900425.43    1154834.94   −6824980.62      ;Position X,Y,Z (m)
 −3438637.41    6282869.60    2021460.22      ;Velocity X,Y,Z (mm/s)
;
2004−114−00:50:16.342                          ;Time Tag (UTC)
 −3253467.38    3907810.30   −5065288.18      ;Position X,Y,Z (m)
 −2081584.49    4949607.70    5159893.37      ;Velocity X,Y,Z (mm/s)
```

```
;
2004−114−00:58:16.342                                ;Time Tag (UTC)
 −3816262.25     5711417.87    −2072735.90   ;Position X,Y,Z (m)
  −213517.36     2407685.77     7050396.10   ;Velocity X,Y,Z (mm/s)
;
2004−114−01:06:16.342                                ;Time Tag (UTC)
 −3449151.04     6123200.41     1425735.83   ;Position X,Y,Z (m)
  1711884.80     −728710.89     7221993.81   ;Velocity X,Y,Z (mm/s)
;
2004−114−01:14:16.342                                ;Time Tag (UTC)
 −2240068.63     5040194.37     4575169.95   ;Position X,Y,Z (m)
  3220779.21    −3689499.77     5626300.02   ;Velocity X,Y,Z (mm/s)
;
;
;###END_OF_FILE
```

These data represent 15 state vectors. The sampling interval for the state vectors is $1/480$ Hz. The coordinates are given in the GEI system. Therefore, the first step is to transform the coordinates into the ECEF coordinate system. This transformation requires an accurate UT1 time measurement of the sampling time. As can be seen from the listing above, the times are given in the UTC coordinate system. To convert to UT1 a table of values tabulated on the website of the U.S. Naval Observatory [13] is used. The difference between UTC and UT1 for the date of interest, April 23, 2004 is tabulated as -0.4526439. This value is used to transform the four state vectors spanning the time of interest, approximately 23:00 UTC. The results are listed in table 6. The script in Appendix B has been used to create Table 6.

Next, the Hermite polynomial approach is used to upsample the ECEF coordinates to, for example, 10Hz. The script in Appendices E, F is used to upsample these data, and, along with the original $1/480$ Hz samples they are illustrated in figures 14 to 15.

Finally, the Cartesian ECEF coordinates are converted into geographic coordinates. With the aid of the MATLAB© code in Appendix C, the geographic coordinates are plotted in figure 16. Additionally, the height above the WGS84 ellipsoid is plotted in figure 17. In 17, one sees that the altitude of the spacecraft above the ellipsoid increases as the spacecraft travels north. This is expected as the shorter axis of the ellipsoid passes through the north pole.

Table 1: *GEI to ECEF coordinate transformation*

Time (UTC)	Υ	Vel/Pos	System	$\hat{x}$	$\hat{y}$	$\hat{z}$
22:52:52.469	3.4127	Pos (m)	**GEI**	-3.8052e+06	6.0805e+06	3.7348e+02
			ECEF	2.0378e+06	-6.8774e+06	3.7348e+02
		Vel (m/s)	**GEI**	9.4666e+02	5.8181e+02	7.3729e+03
			ECEF	-1.5694e+03	-4.5565e+02	7.3729e+03
23:00:52.469	3.4477	Pos (m)	**GEI**	-2.9049e+06	5.6061e+06	3.3937e+06
			ECEF	1.0805e+06	-6.2209e+06	3.3937e+06
		Vel (m/s)	**GEI**	2.7261e+03	-2.5174e+03	6.4703e+03
			ECEF	-2.2944e+03	3.1431e+03	6.4703e+03
23:08:52.469	3.4827	Pos (m)	**GEI**	-1.2955e+06	3.7632e+06	5.9566e+06
			ECEF	-3.8001e+04	-3.9797e+06	5.9566e+06
		Vel (m/s)	**GEI**	2.7261e+03	-2.5174e+03	6.4703e+03
			ECEF	-2.2356e+03	6.0001e+03	3.9855e+03
23:16:52.469	3.5177	Pos (m)	**GEI**	6.2990e+05	1.0023e+06	7.0631e+06
			ECEF	-9.5403e+05	-7.0090e+05	7.0631e+06
		Vel (m/s)	**GEI**	4.0159e+03	-6.2628e+03	5.2945e+02
			ECEF	-1.4860e+03	7.3697e+03	5.2945e+02

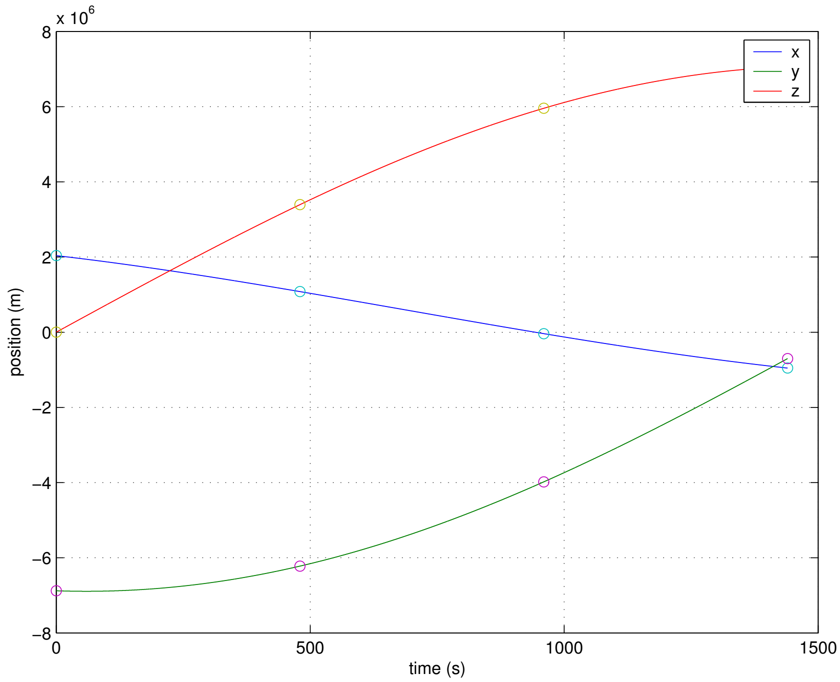

Figure 14: *Upsampled position coordinates (ECEF) using Hermite polynomial approach*

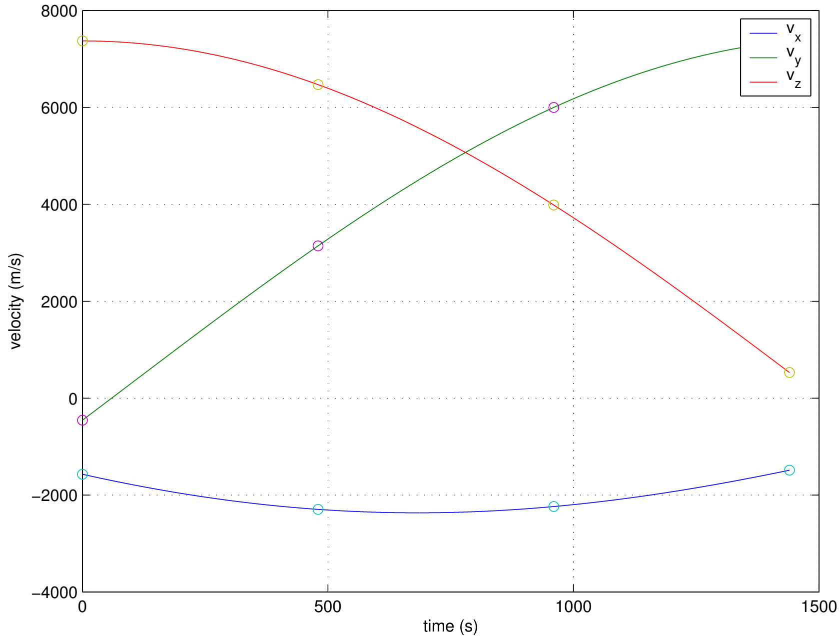

Figure 15: *Upsampled velocity coordinates (ECEF) using Hermite polynomial approach*

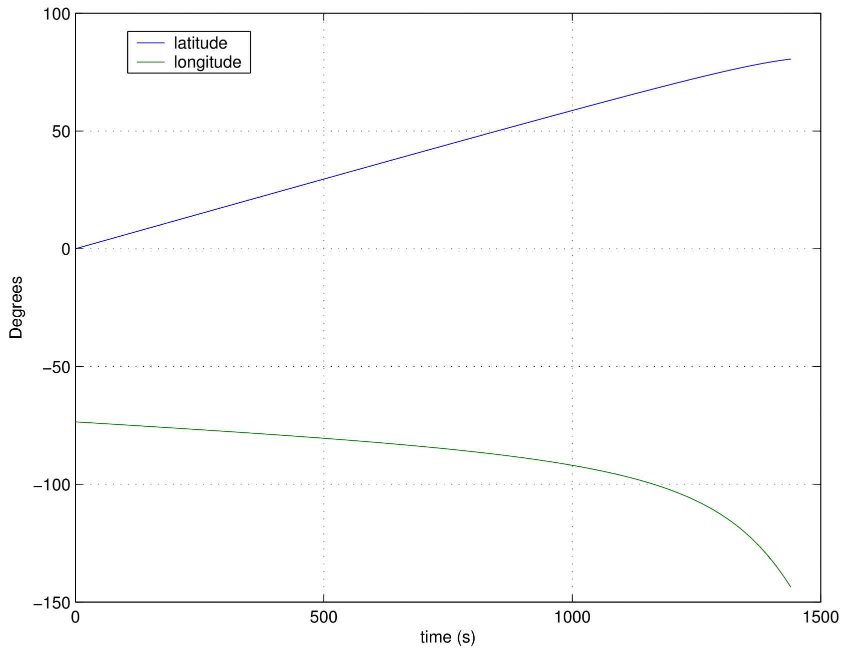

Figure 16: *10Hz latitude and longitude coordinates*

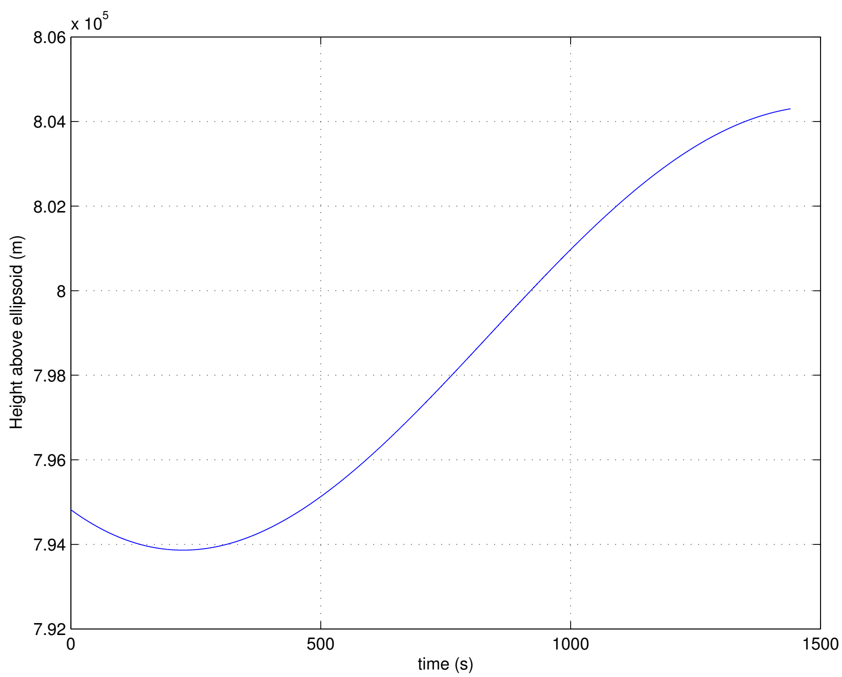

Figure 17: *10Hz height samples (above WGS84 ellipsoid)*

7 Conclusion

To SAR process RADARSAT-1 data, or to institute any other application that requires positions of points on the Earth relative to RADARSAT at relativly fine sampling points, we propose to perform the following steps. First, given the state vectors of RADARSAT-1 in the GEI coordinate system, we transform to the ECEF coordinate system. This transformation requires accurate measurements of sampling times. Secondly, we interpolate the transformed state vectors using a Hermite polynomial interpolation. The tranformation must be made prior to interpolation to ensure the accuracy of the interpolation. It is found that this procedure allows for the interpolation of state vectors to a precision of 10 cm or less. The procedure works with state vectors from any source; predicted from a propagator, measured onboard the spacecraft, or refined/corrected from filtered observations from the ground. The accuracy of the interpolated points, however, is limited to the accuracy of the state vectors used in the interpolation. This approach presents a simple and efficient way to very precisely interpolate satellite position and velocity at arbitrary times without the need to perform any orbit propagation.

8 List of Acronyms

BIH: *Bureau International de l'Heure* or International Bureau for Time
ECEF: Earth Centered Earth Fixed
ECI: Earth Centered Inertial
ECR: Earth Centered Rotating
ERS: European Remote Sensing Satellite
ET: Ephemeris Time
FMS: Fictitious Mean Sun
GEI: Geocentric Equatorial Inertial
GMST: Greenwhich Mean Siderial Time
GMT: Greenwhich Mean Time
GPS: Global Positioning System
JD: Julian Date
LEO: Low Earth Orbit
LST: Local Siderial Time
MJD: Modified Julian Date
SAR: Synthetic Aperture Radar
SI: *Systeme International* of units
TAI: *Temps Atomique International* or Atomic Time
TDB: *Temps Dynamique Barycentrique* or Barycentric Dynamical Time
TT: Terestrial Time
UT: Universal Time
UTC: Universal Time Coordinated
Z: Zulu
ZT: Zulu Time

References

1. Polloc, K. R. (1994). An Analysis Of Orbital Propagators for Low Earth Orbit Rendezvous. Master's thesis. Naval Postgraduate School.

2. Kwak, S, Shin, D, and Kim, T-G (1999). Analysis of Orbit Interpolation and Extrapolation Accuracy for High Precison Topographic Mapping Using Satellite Images. *Proceedings of Asian Conf. on Rem. Sens.*

3. Wertz, J. R. (1986). Spacefraft Attitude Determination and Control, Boston and London: D. Reidel Publishing Company.

4. Vallado, D. A. (2001). Fundamentals of Astrodynamics and Applications, second ed. El Segundo, California: Microcosm Press and Kluwer Academic Publishers.

5. Hernandez-Pajares, M., Juan, M., and Sanz, J. (1996). Time. GPS Data Processing - Lecture Notes. http://maite152.upc.es/ manuel/tdgps/node18.html.

6. Husfeld, D. (1998). Astronomical Time Keeping. The Munich Astro Archive. http://www.maa.mhn.de/Scholar/times.html.

7. Bowring, D. (1976). Transformation from Spatial to Geographic Coordinates. *Survey Review*, **XXIII**, 323–327.

8. Dana, P.H. (1999). Geodetic Datum Overview. The Geographer's Craft Project, Dept. of Geography, University of Colorado at Boulder. http://www.colorado.edu/geography/gcraft/notes/datum/datum.html.

9. Curlander, J. C. and McDonough, R. N. (1991). Synthetic Aperture Radar: Systems and Signal Processing, New York: Wiley Series in Remote Sensing: John Wiley and Sons.

10. Curlander, I. G. and Wong, F. H. (2005). Digital Processing of Synthetic Aperture Radar Data, Boston and London: Artech House.

11. Franceschetti, G. and Lanari, R. (1999). Synthetic Aperture Radar Processing, Washington: CRC Press.

12. Soumekh, M. (1999). Synthetic Aperture Radar Signal Processing With MATLAB Algorithms, New York: Wiley and Sons.

13. Navy, United States. United States Naval Observatory. Time Services Department. http://tycho.usno.navy.mil.

Annex A
Script to Upsample RADARSAT-1 State Vectors

```
function [latitude,longitude,HAE,position,velocity,t,th] ...
=upSampleRsat(pg,vg,day,month,year,hour,minute,second,deltaU,fs,N)
% Function to upsample Radarsat state vectors to frequency fs from 8 minute
% samples
% Usage:
% [latitude,longitude,HAE,position,velocity,t,th]
% =upSampleRsat(pg,vg,day,month,year,hour,minute,second,deltaU,fs,N)
%
% where,
% Input:
% ——
% pg         ——> Matrix of coordinates in GEI system
%                 [x1,y1,z1;x2,y2,z2;...]
% vg         ——> Matrix of velocity in GEI system
%                 [vx1,vy1,vz1;vx2,vy2,vz2;...]
% day        ——> Day of month of first time sample
% month      ——> Month of year of first time sample
% year       ——> Year (must be greater than 1979 and less than 2038)
%                 of first time sample
% hour       ——> Hour part of the day in 24 hour format hh:mm:ss.xxx
%                 of first time sample
% minute     ——> Minute of the hour in 24 hour format hh:mm:ss.xxx of
%                 first time sample
% second     ——> Second of the minute in 24 hour format hh:mm:ss.xxx
%                 of first time sample
% deltaU     ——> Difference bewteen UT1 time and UTC time in seconds
%                 (UT1-UTC) lookup for day. Should satisfy |UT1-UTC|<0.9
% fs         ——> Sampling frequency in seconds
% N          ——> Hermite polynomial order
%
% Output:
% ——-
% latitude   ——> latitude of upsampled data in ECF frame, WGS84
%                 ellipsoid
% longitude  ——> longitude of upsampled data in ECF frame, WGS84
%                 ellipsoid
% HAE        ——> height above ellipsoid of upsampled data in ECF frame,
%                 WGS84 ellipsoid
% position   ——> upsampled position in ECF frame
%                 [x1,y1,z1;x2,y2,z2;...]
% velocity   ——> upsampled velocity in ECF frame
%                 [vx1,vy1,vz1;vx2,vy2,vz2;...]
% t          ——> time samples in seconds since first sample
% th         ——> Vector of computed Greenwich mean sidereal time
```

```
%                              (hour angle in radians, not upsampled)
%
% Author:
% ------
% Ishuwa Sikaneta DRDC-Ottawa ishuwa.sikaneta

% Step 1: Convert GEI Cartesian coordinates to ECF Cartesian coordinates
[pe,ve,th]=rsatgei2ecf(pg,vg,day,month,year,hour,minute,second,deltaU);

% Step 2: Compute interpolating matrix
Q=qmatrix(N)^(-1);

% Step 3: Generate time arrays
[m,n]=size(pg);
if(n~=3)
        fprintf('Problem with state vector size!\n');
    fprintf('Make sure there are 3 columns for x, y and z\n');
        return
end
tOld=0:8*60:(8*60*(m-1));
t=0:1/fs:(8*60*(m-1));

% Step 4: Interpolate values with Hermite polynomial
[position,velocity]=interState(Q,pe,ve,tOld,t);

% Step 5: Convert Cartesian coordinates into latitude/longitude/HAE
%          coordinates
[latitude, longitude, HAE]=xyz2polar(position);
```

Annex B
Script to Convert RADARSAT-1 GEI Coordinates to ECEF Coordinates

```
function [pe,ve,th]=rsatgei2ecf(pg,vg,day,month,year,hour,minute,second,deltaU)
% Function to compute ecf coordinates from gei coordinates and UTC time for
% 8 minute sampled position and velocity data as is common with radarsat 2
% Usage: [pe,ve,th]=rsatgei2ecf(pg,vg,day,month,year,hour,minute,second,deltaU)
% where,
% Input:
% ——
% pg        ——> Matrix of coordinates in GEI system
%               [x1,y1,z1;x2,y2,z2;...]
% vg        ——> Matrix of velocity in GEI system
%               [vx1,vy1,vz1;vx2,vy2,vz2;...]
% day       ——> Day of month of first time sample
% month     ——> Month of year of first time sample
% year      ——> Year (must be greater than 1979 and less than 2038) of
%               first time sample
% hour      ——> Hour part of the day in 24 hour format hh:mm:ss.xxx of
%               first time sample
% minute    ——> Minute of the hour in 24 hour format hh:mm:ss.xxx of
%               first time sample
% second    ——> Second of the minute in 24 hour format hh:mm:ss.xxx of
%               first time sample
% deltaU    ——> Difference bewteen UT1 time and UTC time in seconds
%               (UT1-UTC) lookup for day. Should satisfy |UT1-UTC|<0.9
%
% Output:
% ——-
% pe        ——> matrix of coordinates in ECF frame
%               [x1,y1,z1;x2,y2,z2;...]
% ve        ——> matrix of coordinates in ECF frame
%               [vx1,vy1,vz1;vx2,vy2,vz2;...]
% th        ——> Vector of computed Greenwich mean sidereal time
%               (hour angle in radians)
%
% Author:
% ——-
% Ishuwa Sikaneta DRDC-Ottawa ishuwa.sikaneta

% First compute state vector size
[m,n]=size(pg);

if(n~=3)
        fprintf('Problem with state vector size"\n');
    fprintf('Make sure there are 3 columns for x, y and z\n');
        return
```

```
end

% set up output
pe=zeros(n,m);
ve=zeros(n,m);
th=zeros(m,1);

for k=1:m,
        [pe(:,k),ve(:,k),th(k)]=gei2ecf(pg(k,:),vg(k,:),day,month,year, ...
    hour,minute+(k-1)*8,second,deltaU);
end

% Transpose result to acheive desired format
pe=conj(pe)';
ve=conj(ve)';
```

Annex C
Script to Transform ECEF to Geographic Coordinates

```
function [latitude, longitude, hae]=xyz2polar(pos)
% This function converts cartesian earth centred coordinates to lat long
% and height above ellipsoid. Formulas from Tristan's note.
% Usage: [latitude,longitude,hae]=xyz2polar(pos)
% where,
% Input:
% ——
% pos         ——-> Matrix of coordinates in ECF system
%                  [x1,y1,z1;x2,y2,z2;...]
%
% Output:
% ——-
% latitude    ——-> latitude of upsampled data in ECF frame, WGS84 ellipsoid
% longitude   ——-> longitude of upsampled data in ECF frame, WGS84 ellipsoid
% hae         ——-> height above ellipsoid of upsampled data in ECF frame,
%                  WGS84 ellipsoid
%
% Author:
% ——-
% Ishuwa Sikaneta DRDC-Ottawa ishuwa.sikaneta

% Some constants
a=6378137.0;
b=6356752.3142;

f=(a-b)/a;
es=2*f-f*f;
p=sqrt(pos(:,1).^2+pos(:,2).^2);
latitude=atan(pos(:,3)*a./p*b);
ess=(a^2-b^2)/b^2;
for k=1:1000;
   la = atan2((pos(:,3)+ess*b.*sin(latitude).^3),(p-es*a*cos(latitude).^3));
   if(max(latitude-la)<1e-9)
      break;
   end
   latitude = la;
end
longitude = atan2(pos(:,2),pos(:,1));
Nphi=a./sqrt(1-es*sin(latitude).^2);
hae=p./cos(latitude)-Nphi;
latitude = latitude/pi*180;
longitude = longitude/pi*180;
```

Annex D
Script to Convert GEI Coordinates to ECEF Coordinates

```
function [pe,ve,th]=gei2ecf(pg,vg,day,month,year,hour,minute,second,deltaU)
%Function to compute ecf coordinates from gei coordinates and UTC time
% Usage: [pe,th]=gei2ecf(pg,day,month,year,hour,minute,second,deltaU)
% where,
% Input:
% ——
% pg          ——-> Triplet of position coordinates in GEI system
% vg          ——-> Triplet of velocity coordinates in GEI system
% day         ——-> Day of month
% month       ——-> Month of year
% year        ——-> Year (must be greater than 1979 and less than 2038)
% hour        ——-> Hour part of the day in 24 hour format hh:mm:ss.xxx
% minute      ——-> Minute of the hour in 24 hour format hh:mm:ss.xxx
% second      ——-> Second of the minute in 24 hour format hh:mm:ss.xxx
% deltaU      ——-> Difference bewteen UT1 time and UTC time in seconds
%                  (UT1-UTC) lookup for day. Should satisfy |UT1-UTC|<0.9
%
% Output:
% ——-
% pe          ——-> Row vector triplet of pos coordinates in ECF frame
% ve          ——-> Row vector triplet of velocity coordinates in ECF
% th          ——-> Computed Greenwich mean sidereal time
%                  (hour angle in radians)
%
% Author:
% ——-
% Ishuwa Sikaneta DRDC-Ottawa ishuwa.sikaneta

% First make sure we're looking at a row vector
pg=pg(:);
vg=vg(:);

%First compute day of year
monthDays=[0,31,59,90,120,151,181,212,243,273,304,334];
monthDayLeap=[0,31,60,91,121,152,182,213,244,274,305,335];

years=[−7305,−6939,−6574,−6209,−5844,−5478,−5113,−4748,−4383,−4017,−3652, ...
    −3287,−2922,−2556,−2191,−1826,−1461,−1095,−730,−365,0,366,731,1096, ...
    1461,1827,2192,2557,2922,3288,3653,4018,4383,4749,5114,5479,5844, ...
    6210,6575,6940,7305,7671,8036,8401,8766,9132,9497,9862,10227,10593, ...
    10958,11323,11688,12054,12419,12784,13149,13515,13880,14245,14610, ...
    14976,15341,15706,16071,16437,16802,17167,17532,17898,18263,18628, ...
    18993,19359,19724,20089,20454,20820,21185,21550,21915];
leaps=[1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0, ...
```

```
    1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1, ...
    0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1];

% Compute number of days since 01-01-2000-12:00:00.000
% (-0.5 comes from 12 noon)
days=years(year-1979)-0.5;

% Add month days
if(leaps(year-1979)==1)
        days = days + monthDayLeap(month);
else
        days = days + monthDays(month);
end

% Add day days
days = days + day - 1;

% Add hours
days = days + hour/24;

% Add minutes
days = days + minute/60/24;

% Add seconds and convert to UT1
days = days + (second+deltaU)/60/60/24;

% convert to julian centuries
t=days/36525;

% Calculate Greenwich angle
a0=67310.54841;
a1=876600*3600+8640184.812866;
a2=0.093104;
a3=-6.2e-6;

% Cacluate greenwich angle (from Fundamentals of Astrodynamics edition 2)
th=mod((a0+a1*t+a2*t^2+a3*t^3)/240*pi/180,2*pi);
% Calculate derivative of greenwich angle. Remember that straight forward
% derivative will give radians per julian century, so convert to radians
% per second by dividing by 36525 solar days per Julian century and 86400
% seconds per day.
thp=((a1+2*a2*t+3*a3*t^2)/240*pi/180)/(36525*86400);

% Compute transformation matrix
A=[cos(th),sin(th),0;-sin(th),cos(th),0;0,0,1];
% Compute derivative of transformation matrix
Ap=[-sin(th)*thp,cos(th)*thp,0;-cos(th)*thp,-sin(th)*thp,0;0,0,0];

% Compute the transformed position coordinates
pe=A*pg;
```

```
% Compute the velocity vector using product rule for differentiation
ve=Ap*pg+A*vg;
```

Annex E
Script to Compute the Hermite Interpolation Matrix

```
function Q=qmatrix(n)
% This function will generate the Hermite polynomial matrix
%
% Usage: Q=qmatrix(n);
% where,
% Inputs
% ——
% n          ———> Number of tie points
%
% Outputs:
% ———
% Q          ———> Hermite interpolating matrix
%
% Caveats:
% ———
% State vector samples have been sampled uniformly in time.
%
% Author/Affiliation:
% ————————-
% Ishuwa Sikaneta ARN/DRDC-Ottawa
%
% Date:
% —–
% November 22, 2002

% Create descending powers for time variable
k=(2*n−1):−1:0;

% Create first row
Q=[0.^k;0.^abs(k−1)];

% Create remaining rows
for l=1:(n−1);
  Q=[Q;l.^k;k.*l.^(k−1)];
end
```

Annex F
Script to Interpolate a State Vector Using Hermite Interpolation Matrix

```
function [istatePos,istateVel] ...
=interState(H,statepVec,statevVec,timeVec,interTime)
% This function seeks to interpolate state vectors given an
% interpolating matrix and the time of each state vector sample. It
% will interpolate at a range of input times.
%
% Usage: [istatepos,istatevel]
% =interState(Q,statepVec,statevVec,timeVec,interTime);
% where,
% Inputs
% ——
% Q          ———> Interpolating matrix
% statepVec ———> Triplets of position vector
% statevVec ———> Triplets of velocity vector
% timeVec  ———> Time vector of samples
% interTime ———> Vector of desired interpolation times
%
% Outputs:
% ———
% istatePos ———> Position component of interpolated state vector
% istateVel ———> Velocity component of interpolated state vector
%
% Caveats:
% ———
% State vector samples have been sampled uniformly in time. Format
% of the input position and velocity vectors is
% statepVec = [x1,y1,z1] statevVec = [xv1,yv1,zv1]
%              [x2,y2,z2]             [xv2,yv2,zv2]
%               .   .   .               .   .   .
%              [xn,yx,zn]             [xvn,yvn,zvn]
%
% Author/Affiliation:
% ——————-
% Ishuwa Sikaneta ARN/DRDC-Ottawa
%
% Date:
% —–
% November 22, 2002

% Calculate the sampling period
T = mean(diff(timeVec));

% Interleave position and velocity state vector components
[m,n]=size(statepVec);
```

```
idx=1:m;
state=zeros(2*m,n);
state((2*idx−1),:)=statepVec(idx,:);
state((2*idx),:)=T*statevVec(idx,:);

% Scale time variable
interTime=interTime(:)/T;
[p,q]=size(interTime);

% Create descending powers array
pk=(2*m−1):−1:0;

% Loop through desired time points and interpolate
for idx=1:p,
  % Create polynomial powers in time for interpolation
  ptvec=(interTime(idx)).^pk;
  vtvec=[pk.*(interTime(idx)).^abs(pk−1)];
  % Compute the interpolated position and velocity state vectors
  istatePos(idx,:)=ptvec*H*state;
  istateVel(idx,:)=vtvec*H*state/T;
end
```

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

DEFENCE R&D CANADA – OTTAWA
OTTAWA, ONTARIO, CANADA, K1A 0Z4

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

A Simple and Precise Approach to Position and Velocity Estimation of Low Earth Orbit Satellites (U)

4. AUTHORS (Last name, first name, middle initial)

Beaulne, Pierre D. and Sikaneta, Ishuwa C.

5. DATE OF PUBLICATION (month and year of publication of document)

December, 2005

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

52

6b. NO. OF REFS (total cited in document)

13

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

DRDC Ottawa Technical Memorandum

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

DEFENCE R&D CANADA – OTTAWA
OTTAWA, ONTARIO, CANADA, K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)

15eg

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Ottawa TM 2005-250

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
(   ) Distribution limited to defence departments and defence contractors; further distribution only as approved
(   ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
(   ) Distribution limited to government departments and agencies; further distribution only as approved
(   ) Distribution limited to defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

(U) This technical memorandum presents a procedure for calculating a satellite's state vector (position and velocity), with a precision of up to 10cm, at arbitrary times on its orbit, from a more sparse set of state vectors (measured or predicted) provided by the satellite itself or by its ground stationt. No orbit propagation is necessary. While very precise, the procedure's accuracy is limited to the accuracy of the input state vectors. The first part of the memo deals with time systems important in astrodynamic applications, different coordinate systems used to describe satellite state vectors, and the various coordinate transformations necessary for the task. The second part describes the use of Hermite polynomials to interpolate (upsample) new state vectors in the appropriately transformed coordinate system. In the final part, the precision of the procedure is verified by decimating an ERS-2 state vector set and using the sparse set to upsample new state vectors at the original ERS-2 rate. This interpolated set is then compared to the original measured set, and precisions of up to 10cm are observed. Finally, as an example, a set of RADARSAT-1 state vectors are upsampled to 10 Hz from their natural rate of 1/480 Hz.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

SYNTHETIC APERTURE RADAR
GEOLOCATION
TIME SYSTEMS
COORDINATE TRANSFORMATIONS
ORBIT INTERPOLATION