1998-02-16

Contributions to attitude dynamics of the Petite Amateur Navy Satellite (PANSAT)

Graswald, Markus; Panholzer, Rudolf

Monterey, California. Naval Postgraduate School

https://hdl.handle.net/10945/45993

NPS-SP-99-001

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Contributions to Attitude Dynamics of the Petite Amateur Navy Satellite (PANSAT)

by

M. Graswald
R. Panholzer

February 16, 1998

Approved for public release; distribution is unlimited.

Prepared for: NPS Space Systems Academic Group

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

RADM Robert C. Chaplin
Superintendent

R. Elster
Provost

This report was prepared for and funded by the Naval Postgraduate School, Space Systems Academic Group.

This report was prepared by:

[redacted]

Markus Graswald
Ensign, German Navy

Reviewed by:

[redacted]

Rudolf Panholzer
Space Systems Academic Group

Released by:

[redacted]

D. W. Netzer
Associate Provost and
Dean of Research

REPORT DOCUMENTATION PAGE

Form approved

OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
Feb. 16, 1999

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
Contributions to Attitude Dynamics of the Petite Amateur Navy Satellite (PANSAT)

5. FUNDING
N/A

6. AUTHOR(S)
Markus Graswald, Ensign German Navy

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Space Systems Academic Group
Code (SP), 777 Dyer Rd., Rm. 200
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-SP-99-001

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Space Systems Academic Group
Code (SP), 777 Dyer Rd., Rm. 200
Monterey, CA 93943

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
N/A

11. SUPPLEMENTARY NOTES
The views expressed in this report are those of the author and do not reflect the official policy or position of the Department of Defense, the U.S. Government, or any Government Agency.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words.)

The purpose of this technical report is to document the theoretical and practical aspects of the attitude dynamics of the Petite Amateur Navy Satellite (PANSAT). PANSAT is an experimental, small, low cost, communications satellite that was designed and built by officer students supported by the Space Systems Academic Group at the Naval Postgraduate School in Monterey, California. The satellite was deployed during the STS 95 mission into a low-Earth orbit providing an altitude of about 300 nautical miles and an inclination of 28.5° on October 30, 1998. The dynamics of the non-stabilized spacecraft are modeled by the equations of rotational motion incorporating environmental torques and considering several reference systems. The initial state of motion and attitude as launched from the Space Shuttle as well as telemetry data received after deployment are investigated in order to obtain an understanding on the satellite's actual dynamics and attitude. Visualization tools are provided to display the attitude of PANSAT at time of deployment and the roll-rate relative to the Sun.

14. SUBJECT TERMS
Attitude and Dynamics, Communications, PANSAT, Small Satellite

15. NUMBER OF PAGES
204

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Enclosure (6)

ABSTRACT

The purpose of this technical report is to document the theoretical and practical aspects of the attitude dynamics of the Petite Amateur Navy Satellite (PANSAT). PANSAT is an experimental, small, low cost, communications satellite that was designed and built by officer students supported by the Space Systems Academic Group at the Naval Postgraduate School in Monterey, California. The satellite was deployed during the STS 95 mission into a low-Earth orbit providing an altitude of about 300 nautical miles and an inclination of 28.5° on October 30, 1998. The dynamics of the non-stabilized spacecraft are modeled by the equations of rotational motion incorporating environmental torques and considering several reference systems. The initial state of motion and attitude as launched from the Space Shuttle as well as telemetry data received after deployment are investigated in order to obtain an understanding on the satellite's actual dynamics and attitude. Visualization tools are provided to display the attitude of PANSAT at time of deployment and the roll-rate relative to the Sun.

[This page is intentionally left blank.]

TABLE OF CONTENTS

I. INTRODUCTION

A. PURPOSE AND SCOPE

The purpose of this report is to document the theoretical and practical aspects of the attitude dynamics of the Petite Amateur Navy Satellite (PANSAT). PANSAT is an experimental, small, low cost, communications satellite that was designed and built by officer students supported by the Space Systems Academic Group (SSAG) at the Naval Postgraduate School (NPS) in Monterey, California. The dynamics of the spacecraft are modeled by the equations of rotational motion incorporating environmental torques and considering several reference systems. The initial state of motion and attitude as launched from the Space Shuttle as well as telemetry data received after deployment are investigated in order to obtain an understanding on the satellite's actual dynamics and attitude.

Chapter II provides the modeling of the rotational motion of the spacecraft in orbit and examines the effects of external torques, which leads to the equations of motion. The third chapter describes the orbital mechanics required to observe the dynamics of the spacecraft by providing various reference systems, orbit classifications, and coordinate transformations. Chapter IV investigates the initial conditions at the time of PANSAT's deployment. Chapter V discusses and analyzes the dynamics and attitude of the spacecraft based on solar panel data. Last but not least, the sixth chapter presents the conclusion. Several appendices follow containing further information on celestial and satellite mechanics, pictures showing the deployment of PANSAT, solar panel data obtained from the spacecraft, attitude and roll-rate visualization tools, and software source code.

B. PANSAT BACKGROUND

PANSAT is a small satellite for digital store-and-forward communications using direct sequence spread spectrum techniques. The objectives of the project are to enhance the education of officer students through the development and operation of a satellite and to serve as a proof of concept of low cost, spread spectrum, digital over-the-horizon communications. The store-and-forward capability allows NPS and amateur radio operators to send and receive messages during several short communication windows every day, each four to eight minutes in duration.

The entire satellite structure weighs approximately 125 pounds and has a diameter of about 19 inches. PANSAT is a 26-sided polyhedron, a configuration chosen to maximize solar panel area and thus power generation. The satellite was launched into a low-Earth orbit (LEO) as a secondary payload by means of the Space Shuttle Hitchhiker Program. The LEO providing an altitude of about 300 nautical miles and an inclination of 28.5° will offer an orbit lifetime in excess of four years. Since PANSAT has neither a propulsion system nor an attitude control system, it tumbles throughout its orbit.

The long mission life requires regular charging and discharging cycles of the nickel-cadmium batteries depending on the motion of the satellite in orbit and its attitude with respect to the Sun. Furthermore, communications with the ground station located at NPS in Monterey strongly depends upon the attitude due to a non-uniform antenna pattern. Some solar panels are equipped with current sensors to monitor the output current of these panels. This data is transmitted to the ground station and utilized for the determination of the attitude dynamics of PANSAT.

II. MODELING THE DYNAMICS OF THE SPACECRAFT

In order to investigate the rotational motion of PANSAT, a dynamics model is required. This is done be taking the basic equations of classical mechanics and applying them specifically for PANSAT. Furthermore, an estimation of the external torques follows, because the motion of a spacecraft is influenced by them. Using the adopted dynamics equations and the dominant torques leads finally to the desired equations of rotational motion.

A. THE BASIC MECHANICAL EQUATIONS

The classical mechanical equations are given by the differential vector equations of the linear momentum, the angular momentum and the kinematical relationship

$$\frac{\mathrm{d}\mathbf{B}}{\mathrm{d}t}=\frac{\mathrm{d}}{\mathrm{d}t}(m\cdot\mathbf{v})=\mathbf{F}, \tag{2.1}$$

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}t}=\mathbf{M}, \tag{2.2}$$

$$\frac{\mathrm{d}\mathbf{L}}{\mathrm{d}t}=\frac{\mathrm{d}^{F}\mathbf{L}}{\mathrm{d}t}+\boldsymbol{\omega}\times\mathbf{L}, \tag{2.3}$$

where $\mathbf{B}$ is the linear momentum of a particle, $\mathbf{L}$ is the angular momentum (or moment of momentum), m is the mass and $\mathbf{v}$ is the absolute velocity of a (momentarily) arbitrary particle dm. $\mathbf{F}$ is the resulting external force and $\mathbf{M}$ the resulting external torque. Equation (2.3) expresses the relationship of a time derivative $\mathrm{d}/\mathrm{d}t$ of an arbitrary vector (here the angular momentum) in an inertial reference system and the time derivative of a rotating coordinate system. The latter system rotates with an angular velocity $\boldsymbol{\omega}$.

Since the system of above equations is of 12th order and the linear motion in the orbit and the rotational motion are coupled, it is a challenging problem to calculate the motion of a satellite. Because in this case the rotational motion is only of interest, it is desired to simplify the system of equations (2.1) – (2.3).

Comparing the dimensions of the spacecraft and its distance from the center of Earth and neglecting terms of second order or higher, the orbital and rotational motion can be solved separately [Ref. 11]. This is feasible because the satellite is only 50 cm in diameter and its altitude is about 550 km, therefore $(6380\,\mathrm{km}+550\,\mathrm{km})/0.5\,\mathrm{m} << 1$. Hence, the rotational motion of PANSAT can be determined by Equation (2.2) under consideration of the kinematical relationship, Equation (2.3).

The angular momentum can also be calculated by the inertia tensor $\mathbf{I}$ and the angular velocity $\boldsymbol{\omega}$

$$\mathbf{L}^{\mathrm{c.m.}} = \mathbf{I}^{\mathrm{c.m.}}\,\boldsymbol{\omega}. \tag{2.4}$$

Equation (2.4) is referred to the center of mass, c.m., because the reference system is accelerated, i.e., not an inertial one.

Applying Equation (2.4) on Equation (2.2) and considering (2.3), the following result can be reached

$$\mathbf{I}^{\mathrm{c.m.}}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega}\times\left(\mathbf{I}^{\mathrm{c.m.}}\boldsymbol{\omega}\right) = \mathbf{M}^{\mathrm{c.m.}}. \tag{2.5}$$

Presuming a principal axis system, the components of Equation (2.5) can be written as follows

$$\begin{aligned} I_1\dot{\omega}_1 - \left(I_2 - I_3\right)\omega_2\omega_3 &= M_1, \\ I_2\dot{\omega}_2 - \left(I_3 - I_1\right)\omega_3\omega_1 &= M_2, \\ I_3\dot{\omega}_3 - \left(I_1 - I_2\right)\omega_1\omega_2 &= M_3. \end{aligned} \tag{2.6}$$

Equation (2.6) are known as Euler's equations, where M_1, M_2 and M_3 are the external torques about the principal axes. They are discussed in the succeeding section.

B. ESTIMATION OF EXTERNAL TORQUES

The motion of a spacecraft in its orbit is strongly influenced by external torques, which results in attitude changes. The external torque depends on the geometry of the satellite, its mass and the mass distribution, the attitude, and the state of motion [Ref. 12].

External torques have different causes and can be summarized as follows [Ref. 12]

- Aerodynamic torques,
- Gravity gradient torques,
- Solar-pressure torques,
- Magnetic Torques,
- Electrical Torques,
- Statistical Torques.

These individual portions of **M** were investigated in the past (e.g., [Ref. 4, 11, 12]) and, thus, are only described briefly in order to estimate their influence on the motion of PANSAT.

For a classification of altitudes, orbits are naturally separated by the radiation environment and the Earth's atmosphere. Therefore, altitudes below 1000 km are referred to as low-Earth orbits, while geosynchronous orbits (GEO) are well above the Van Allen Belts. Mid-range altitudes are relatively exceptional due to the region's increased radiation environment [Ref. 1].

Environmental torques tend to have a simple dependence on vehicle position [Ref. 4]. Aerodynamic torques are significant in altering the attitude of satellites in LEO. They decrease exponentially with altitude according to the simplest model atmospheres. Gravity gradient torques and magnetic torques vary as σ^{-3}, where σ is the distance to the gravitational center, the center of Earth, or magnetic center, respectively; magnetic torques are caused by interactions of ferromagnetic materials in the satellite with the magnetic field of the Earth. Solar-pressure torques are essentially constant for Earth orbits, varying inversely as the square of the distance from the Sun. Existence of a radial electrical field can cause interactions with the spacecraft, leading to electrical torques. In general, they are much smaller than aerodynamic or magnetic torques. Last but not least, statistical torques are provoked by collisions with meteoroids. Because the probability of such an event is low, particularly in the vicinity of the Earth, they are not taken into account.

Apparently, the strongest influences on the satellite's motion are caused by interactions with the atmosphere and the gravitational field of the Earth. This is because PANSAT is deployed in a low-Earth orbit with an altitude of approximately 550 km. However, the spacecraft has body-mounted solar panels instead of large deployable solar arrays making the momentum arm short for solar-pressure torques. Magnetic and especially electrical fields caused in the satellite are considered small, because the structure consists mainly of aluminum and those fields are not used for stabilizing the spacecraft. Therefore, the main sources of torques acting on PANSAT, the atmospheric and gravity gradient torque, are estimated in the following sections.

1. Aerodynamic Torque

Aerodynamic torques are primarily influenced by the orbit altitude and spacecraft configuration. They are defined as follows [Ref. 7]

$$\mathbf{M}_a = \mathbf{r} \times \mathbf{F}_a, \tag{2.7}$$

where $\mathbf{r}$ is the displacement vector measured from the center of mass to the center of pressure and $\mathbf{F}_a$ is the aerodynamic force. It is given by

$$\mathbf{F}_a = \frac{1}{2}\rho\, \mathbf{v}^2 A C_D, \tag{2.8}$$

where $1/2\,\rho\,\mathbf{v}^2$ is referred to as the dynamic pressure with ρ as the atmospheric density and $\mathbf{v}$ the velocity; A is the cross-sectional area, to which the force and the velocity vector is perpendicular to, and C_D the drag coefficient.

In the following part, the different coefficients of Equations (2.7) and (2.8) are determined for an estimation of the torque.

Although the overall behavior of atmospheric density is well established, there are many diverse physical processes that occur in the thermosphere. Therefore, the density does not only vary with altitude, but also with temperature, which is affected by solar activity, measured by the F10.7 index (solar flux index), and a day-to-night rhythm, known as diurnal variations. Inspection of altitudes between 500 and 600 km reveals that the density variations between solar maximum and minimum are roughly one order of magnitude, while the maximum daytime density may be five times greater than the nighttime minimum [Ref. 1].

Tables of the observed and predicted solar flux index show that in 1997/98 a solar minimum occurs and the next solar maximum will occur in 2001/02 [Ref. 6]. Therefore, the atmospheric density is interpolated from the 500 and 600 km solar minimum and daytime values for an altitude of 550 km [Ref. 1]

$$\begin{aligned}\rho &= 2.0\cdot10^{-13}\,\mathrm{kg/m^3} + \left(3.9\cdot10^{-14} - 2.0\cdot10^{-13}\right)\mathrm{kg/m^3}\,\frac{550-500}{600-500}.\\ &= 1.2\cdot10^{-13}\,\mathrm{kg/m^3}\ @\ 550\,\mathrm{km}\end{aligned}$$

Since the orbit of PANSAT is circular as a first-order approximation, its velocity can be calculated from the two-body equation of motion [Ref. 1]

$$v = \sqrt{\frac{K}{\sigma}} \tag{2.9}$$

where K is the Earth's gravitational constant and σ is the distance of the spacecraft from the center of the Earth. Using an altitude of 550 km (the values of the geocentric gravitational constant and Earth's radius can be found in Appendix E), the velocity in question is

$$\begin{aligned}v &= \sqrt{\frac{3.986\cdot10^{14}\,\mathrm{m^3/s^2}}{6928\cdot10^{3}\mathrm{m}}}.\\ &= 7590\,\mathrm{m/s}\end{aligned}$$

The determination of the atmospheric drag coefficient for any given body is quite difficult. For spacecrafts above 200 km, it is conventional to use a coefficient of 2.2 for a spherical body [Ref. 15].

The spacecraft has a highly symmetric structure consisting of a 26-sided polyhedron and the launch vehicle interface (LVI) at the bottom (Figure 1.1). For an estimate of the aerodynamic torque, the largest cross-sectional area is taken. In this case, the axis of symmetry is normal to the direction of motion. The center of pressure is assumed the center of the cross-sectional area.

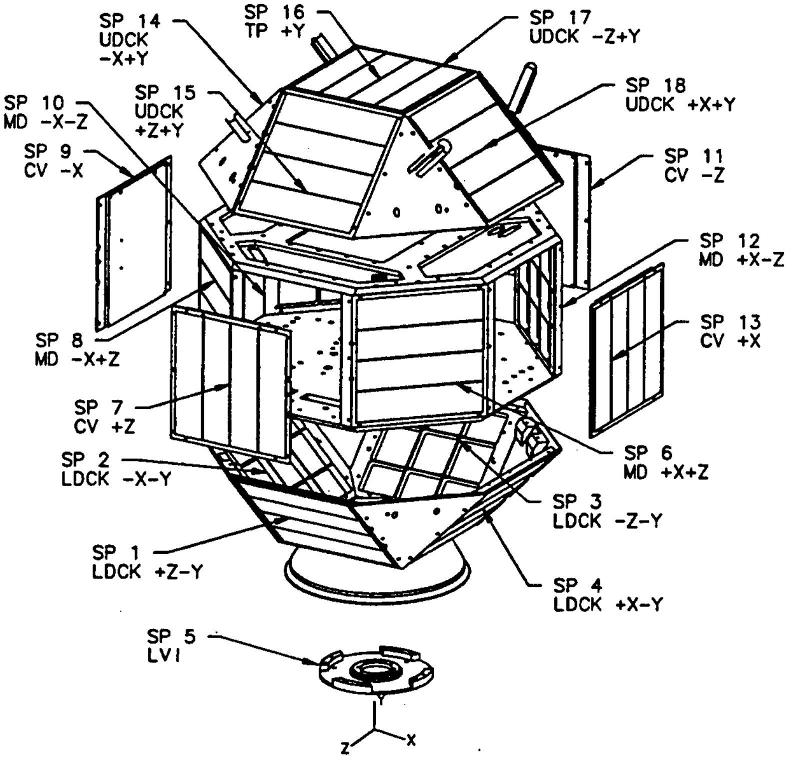


Figure 1.1: Schematic of PANSAT Showing the Geometrical Axes and Identifying the Solar Panels

The cross-sectional area of the polyhedron is calculated as 0.1580 m^2 and of the LVI as 0.0107 m^2, given an overall area of 0.1687 m^2. The center of the overall cross-sectional area lies on the geometrical axis of symmetry, Y^1. Therefore, only the location of the center of the area on this axis needs to be determined. This is done by applying Steiner's law on the areas

$$y_{OA} \cdot A_{OA} = y_{PH} \cdot A_{PH} + y_{LVI} \cdot A_{LVI} . \tag{2.10}$$

Solving Equation (2.10) for y_{OA}, it can be determined to be (with respect to the X, Y, Z-frame)

$$\begin{aligned} y_{OA} &= \left(\frac{0.268 \cdot 0.1580 + 0.025 \cdot 0.0107}{0.1687} \right) \mathrm{m} . \\ &= 25.3\,\mathrm{cm} \end{aligned}$$

Finally, the 'center of pressure' for this cross-sectional area, where the LVI and the main body are facing to the forward direction of motion, is

$$x^{c.p.} = 0,$$
$$y^{c.p.} = 25.3 \text{ cm},$$
$$z^{c.p.} = 0.$$

The 'center of mass' of the spacecraft, as determined experimentally at NASA Goddard Space Flight Center (GSFC), Greenbelt, MD, is given with respect to the X, Y, Z-system as

$$x^{c.m.} = 0.1 \text{ cm},$$
$$y^{c.m.} = 23.2 \text{ cm},$$
$$z^{c.m.} = -0.2 \text{ cm}.$$

Neglecting the small values of the X and Z axis, the displacement of the center of pressure to the 'center of mass' in that given area is $r = 2.1 \text{cm}$.

With these parameters, the aerodynamic force is calculated by using Equation (2.8) in scalar form

$$\begin{aligned} F_a &= \frac{1}{2} 1.2 \cdot 10^{-13} \text{ kg/m}^3\, 7590^2 \text{ m}^2/\text{s}^2\, 0.1690 \text{m}^2\, 2.2 \\ &= 1.29 \cdot 10^{-6} \text{N} \end{aligned},$$

and the aerodynamic torque in question by applying Equation (2.7) in scalar form

$$\begin{aligned} M_a &= 2.1 \cdot 10^{-2} \text{m} \cdot 1.29 \cdot 10^{-6} \text{N} \\ &= 2.7 \cdot 10^{-8} \text{Nm} \end{aligned}$$

This is a worst-case value based on the current conditions in space. It may be one order higher for increased solar activity, which is expected in the next few years.

2. Gravity Gradient Torque

In this section, the gravity gradient torque acting on a spacecraft in the central symmetric gravitational field of the Earth is estimated. The following considerations are based on some simplifications. The Earth's oblateness and its impact on the gravity field

[1] The center of the body fixed coordinate frame X, Y, Z shown in Figure 1.1 is located at the base of the LVI.

is neglected. The second major assumption is that the equations are based on a two-body problem, neglecting the influence of the Sun and the Moon.

In considering the coordinate system for the determination of the gravity torque, there is a natural orientation of the body where the gravity torque is zero. It serves as a logical reference position and deviation of the body from this reference results in a gravity torque. This reference system is known as the orbit-plane coordinate system. The 'center of mass' of the spacecraft is taken as the origin. Axis O_3 points toward nadir, the 'center of Earth', and O_2 is the negative orbit normal. Axis O_1 is directed in the forward direction of the orbital motion and completes the right-handed Cartesian system.

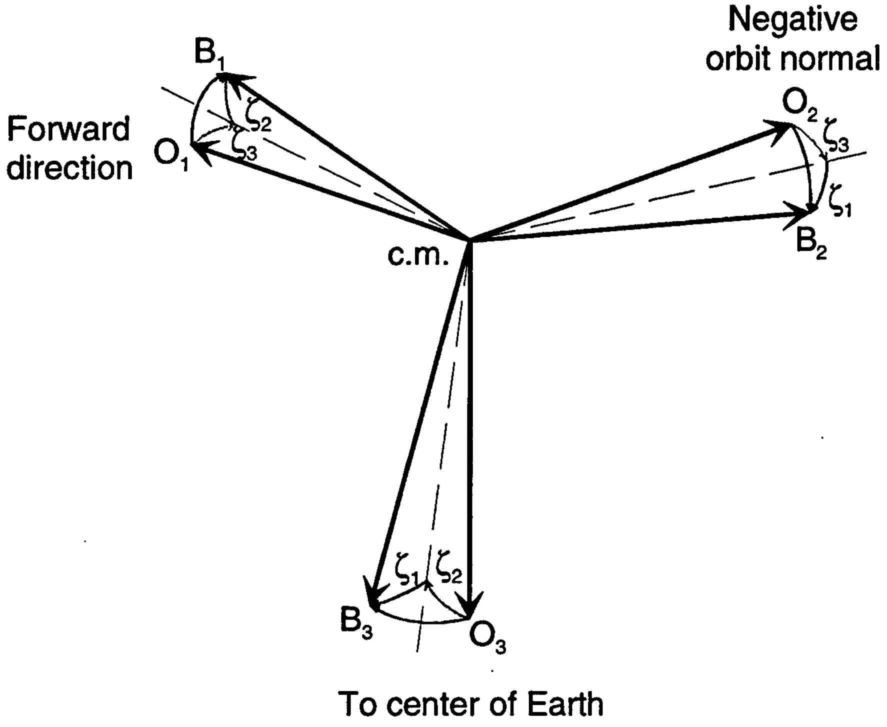


Figure 1.2: Transformation of Orbit-plane Axes to Body Axes

The body fixed axes B_1, B_2, and B_3 are defined by the following sequence of three individual rotations. In the nominal orientation, the body and the orbit-plane axes are coincident. First, allow a rotation ζ_3 about O_3. The angle ζ_2 rotates about the newly displaced axis 2 and the final rotation ζ_1 about the final position of axis 1 completes the sequence, as shown in Figure 1.2. Therefore, the angles ζ_i , which are called Euler angles, define the deviations of the body axes from the orbit-plane axes[2].

The following gravity gradient torque equations are derived from Newton's gravity law for a principal axis system with its origin located at the 'center of mass' [Ref. 7]

$$\begin{aligned} M_{g,1} &= \frac{3K}{2\sigma^3}(I_3 - I_2)\sin 2\zeta_1 \cos^2 \zeta_2 , \\ M_{g,2} &= \frac{3K}{2\sigma^3}(I_3 - I_1)\sin 2\zeta_2 \cos \zeta_1 , \\ M_{g,3} &= \frac{3K}{2\sigma^3}(I_1 - I_2)\sin 2\zeta_2 \sin \zeta_1 , \end{aligned} \tag{2.11}$$

where $M_{g,i}$ are the components of the gravity gradient torque, and I_i are the principal moments of inertia. It should be noted that the angle ζ_3 does not enter the above equations since a rotation about axis 3 produces no gravity torque.

In the worst-case, the biggest gravity torque occurs about axis 2 due to the highest difference of the moments of inertia (see Appendix C). For estimation purposes, the satellite's altitude is taken as 550 km and the direction cosines are assumed as $|\sin 2\zeta_2 \cos \zeta_1| = 1$ leading to

$$\begin{aligned} M_{g,2} &= \frac{3 \cdot 3.986 \cdot 10^{14}\,\mathrm{m^3/s^2}}{2\left(6928 \cdot 10^3\right)^3 \mathrm{m^3}}(0.98 - 1.42)\mathrm{kg\,m^2} . \\ |M_{g,2}| &= 7.9 \cdot 10^{-7}\,\mathrm{Nm} \end{aligned}$$

Thus, the gravity gradient torque is approximately one order higher than the aerodynamic torque and thus the dominant torque in altering the attitude of the spacecraft.

C. EQUATIONS OF ROTATIONAL MOTION

In order to obtain the final equations of rotational motion, the ζ angles introduced in the last section are applied on Equation (2.6) and the external torques are given by Equation (2.11). Hence, the equations of motion are

[2] Further treatment on these coordinate systems and Euler angles is given in Chapter III.

$$I_1\ddot{\zeta}_1 - (I_2 - I_3)\dot{\zeta}_2\dot{\zeta}_3 = \frac{3K}{2\sigma^3}(I_3 - I_2)\sin 2\zeta_1 \cos^2 \zeta_2 ,$$
$$I_2\ddot{\zeta}_2 - (I_3 - I_1)\dot{\zeta}_3\dot{\zeta}_1 = \frac{3K}{2\sigma^3}(I_3 - I_1)\sin 2\zeta_2 \cos \zeta_1 , \tag{2.12}$$
$$I_3\ddot{\zeta}_3 - (I_1 - I_2)\dot{\zeta}_1\dot{\zeta}_2 = \frac{3K}{2\sigma^3}(I_1 - I_2)\sin 2\zeta_2 \sin \zeta_1 .$$

As discussed earlier, this nonlinear 6th order system of differential equations can be solved independently from the translational motion in the orbit, i.e., in the body fixed coordinate system. In the next chapter, various references in space and their coordinate transformations required in this report are introduced.

III.ORBITAL MECHANICS

In this chapter, topics involved in orbital mechanics, which are used in this report, are presented. Several reference systems in space are introduced and coordinate transformations between them are given. A brief discussion on elliptical orbits is included even as the orbit of the spacecraft can be treated as circular. This is necessary because the tracking data of the satellite obtained from Air Force Space Command contains elliptical elements. Last but not least, the sun position vector, called solar vector, is determined, because the sun is the only celestial object used to obtain the attitude of the spacecraft.

A. REFERENCE SYSTEMS IN SPACE

To investigate the dynamics of a satellite in space there are different reference systems required. When describing the attitude and motion of the spacecraft many possible references exist. The three basic choices of coordinate systems are those fixed in inertial space, those fixed relative to the body of the spacecraft, and those defined relative to the orbit and not fixed to either the spacecraft or inertial space [Ref. 2]. Furthermore, in order to investigate the attitude with respect to the ground station, an observer-fixed reference system rotating with the Earth is adopted.

1. Inertial Systems

The most common inertial coordinate system for studying the motion of Earth-orbiting satellites is the geocentric system defined relative to the rotation axis of the Earth (via S_2) and using the equator as a fundamental plane, as shown in Figure 3.1. The origin of the coordinate frame is the center of the Earth. The essential reference axis S_3 is directed to a point fixed in space, known as vernal equinox[3], while the first one completes

[3] The vernal equinox refers to the Northern Hemisphere, when the line of equinoxes does pass through the sun occurring on or about March 21. The position of the line of equinoxes on the celestial sphere is also called the First Point of Aries. The vernal equinox has separated by nearly 30° from the constellation from which it took originally its name. Therefore, the name First Point of Aries is not used here.

the right-handed system, so that $\mathbf{S}_1 = \mathbf{S}_2 \times \mathbf{S}_3$. The third axis is commonly called the line of equinoxes. It is the intersection between the ecliptic and equatorial plane, which the Earth passes twice a year. This occurs on or about the first day of spring and autumn.

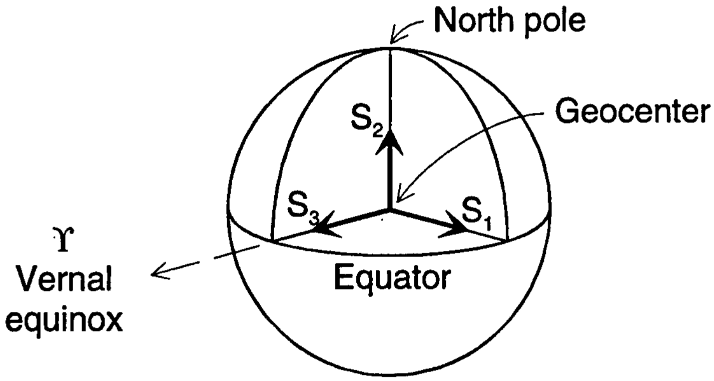


Figure 3.1: Geocentric Inertial Reference System [Ref. 7]

Inertial reference systems imply that Newtonian mechanics hold. They require that the coordinate system either is in the state of absolute rest or moves unaccelerated in space. Unfortunately, the geocentric inertial coordinate system[4] is not truly inertial because it is not fixed to the mean positions of the stars in the vicinity of the Sun. The equatorial plane of the Earth and the ecliptic are not actually fixed relative to the stars, which is caused by gravitational forces of the Moon, the Sun and the planets. This precession of the equinoxes results in a shift of the position of the vernal equinox relative to the fixed stars at a rate of 0.014° per year. Because of this slow drift, geocentric inertial coordinates require a corresponding date to accurately define the position of the vernal equinox. An inertial coordinate system can be almost achieved by referring it to a particular epoch; B1950, J2000, and True-of-Date (TOD) are commonly used. The standard epoch used throughout this report is J2000. Nevertheless, NASA still uses the old B1950 system, i.e., the attitude information at PANSAT deployment is referred to this epoch. Therefore, further treatment is given in Appendix B and in Chapter IV.

[4] Also known as ECI coordinates (Earth-Centered Inertial).

2. Orbit-plane System

The orbit-plane reference system, briefly introduced in Section II.B.2., is a coordinate system that maintains its orientation relative to the Earth as the spacecraft moves in its orbit. These coordinates are also known as roll, pitch, yaw (RPY)[5], and are illustrated in Figure 3.2. The origin is the 'center of mass' of the spacecraft. In this system, the axis O_3 is directed toward the nadir (i.e., toward the center of Earth), the second axis is directed toward the negative orbit normal, and O_1 completes the right-handed system, so that the unit vectors fulfil the relation $\mathbf{O}_1 = \mathbf{O}_2 \times \mathbf{O}_3$. Thus, in a circular orbit, O_1 will be along the velocity vector.

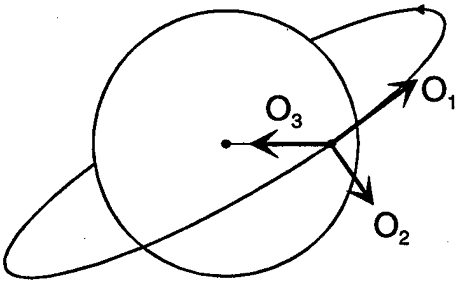


Figure 3.2: Orbit-plane Axes [Ref. 7]

Figure 3.3 depicts the relationships between the orbit-plane system and the geocentric inertial system for a circular orbit. The angles ξ_1, ξ_2, ξ_3, define the orientation of the orbit with respect to the Earth. They are defined as right-handed rotations about their respective axes. The angle ξ_1 is known as the right ascension of the ascending node (RAAS) and gives the orientation of the line of nodes on the equatorial plane. The inclination ξ_2 inclines the orbit with respect to the equator, and ξ_3 , the argument of perigee, gives the position of the satellite in the orbit referred to the line of apsides.

[5] RPY systems can be also defined as spacecraft-fixed coordinates rather than orbit-defined coordinates. The latter system is frequently called LVLH (Local Vertical/Local Horizontal) in the literature.

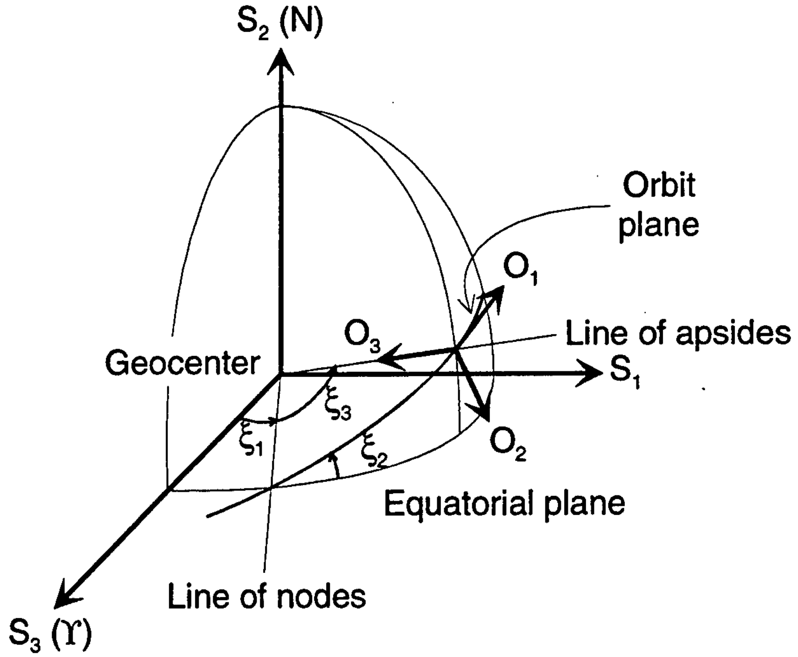


Figure 3.3: Definition of a Circular Orbit with respect to Geocentric Axes

3. Body-fixed Reference Systems

There are different choices in defining body-fixed coordinates. For the purpose of analysis of satellite dynamics, it is useful to define principal axes as body axes and using the 'center of mass' as the origin of the coordinate frame. They are designated by B_i, $i = 1, 2, 3$.

Furthermore, geometrical axes X, Y, Z are used in designing the satellite. The origin is located at the baseplate of the LVI and the directions are shown in Figure 1.1. X', Y', Z' is a body-fixed coordinate system using the same axis directions, but its origin is moved into the 'center of mass' of the spacecraft. The latter body coordinate system is useful in evaluating the telemetry data concerning the attitude with respect to the Sun. Additional information about transformations between those body-fixed systems is given in Appendix C.

4. Observer-fixed Systems

Observer-based systems are necessary in order to determine "look angles" to view the satellite from a ground station. Because PANSAT is a tumbling satellite, it is desired to determine the attitude with respect to the ground station to observe the impact on the transmission quality.

Allowing the geocentric coordinate system to rotate with the Earth, one can obtain the Earth-Centered, Earth-Fixed, ECEF, coordinate system, designated by F_i, $i = 1, 2, 3$. Like for the geocentric inertial coordinate system, the fundamental plane is defined by the equator. The main difference is that the primary axis is always aligned with a particular meridian, here the Greenwich meridian. This system is required as a step to reach a system fixed to the site of the viewer.

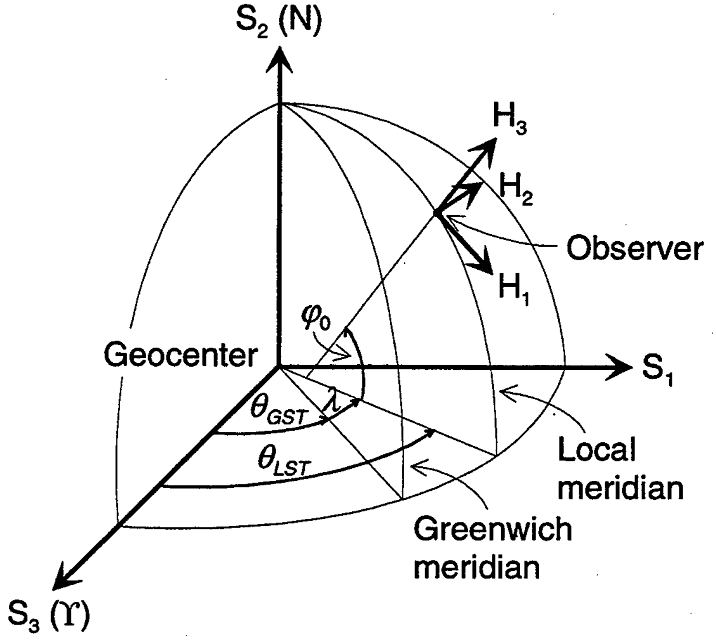


Figure 3.4: Topocentric Horizon Reference System

A truly observer-based coordinate system is the topocentric, i.e., local, horizon reference system, H_i, $i = 1, 2, 3$. This system rotates with the site and is shown in Figure 3.4. The local horizon forms the fundamental plane, which is normal to the axis pointing to zenith, H_3. This line defines also the geodetic (geographic) latitude. The second axis points east from the site and is undefined for the North or South Pole. Finally, the axis H_1 points due south from the site.

B. SATELLITE STATE REPRESENTATIONS

To define the state of a satellite in space six quantities are needed. This can be done using either a state vector, usually associated with position and velocity vectors, or an element set, typically used with scalar magnitude and angular representations of the orbit [Ref. 3].

Either set of quantities completely specifies the two-body orbit. The classical orbital elements and the two line elements belong to the second class and are discussed in this section after a short introduction to elliptical orbits.

1. Elliptical Orbit

As observed by Kepler, all satellites move in elliptical orbits around the Earth. PANSAT's orbit can be approximated by a circle, but the orbital elements are provided in a general form. Therefore, relationships to describe the motion in such an orbit are given in the following.

Geometry of a Ellipse

The polar equation of an ellipse with the origin at one focus is [Ref. 9]

$$\sigma = \frac{a(1-e^2)}{1+e\cos\tau}, \tag{3.1}$$

where the angle τ is measured from the line joining the occupied focus and the curve's closest point of approach to that focus. The semi-major axis is denoted by a and the eccentricity by e. If the semi-minor axis is b, the eccentricity is defined by

$$e = \sqrt{\frac{a^2 - b^2}{a^2}}. \tag{3.2}$$

The special case $e = 0$ yields a circular orbit ($a = b = \sigma$).

The angle τ is known as true anomaly. In the case of the Earth, the closest point a satellite can get to its focus is called perigee ($\tau = 0$) and the point farthest away apogee.

Kepler's Equation

Kepler's equation for elliptical orbits is derived from observations of the planets under the assumption of two-body motions. It may be stated as follows [Ref. 10]

$$M = n(t - T) = M_0 + n(t - t_0) = E - e\sin E, \tag{3.3}$$

where M is the mean anomaly, which is essentially defined by $n(t - T)$, n is the mean motion which can be determined from the given orbit period T_O by

$$n = \frac{2\pi}{T_O} = \sqrt{\frac{K}{a^3}}. \tag{3.4}$$

In this equation, the mass of the satellite is neglected compared to the mass of the Earth. M_0 is the value of M at a particular time, the epoch, and E is called the eccentric anomaly (Figure 3.5). Kepler's equation is a useful link to relate the relative position of the satellite in the ellipse (through E) to time, which is directly proportional to the mean anomaly M.

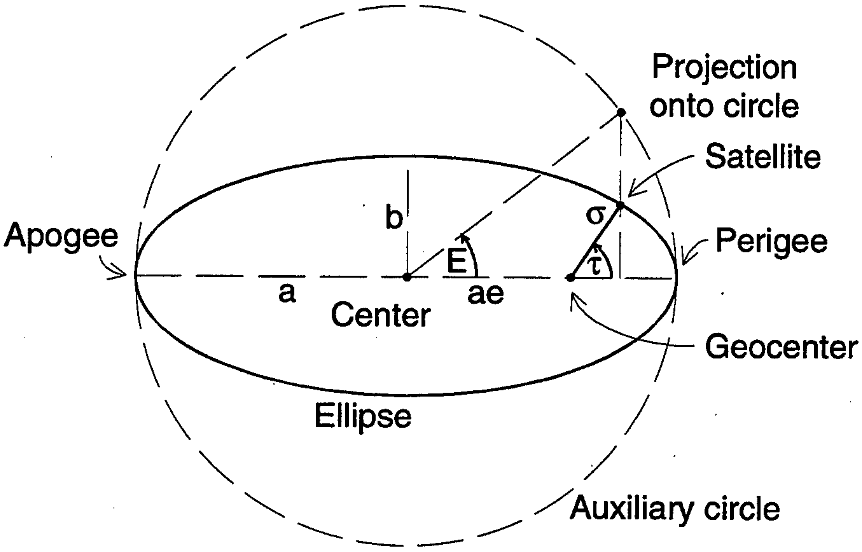


Figure 3.5: Geometry of an Ellipse

To determine the position of the satellite, the following equations are required to calculate an angle referred to the focus, the true anomaly. The solution of equation (3.3) can be obtained by a series expansion in powers of e, which was derived by Plummer [Ref. 10]

$$E = M + 2\sum_{i=1}^{n} \frac{\sin(iM)}{i} J_i(ie), \tag{3.5}$$

where $J_i(ie)$ are Bessel functions of the first kind and i is an index.

The distance from the center of the Earth to the 'center of mass' of the spacecraft at any particular time is calculated as follows

$$\sigma = a(1 - e\cos E). \tag{3.6}$$

In order to avoid singularities in determining the true anomaly, an approximation for numerical calculations is used. The relationship gives the true anomaly as a function of the mean anomaly [Ref. 9]

$$\begin{aligned}\tau = M + e\left(2 - \frac{1}{4}e^2 + \frac{5}{96}e^4\right)\sin M + e^2\left(\frac{5}{4} - \frac{11}{24}e^2 + \frac{17}{192}e^4\right)\sin 2M \\ + e^3\left(\frac{13}{12} - \frac{129}{192}e^2\right)\sin 3M + \ldots\end{aligned} \tag{3.7}$$

2. Classical Orbital Elements

The most common element set one encounters in working with satellite states are the classical orbital elements. Figure 3.6 shows some of these classical orbital elements. The semi-major axis a and the eccentricity e describe the scale and the shape, respectively, of the elliptical orbit. The position with respect to the Earth, the geocentric inertial axes, is stated by the inclination ξ_2, the right ascension of the ascending node ξ_1, and the argument of perigee ξ_3. The true anomaly gives the location of the spacecraft on the ellipse referred to the line to pericenter.

For convenience, the argument of perigee and the true anomaly can be combined to the angle $\bar{\xi}_3$. As a result, the coordinate transformation matrices remain identical for circular and elliptical orbits.

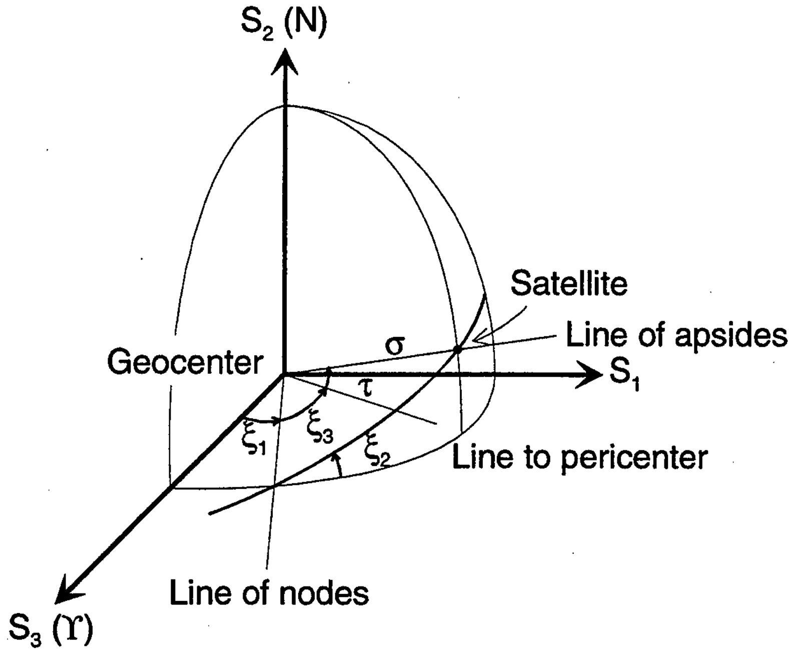


Figure 3.6: Classical Orbital Elements

3. Two Line Elements

Two line elements (TLE) are orbital tracking data that is intended to be used by satellite tracking programs like the Satellite Tool Kit (STK). They contain the classical orbital elements in a compact form to compute the position of the satellite at any particular moment in time based on its previous location and known velocity. They are updated and provided by the U.S. Air Force Space Command.

The first six values represent the independent quantities required for calculations, while the remaining variables (mean motion rate, mean motion acceleration, and B^*, a drag-like parameter) are needed to describe the effect of perturbations on satellite motion. The time is based on UTC. They contain the following variables

$$n,\ e,\ \xi_1,\ \xi_2,\ \xi_3,\ M,$$
$$\frac{\dot{n}}{2},\ \frac{\ddot{n}}{2},\ B^*,\ UTC.$$

Further treatment on this subject can be found in [Ref. 3].

C. COORDINATE TRANSFORMATIONS

When referring position or velocity vectors from one coordinate system to another, coordinate transformations have to be carried out. The most common one transforms a set of reference axes in inertial space to a set in the spacecraft. Alternative parameterizations for this transformation with its advantages and disadvantages are described below. The transformations to different required reference frames, as introduced in the last section, are shown afterwards.

1. Parameterization of Attitude

To define the attitude of a rigid body, an orthogonal, right-handed triad **u**, **v**, **w** of unit vectors is necessary, which satisfies the relationship $\mathbf{u} \times \mathbf{v} = \mathbf{w}$. Specifying the components along the three axes of the coordinate frame will fix the orientation completely. This requires nine parameters, which can be regarded as the elements of a 3×3 matrix, called the attitude matrix **T**

$$\mathbf{T} = \begin{bmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{bmatrix}, \tag{3.8}$$

where $\mathbf{u} = [u_1 \quad u_2 \quad u_3]^T$, $\mathbf{v} = [v_1 \quad v_2 \quad v_3]^T$, $\mathbf{w} = [w_1 \quad w_2 \quad w_3]^T$. Each of these elements is the cosine of the angle between a body unit vector and a reference axis. For this reason, **T** is often referred to as the direction cosine matrix [Ref. 2].

The direction cosine matrix, DCM, is a coordinate transformation that maps vectors from a reference frame to the body frame. It can be shown that a proper real orthogonal matrix transformation preserves the length of vectors and the angles between them, and thus represents a rotation. The product of two proper real orthogonal matrices $\mathbf{T}'' = \mathbf{T}'\mathbf{T}$ represents the results of successive rotations by **T** and **T'** in that order. An useful property of an orthogonal matrix is that the transpose and inverse are identical. Thus, $\mathbf{T}^T$ maps vectors from the body frame to the reference frame [Ref. 2].

The direction cosine matrix is the fundamental quantity specifying the orientation of a rigid body. It is free of singularities and contains no trigonometric functions allowing fast computations. However, it requires nine elements and, therefore, carries six redundant parameters. That is why other parameterizations are discussed below.

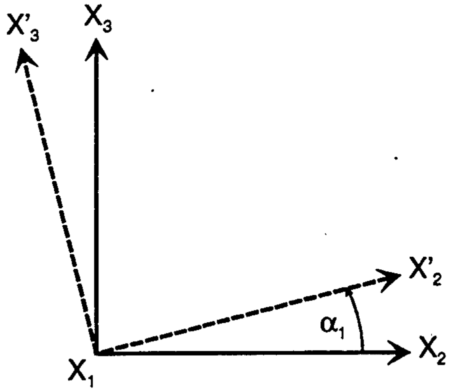
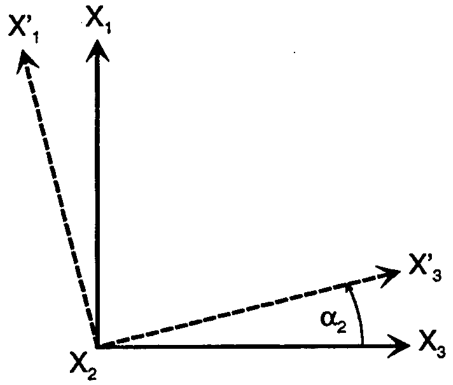
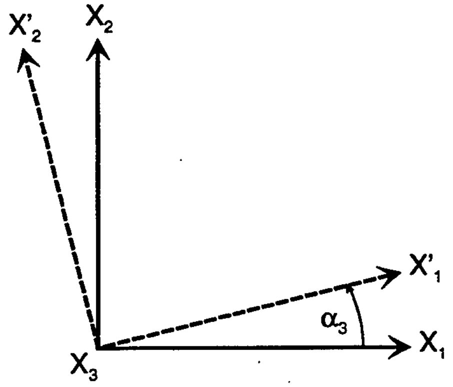
Euler Angles

Euler angles allow a parameterization with three independent rotation angles, which is the minimum number of parameters to specify the orientation of a rigid body in space. They are useful for analysis and their geometrical significance is more apparent than the other parameter sets. In order to describe a complete rotation, the three rotation angles correspond with the three degrees of freedom of a rigid body. Unfortunately, singularities can appear for some angles. Nevertheless, Euler angles are needed for dynamic analysis and, therefore, are used throughout this report.

In order to transform a triad from one reference frame to another, three finite rotations about the three orthogonal axes are required. Each individual rotation is represented by a rotation matrix, given in Table 3.1. To describe a complete rotation sequence, allow first a rotation about axis 3 with rotation angle 1, then about axis 1 with angle 2, and finally about axis 2 with the angle 3. This arbitrary sequence of rotations is called a 3-1-2 sequence. The direction cosine matrix for the overall rotation sequence is obtained by multiplying the three matrices for the single-axis rotations, with the first rotation matrix on the right and the last on the left

$$\mathbf{T}_{3-1-2}(\alpha_1,\alpha_2,\alpha_3)=\mathbf{T}_2(\alpha_3)\mathbf{T}_1(\alpha_2)\mathbf{T}_3(\alpha_1). \tag{3.9}$$

Table 3.1: Matrices Representing Finite Rotations about Each Axis of a Cartesian Coordinate System [Ref. 7]

Axis of Rotation	*Sketch*	*Matrix Representation*
1		$\mathbf{X'}=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\alpha_1 & \sin\alpha_1\\ 0 & -\sin\alpha_1 & \cos\alpha_1\end{bmatrix}\mathbf{X}$
2		$\mathbf{X'}=\begin{bmatrix}\cos\alpha_2 & 0 & -\sin\alpha_2\\ 0 & 1 & 0\\ \sin\alpha_2 & 0 & \cos\alpha_2\end{bmatrix}\mathbf{X}$
3		$\mathbf{X'}=\begin{bmatrix}\cos\alpha_3 & \sin\alpha_3 & 0\\ -\sin\alpha_3 & \cos\alpha_3 & 0\\ 0 & 0 & 1\end{bmatrix}\mathbf{X}$

Notes:

1. **X** is a fixed vector on the original position of the axis set,
2. **X'** is the same vector on the rotated position of the axis set,
3. α_i is a rotation in the positive sense about the *ith* axis.

Quaternions

A parameterization of the direction cosine matrix in terms of quaternions[6], four in number, has proved to be quite useful in spacecraft work [Ref. 2].

A quaternion is a four-element complex number that is defined as follows

$$\mathbf{Q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix}, \tag{3.10}$$

where $\mathbf{q}$ is the vector $[q_1 \quad q_2 \quad q_3]^T$ and q_4 is a scalar. Quaternions are not independent parameters and satisfy the constraint equation

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \tag{3.11}$$

Quaternions are physically defined using Euler's Theorem which states that "the most general displacement of a rigid body with one point fixed is a rotation about some axis" [Ref. 2]:

$$\begin{aligned} q_1 &= e_1 \sin\Phi/2, \\ q_2 &= e_2 \sin\Phi/2, \\ q_3 &= e_3 \sin\Phi/2, \\ q_4 &= \cos\Phi/2, \end{aligned} \tag{3.12}$$

where $\mathbf{e}$ is the axis of rotation, the eigenaxis, and Φ the angle of rotation, the eigenvalue.

The direction cosine matrix can be expressed in terms of quaternions by

$$\begin{aligned} \mathbf{T(Q)} &= \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 - q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 + q_2q_4) \\ 2(q_1q_2 + q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 + q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}, \\ &= (q_4^2 - \mathbf{q}^2)\mathbf{E} + 2\mathbf{q}\mathbf{q}^T - 2q_4\overline{\mathbf{Q}} \end{aligned} \tag{3.13}$$

where $\overline{\mathbf{Q}}$ is the skew-symmetric matrix and defined as follows

[6] Also called Euler symmetric parameters.

$$\overline{\mathbf{Q}} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}. \tag{3.14}$$

The great advantages of using quaternions to describe an attitude are that these parameters are more compact than a direction cosine matrix, do not involve trigonometric functions for the calculation of the DCM, and avoid singularities.

2. Orbit-plane Reference System

Transforming the body fixed axes $\mathbf{B}_i$ to the orbit-plane axes $\mathbf{O}_i$, a rotation sequence 3-2-1 using the Euler angles ζ_3, ζ_2, ζ_1 is required. The rotation matrices are provided in Table 3.1. Hence, the overall transformation matrix $\mathbf{T}_{BO}$ is as follows[7]

$$\mathbf{T}_{BO} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cs\zeta_1 & sn\zeta_1 \\ 0 & -sn\zeta_1 & cs\zeta_1 \end{bmatrix} \begin{bmatrix} cs\zeta_2 & 0 & -sn\zeta_2 \\ 0 & 1 & 0 \\ sn\zeta_2 & 0 & cs\zeta_2 \end{bmatrix} \begin{bmatrix} cs\zeta_3 & sn\zeta_3 & 0 \\ -sn\zeta_3 & cs\zeta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} cs\zeta_2 cs\zeta_3 & cs\zeta_2 sn\zeta_3 & -sn\zeta_2 \\ sn\zeta_1 sn\zeta_2 cs\zeta_3 - cs\zeta_1 sn\zeta_3 & sn\zeta_1 sn\zeta_2 sn\zeta_3 + cs\zeta_1 cs\zeta_3 & sn\zeta_1 cs\zeta_2 \\ cs\zeta_1 sn\zeta_2 cs\zeta_3 + sn\zeta_1 sn\zeta_3 & cs\zeta_1 sn\zeta_2 sn\zeta_3 - sn\zeta_1 cs\zeta_3 & cs\zeta_1 cs\zeta_2 \end{bmatrix}, \tag{3.15a}$$

$$\mathbf{B} = \mathbf{T}_{BO}\mathbf{O}. \tag{3.15b}$$

To obtain the Euler angles of the 3-2-1 sequence from the direction cosine matrix elements of $\mathbf{T}_{BO}$, the following equations can be applied

$$\zeta_1 = \arctan[\mathbf{T}_{BO}(2,3)/\mathbf{T}_{BO}(3,3)], \tag{3.16a}$$

$$\zeta_2 = \arcsin[-\mathbf{T}_{BO}(1,3)], \tag{3.16b}$$

$$\zeta_3 = \arctan[\mathbf{T}_{BO}(1,2)/\mathbf{T}_{BO}(1,1)]. \tag{3.16c}$$

The angles are determined up to a twofold ambiguity except at certain values of the angle ζ_2. In this case, the singular values of ζ_2 are odd multiples of 90° [Ref. 2].

[7] As needed for formatting purposes, sine and cosine functions are abbreviated by sn and cs, respectively.

To express the angular velocities about the body axes in terms of Euler angles, the matrix given below must be applied. This is because the Euler angles are independent quantities to describe the orientation of body-fixed axes with respect to reference axes, but the rotations about the three axes are coupled. Therefore, the angular velocities $\boldsymbol{\omega}_{BR,B_i}$ referred to the orbit-plane axes are [Ref. 7]

$$\boldsymbol{\omega}_{BR,B_i} = \begin{bmatrix} \dot{\zeta}_1 - \dot{\zeta}_3 \sin\zeta_2 \\ \dot{\zeta}_2 \cos\zeta_1 + \dot{\zeta}_3 \sin\zeta_1 \cos\zeta_2 \\ -\dot{\zeta}_2 \sin\zeta_1 + \dot{\zeta}_3 \cos\zeta_1 \cos\zeta_2 \end{bmatrix}. \tag{3.17}$$

3. Geocentric Inertial Reference System

A 2-3-2 rotation sequence is adopted to transform the geocentric inertial axes $\mathbf{S}_i$ into the orbit-plane axes $\mathbf{O}_i$ (Table 3.1). To change the directions of the orbit-plane axes 2 and 3 following their definitions, another transformation matrix is applied. As mentioned in the last section, $\bar{\xi}_3 = \xi_3 + \tau$ is valid for elliptical orbits, while $\bar{\xi}_3 = \xi_3$ is taken in the case of a circular orbit. Hence, the final matrix is given as

$$\mathbf{T}_{OS} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} cs\bar{\xi}_3 & 0 & -sn\bar{\xi}_3 \\ 0 & 1 & 0 \\ sn\bar{\xi}_3 & 0 & cs\bar{\xi}_3 \end{bmatrix} \begin{bmatrix} cs\xi_2 & sn\xi_2 & 0 \\ -sn\xi_2 & cs\xi_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cs\xi_1 & 0 & -sn\xi_1 \\ 0 & 1 & 0 \\ sn\xi_1 & 0 & cs\xi_1 \end{bmatrix}$$
$$= \begin{bmatrix} cs\xi_1 cs\xi_2 cs\bar{\xi}_3 - sn\xi_1 sn\bar{\xi}_3 & sn\xi_2 cs\bar{\xi}_3 & -sn\xi_1 cs\xi_2 cs\bar{\xi}_3 - cs\xi_1 sn\bar{\xi}_3 \\ cs\xi_1 sn\xi_2 & -cs\xi_2 & -sn\xi_1 sn\xi_2 \\ -cs\xi_1 cs\xi_2 sn\bar{\xi}_3 - sn\xi_1 cs\bar{\xi}_3 & -sn\xi_2 sn\bar{\xi}_3 & sn\xi_1 cs\xi_2 sn\bar{\xi}_3 - cs\xi_1 cs\bar{\xi}_3 \end{bmatrix}, \tag{3.18a}$$

$$\mathbf{O} = \mathbf{T}_{OS}\mathbf{S}. \tag{3.18b}$$

The 2-3-2 Euler angles can be obtained from the above direction cosine matrix elements by

$$\xi_1 = \arctan[-\mathbf{T}_{OS}(2,3)/\mathbf{T}_{OS}(2,1)], \tag{3.19a}$$

$$\xi_2 = \arccos[-\mathbf{T}_{OS}(2,2)], \tag{3.19b}$$

$$\bar{\xi}_3 = \arctan[-\mathbf{T}_{OS}(3,2)/\mathbf{T}_{OS}(1,2)]. \tag{3.19c}$$

Note that Equation (3.19b) leaves a twofold ambiguity in ξ_2, corresponding to $\sin\xi_2$ being positive or negative. Once this ambiguity is solved, ξ_1 and $\bar{\xi}_3$ are determined uniquely by the signs and magnitudes of $\mathbf{T}_{OS}(1{,}2)$, $\mathbf{T}_{OS}(2{,}1)$, $\mathbf{T}_{OS}(2{,}3)$, and $\mathbf{T}_{OS}(3{,}2)$, with the exception that when ξ_2 is a multiple of 180°, only the sum or difference of ξ_1 and $\bar{\xi}_3$ is determined [Ref. 2].

The angular velocities of the orbit-plane system referred to geocentric inertial axes are determined from projections of the angular velocity vector $\mathbf{d\xi}/\mathrm{d}t$ along the orbit-plane axes $\mathbf{O}_i$. The result is [Ref. 7]

$$\boldsymbol{\omega}_{O,S} = \begin{bmatrix} \dot{\xi}_1 \sin\xi_2 \cos\bar{\xi}_3 - \dot{\xi}_2 \sin\bar{\xi}_3 \\ \dot{\xi}_1 \cos\xi_2 + \dot{\bar{\xi}}_3 \\ \dot{\xi}_1 \sin\xi_2 \sin\bar{\xi}_3 + \dot{\xi}_2 \cos\bar{\xi}_3 \end{bmatrix}. \tag{3.20}$$

Further transformations are necessary to refer the angular velocities of the spacecraft to geocentric inertial axes, because the motion of the satellite in its orbit, described by the angular velocities of the orbit-plane axes, and the body axes rotations of the satellite are superimposed. Therefore, the angular velocity vector of the orbit-plane axes referred to the orbit-plane axes is

$$\boldsymbol{\omega}_R = \mathbf{T}_{OS}\boldsymbol{\omega}_{O,S}, \tag{3.21}$$

and along the body axes

$$\boldsymbol{\omega}_{R,B_i} = \mathbf{T}_{BO}\boldsymbol{\omega}_R. \tag{3.22}$$

The desired angular velocities of the spacecraft referred to geocentric inertial axes along body axes are given by the following fundamental kinematical relationship

$$\boldsymbol{\omega} = \boldsymbol{\omega}_{B_i} = \boldsymbol{\omega}_{BR,B_i} + \boldsymbol{\omega}_{R,B_i}. \tag{3.23}$$

4. Observer-fixed Reference System

For conversion from the geocentric inertial axes to those in the topocentric horizon system, H_i, a 2-1 rotation sequence according to Table 3.1 is applied.

The first rotation angle α_0 is between the observer's meridian and the line of equinoxes. The relationships to determine this angle, which involve time considerations to a large extent, are given in Appendix A[8]. Using only this rotation matrix, the result is a transformation to the ECEF coordinate system. The geodetic (geographic) latitude of the observer φ_0 is required for the last rotation. Furthermore, the directions of the axis need to be changed to meet the definition of axis directions for the topocentric horizon system. The complete transformation may be stated as follows

$$\mathbf{T}_{HS} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & sn\varphi_0 & cs\varphi_0 \\ 0 & -cs\varphi_0 & sn\varphi_0 \end{bmatrix} \begin{bmatrix} cs\alpha_0 & 0 & -sn\alpha_0 \\ 0 & 1 & 0 \\ sn\alpha_0 & 0 & cs\alpha_0 \end{bmatrix} = \begin{bmatrix} sn\varphi_0 sn\alpha_0 & -cs\varphi_0 & sn\varphi_0 cs\alpha_0 \\ cs\alpha_0 & 0 & -sn\alpha_0 \\ cs\varphi_0 sn\alpha_0 & sn\varphi_0 & cs\varphi_0 cs\alpha_0 \end{bmatrix}, \tag{3.24a}$$

$$\mathbf{H} = \mathbf{T}_{HS}\mathbf{S}. \tag{3.24b}$$

D. SOLAR VECTOR

The determination of the solar vector, frequently called Sun position vector, and the beta angle, which is the angle between the solar vector and the orbital plane, is the objective in this section. It is required later because the attitude determination of PANSAT relies on the position of the Sun with respect to the spacecraft.

The geometry of the solar vector is shown in Figure 3.7. Its solution rests on determining the ecliptic longitude (the ecliptic latitude is 0°) and the range.

[8] In this appendix, the rotation angle α_0 is renamed to θ_{LST} following the common name in sidereal time considerations.

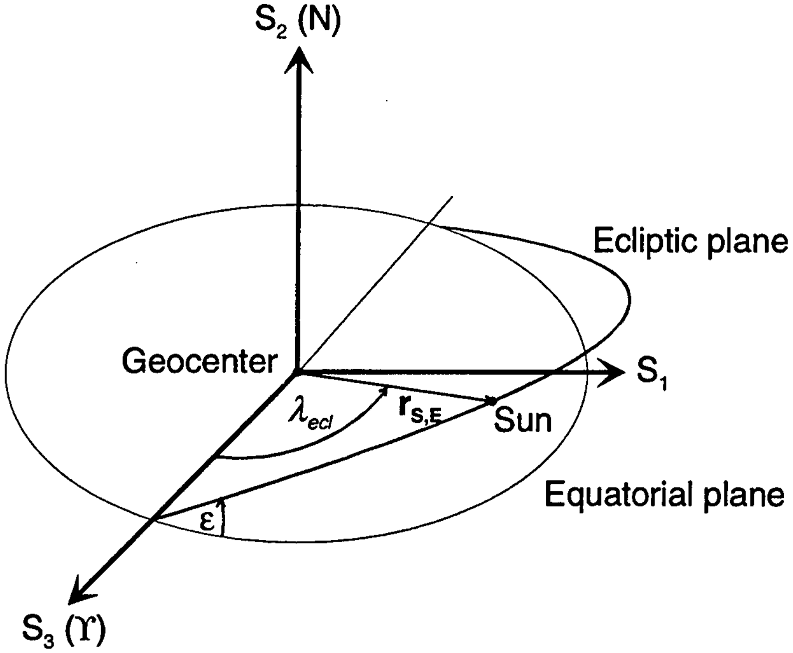


Figure 3.7: Sun-Earth Geometry for the Solar Vector

At first, the longitude of the Sun can be calculated using the following formula [Ref. 3]

$$\lambda_{S,M} = 280.4606° + 36{,}000.7701\ T_{UT1}, \tag{3.25}$$

where T_{UT1} is the number of Julian centuries from the epoch J2000 (using Equation (A.3) in the appendix [9]). The mean anomaly is computed by [Ref. 3]

$$M_S = 357.5277° + 35{,}999.0503\, T_{UT1}. \tag{3.26}$$

After reducing both $\lambda_{S,M}$ and M_S to the range of 0° to 360°, the ecliptic longitude of the Sun is given by the following equation [Ref. 3]

$$\lambda_{ecl} = \lambda_{S,M} + 1.9147° \sin(M_S) + 0.0200° \sin(2\, M_S). \tag{3.27}$$

Finally, the solar vector in question referred to the geocentric inertial axes is determined to be

$$\mathbf{r}_{S,E} = \sigma_{S,E} \begin{bmatrix} \sin\lambda_{ecl} \cos\varepsilon \\ \sin\lambda_{ecl} \sin\varepsilon \\ \cos\lambda_{ecl} \end{bmatrix}, \tag{3.28}$$

[9] To be precise, the JD of the particular date and time should be used instead of JD_0. Since the Earth moves slowly in its orbit, the accuracy would not suffer for the purposes in this report.

where $\sigma_{S,E}$ is the distance from the Earth to the Sun and can be approximated by 1 AU, and ε is the obliquity of the ecliptic which can be assumed as time independent.

Moreover, the beta angle, which is the angle between the orbit plane and the line to the Sun, can be calculated as follows [Ref. 13]

$$\sin\beta = \sin\varepsilon\cos\xi_2\sin\lambda_{ecl} - \cos\varepsilon\sin\xi_2\cos\xi_1\sin\lambda_{ecl} + \sin\xi_2\sin\xi_1\cos\lambda_{ecl}\,. \quad (3.29)$$

[This page is intentionally left blank.]

IV. INITIAL CONDITIONS AT DEPLOYMENT

The determination of the initial attitude and roll-rate of PANSAT at the time of deployment is provided in this chapter. Because PANSAT uses only one celestial body for attitude determination, the initial attitude is the only absolute attitude, which can be used as a reference. Furthermore, it is observed that the ejection mechanism introduced a spin about the axis aligned with the separation vector.

A. DETERMINATION OF ATTITUDE

The attitude data provided by Johnson Space Center, JSC, is examined in this section. NASA defines its reference systems different from those introduced in Chapter III. Therefore, a brief introduction is given.

The ECI frame, called M50 (mean equinox of 1950), uses the same origin, the center of the Earth, the same fundamental plane, the equator, and the same axis directions like the geocentric inertial system; however, the naming of the axes is different. The x axis is pointing towards the vernal equinox, the z axis is aligned with the Earth's rotation axis and y completes the right-handed system (compare to Figure 3.1).

The LVLH coordinate system follows the definition given in the last chapter. The x axis (roll) points to the forward direction, y (pitch) is the negative orbit normal, and z (yaw) points to nadir. The origin of the frame is the 'center of mass' of the orbiter (Figure 3.2).

The body axes of the orbiter shown in Figure 4.1 are defined as follows: the roll axis x is parallel to the centerline of the cargo bay and points toward the nose; the yaw axis z is parallel to the orbiter plane of symmetry and is perpendicular to x, it points positive down with respect to the orbiter fuselage. Finally, the y axis (pitch) completes a right-handed system and points toward starboard of the orbiter.

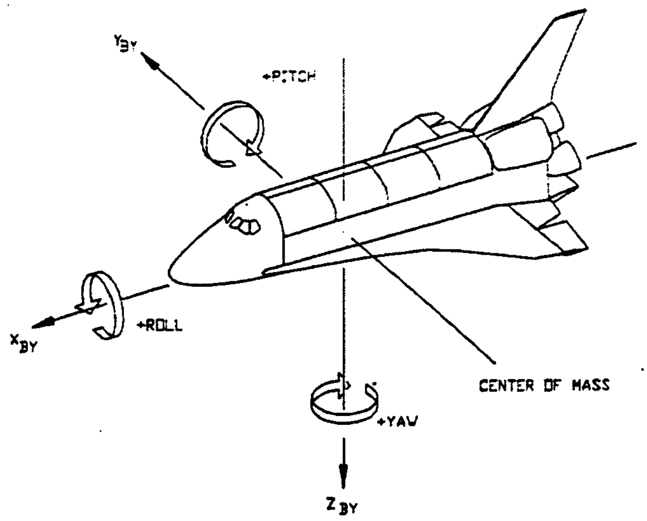


Figure 4.1: Body Axes of the Orbiter [Ref. 16]

To transform from one reference frame to another, PYR sequences are used (according to Table 3.1: 2-3-1). This is true for transformations from M50 to LVLH and from LVLH to body. Thus, the overall rotation matrix is

$$\mathbf{T}_{PYR} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cs\alpha_R & sn\alpha_R \\ 0 & -sn\alpha_R & cs\alpha_R \end{bmatrix} \begin{bmatrix} cs\alpha_Y & sn\alpha_Y & 0 \\ -sn\alpha_Y & cs\alpha_Y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cs\alpha_P & 0 & -sn\alpha_P \\ 0 & 1 & 0 \\ sn\alpha_P & 0 & cs\alpha_P \end{bmatrix}$$

$$= \begin{bmatrix} cs\alpha_P cs\alpha_Y & sn\alpha_Y & -sn\alpha_P cs\alpha_Y \\ -cs\alpha_R cs\alpha_P sn\alpha_Y + sn\alpha_R sn\alpha_P & cs\alpha_R cs\alpha_Y & cs\alpha_R sn\alpha_P sn\alpha_Y + sn\alpha_R cs\alpha_P \\ sn\alpha_R cs\alpha_P sn\alpha_Y + cs\alpha_R sn\alpha_P & -sn\alpha_R cs\alpha_Y & -sn\alpha_R sn\alpha_P sn\alpha_Y + cs\alpha_R cs\alpha_P \end{bmatrix} \tag{4.1}$$

where α_R is the roll angle, α_P is the pitch angle, and α_Y the yaw angle.

PANSAT was deployed in space on October 30, 1998, approximately 23 and a half hours after lift-off of the space shuttle at Cape Canaveral, FL. The classical orbital elements at the time of deployment, GMT 303/18:45:33, MET 00/23:25:59[10], are as follows

[10] GMT – Greenwich Mean Time, day of year/hh:mm:ss (see Appendix A for more information); MET – Mission Elapsed Time, day/hh:mm:ss.

$$\begin{aligned}\text{semi-major axis}&: a = 6941.5\,\text{km},\\ \text{eccentricity}&: e = 0.0016,\\ \text{inclination}&: \xi_2 = 28.60°,\\ \text{argument of perigee}&: \xi_3 = 11.67°,\\ \text{RAAN}&: \xi_1 = 150.83°,\\ \text{true anomaly}&: \tau = 320.60°,\\ \text{mean anomaly}&: M = 320.72°.\end{aligned}$$

The rotation angles for the M50 to LVLH transformation at deployment are given as

$$\begin{aligned}\alpha_R &= 280.46°,\\ \alpha_P &= 151.33°,\\ \alpha_Y &= 56.39°,\end{aligned}$$

and for the LVLH to orbiter body system transformation

$$\begin{aligned}\alpha_R &= 141.6°,\\ \alpha_P &= 96.1°,\\ \alpha_Y &= 359.9°.\end{aligned}$$

For convenience, the attitude data given above is transformed into the variables of the coordinate systems used in this report.

At first, allow a transformation of the B1950 epoch to the J2000 standard epoch by using the following transformation matrices

$$\mathbf{T}_{OS,J2000} = \mathbf{T}_{PYR}\,\mathbf{T}_{J2000,B1950}{}^{T}\,\mathbf{T}_{M50,S}\,, \tag{4.2}$$

where the $\mathbf{T}_{PYR}$ matrix is given with Equation (4.1) and using the rotation angles for the M50 to LVLH transformation; $\mathbf{T}_{J2000,B1950}$ can be found in Appendix B; and $\mathbf{T}_{M50,S}$ must be applied in order to use the geocentric inertial coordinates instead of the M50 coordinates. This matrix is determined to be

$$\mathbf{T}_{M50,S} = \begin{bmatrix} 0 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}. \tag{4.3}$$

Applying Equation (3.19) on the direction cosine matrix $\mathbf{T}_{OS,J2000}$ and inspecting the quadrants of the sine and cosine functions of ξ_1 and $\bar{\xi}_3$, the following result can be reached

$$\begin{aligned}\xi_1 &= 180° - 28.09° = 151.91°,\\ \xi_2 &= 28.46°,\\ \bar{\xi}_3 &= 360° - 28.28° = 331.72°.\end{aligned}$$

These are the desired angles of the classical orbital elements referred to the J2000 standard epoch.

Furthermore, the ζ angles are determined for the transformation of the orbit-plane axes to the body axes of the orbiter. Equation (3.16) is applied on the direction cosine matrix $\mathbf{T}_{PYR}$ that uses the Euler angles for the LVLH to body transformation given above. Observation of the quadrants of the sine and cosine functions yield

$$\begin{aligned}\zeta_1 &= 360° - 37.5° = 322.5°,\\ \zeta_2 &= 84.1°,\\ \zeta_3 &= 180° + 0.9° = 180.9°.\end{aligned}$$

To resolve the attitude of PANSAT with respect to the orbit-plane coordinate system, its orientation in the orbiter needs to be known. PANSAT was integrated into the canister in the payload bay so that the axis of symmetry Y' was aligned with the orbiter's z axis, but pointing into the negative direction; the Z' axis pointed towards the tail and, therefore in the negative direction of the orbiter's x axis; and last but not least, the PANSAT's X' axis was directed to the starboard side. Therefore, the transformation matrix, which takes the body axes of the orbiter to the geometrical axes of PANSAT, is obtained to

$$\mathbf{T}_{X',x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}. \tag{4.4}$$

The Euler angles of the orbit-plane to geometrical axes transformation are determined from the following direction cosine matrix

$$\mathbf{T}_{X'O} = \mathbf{T}_{X'x}\,\mathbf{T}_{PYR}, \tag{4.5}$$

where $\mathbf{T}_{PYR}$ uses the angles for the LVLH to orbiter body transformation.

Applying Equation (3.16) on $\mathbf{T}_{X'O}$ and examining the quadrants of the trigonometric functions as usual, the desired ζ angles are determined to be

$$\zeta_1 = 360° - 4.7° = 355.3°,$$
$$\zeta_2 = 3.6°,$$
$$\zeta_3 = 360° - 51.8° = 308.2°.$$

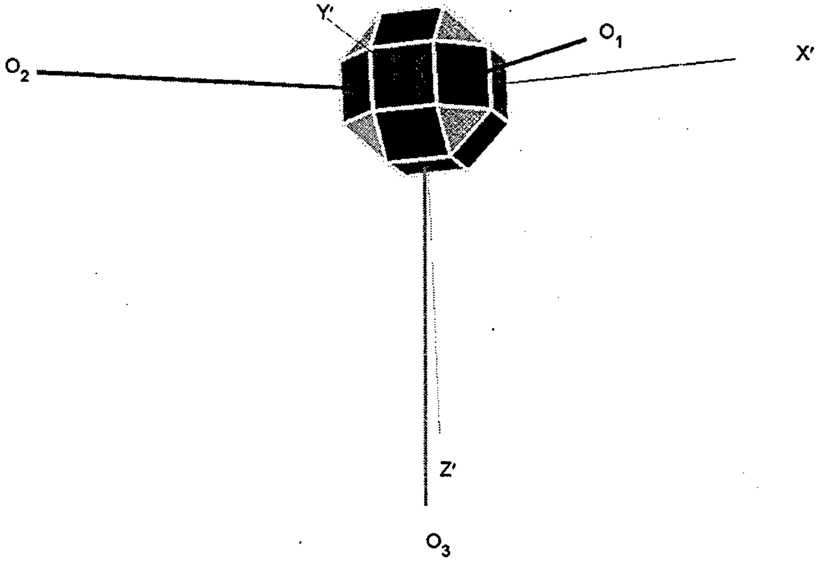


Figure 4.2: Initial Attitude of PANSAT with respect to Orbit-plane Axes[11]

These Euler angles give the attitude of PANSAT at deployment with respect to the orbit-plane frame, as shown in Figure 4.2. At this time, the symmetric axis of PANSAT is pointing towards the Sun in order to obtain the maximum energy for the solar panels. This knowledge can be used to verify the attitude data. Hence, the following sequence transforms the Y' axis of PANSAT to the geocentric inertial reference system

$$\mathbf{Y'}_S = \mathbf{T}_{OS}{}^{T}\,\mathbf{T}_{BO}{}^{T}\,\mathbf{Y'}, \tag{4.6}$$

where the vector $\mathbf{Y'}$ is $\begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$, $\mathbf{T}_{BO}$ is given by Equation (3.15) and, using the rotation angles for the orbit-plane to geometrical axes transformation as shown above, and $\mathbf{T}_{OS}$ given by Equation (3.18) uses the J2000 classical orbital element angles as determined. This yields the result

$$\mathbf{Y'}_S = \begin{bmatrix}-0.555\\-0.237\\-0.797\end{bmatrix}.$$

Comparing this vector to the solar vector at the time of deployment, Equation (3.28) is resolved and the solar vector normalized so that this result can be reached

$$\mathbf{r}_{S,E_n} = \begin{bmatrix}-0.554\\-0.240\\-0.797\end{bmatrix}.$$

It shows that the axis of symmetry of the satellite, Y', was indeed coaligned with the vector to the Sun as PANSAT was deployed into its orbit.

B. INITIAL STATE OF MOTION AT DEPLOYMENT

In this section, the deployment of PANSAT is examined concerning the initial state of motion at the time of separation. This is done by observing photo and video material provided by NASA.

PANSAT was launched into low-Earth orbit via the Shuttle under the Hitchhiker program utilizing a Get Away Special (GAS) canister. The canister was located at the IEH-3 bridge on the port side of the centerline of the cargo bay (the configuration is shown in Figure E.3). A NASA standard Ejection Mechanism for GAS payloads was used to deploy the spacecraft.

There are two high-resolution photographs, Figures E.1 and E.2 (Appendix E), available with a Δt of 2 s showing PANSAT as it separates from the orbiter. They were

[11] Panels equipped with current sensors are shown in magenta color, while dark blue is used for the others.

taken through the window of the crew compartments located at the port side of the cargo bay. Obviously, a spin about the geometrical axis of symmetry is introduced by the spring of the Ejection Mechanism. In order to investigate that, the location of the observer, who took the pictures, with respect to the symmetric axis is determined. By knowing the dimensions of the satellite and measuring them on the pictures, the angles between the vector to the observer and the normal vector of the panel facing the viewer, SP11, can be determined. Referring the difference in the angles to the difference in time between the pictures, the angular velocity is obtained.

In order to follow this procedure, the scale factors of the pictures need to be determined. This can be done by observing lengths on the pictures, l_m, and referring them to the true value, l_t, as follows

$$f_S = l_t / l_m . \tag{4.7}$$

Since the spacecraft is at the same height as the observer, the scale factor for the first picture is determined by using the distance from the baseplate of the launch vehicle interface to the top panel. This results in a scale factor of $f_{S,I} = 48.6\,\mathrm{cm}/7.7\,\mathrm{cm} = 6.3$[12]. For the second picture, Figure E.2, the diameter at the bottom of the LVI is taken, because PANSAT has already gained height relative to the observer. This yields a scale factor of $f_{S,II} = 24.1\,\mathrm{cm}/3.6\,\mathrm{cm} = 6.7$.

The angle between the plane of the picture, to which the vector to the observer is perpendicular, and the visible side panels, is determined by applying a sine or cosine function on the measured horizontal length of the panel (multiplied by the scale factor) and the true distance (Figure 4.3). This is done for different panels to obtain a more reliable result. For SP10 and SP13, the equation is

$$\delta_i = \arcsin\left(l_{h,m_i} / l_{h,t_i}\right), \tag{4.8}$$

[12] - The first picture taken at t_1, Figure E.1, is denoted by the index I, while the second picture taken at t_2, Figure E.2, is denoted by the index II.
- Note that the actual measured length depends on the print out.

where δ is the desired angle, $l_{h,m}$ is the measured horizontal length of side panel, $l_{h,t}$ is its true length (= 18.1 cm), and i is an index for the panel. In the case of SP12, the required angle is determined using a cosine function

$$\delta_{12} = \arccos\left(l_{h,m_{12}} / l_{h,t_{12}}\right). \tag{4.9}$$

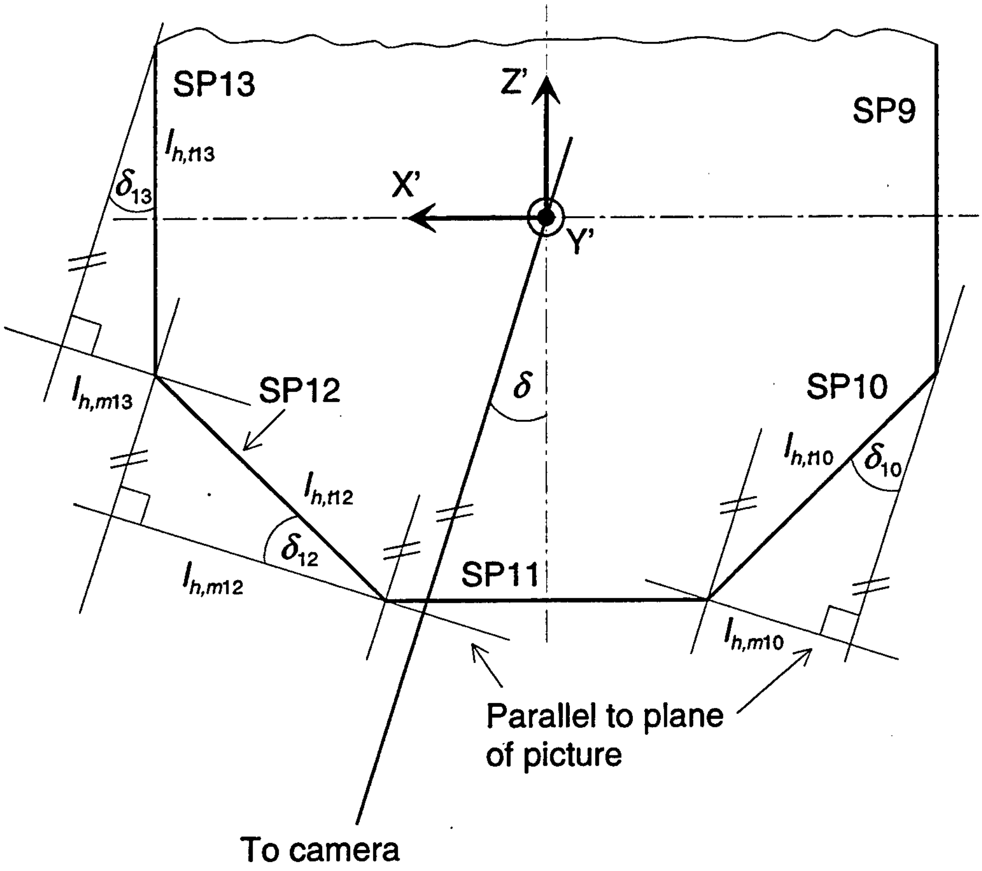


Figure 4.3: Geometry Showing a Top View of PANSAT to Determine the Initial Spin about Y'[13]

Applying Equation (4.8) for SP13, an angle of $\delta_{13,I} = \arcsin(0.50 \cdot 6.3/18.1) = 10.0°$ is obtained for the first picture and $\delta_{13,II} = \arcsin(0.90 \cdot 6.7/18.1) = 19.5°$ for the second one. Because the plane of this solar panel is parallel to the normal vector of SP11, it is also the angle between the observer and this normal vector (or the Z' axis). The difference in the angles is calculated

[13] The lines shown parallel to the camera vector can be assumed parallel, because the camera distance is long compared to the dimensions of the spacecraft (16 m >> Ø 44 cm).

as $\Delta\delta_{13} = 9.5°$. In the case of SP10, the first angle is $\delta_{10,I} = \arcsin(1.65 \cdot 6.3/18.1) = 35.1°$ and the angle two seconds later is $\delta_{10,II} = \arcsin(1.15 \cdot 6.7/18.1) = 25.2°$, which leads to a delta angle of $\Delta\delta_{10} = 9.9°$.

From Equation (4.9), the angles for SP12 are determined to be $\delta_{12,I} = \arccos(2.20 \cdot 6.3/18.1) = 40.0°$ and $\delta_{12,II} = \arccos(2.50 \cdot 6.7/18.1) = 29.5°$, which gives a difference of $\Delta\delta_{12} = 10.5°$.

The mean value of the these three delta angles is

$$\Delta\overline{\delta} = \frac{\Delta\delta_{13} + \Delta\delta_{12} + \Delta\delta_0}{3} = 10.0°. \tag{4.10}$$

Referring this value to the sampling period of the pictures and observing the direction of the spin, the angular velocity about the axis of geometry at the time of deployment is determined to be

$$\omega_{BR,Y'} = 10.0°/2\,\text{s} = 5\,°/\text{s} = 0.83\,\text{rpm}.$$

This angular velocity is a good estimate but contains uncertainties due to the measurements of the lengths in the pictures. Furthermore, another error is added because the time the pictures were taken is only known in seconds and a higher precision is not available.

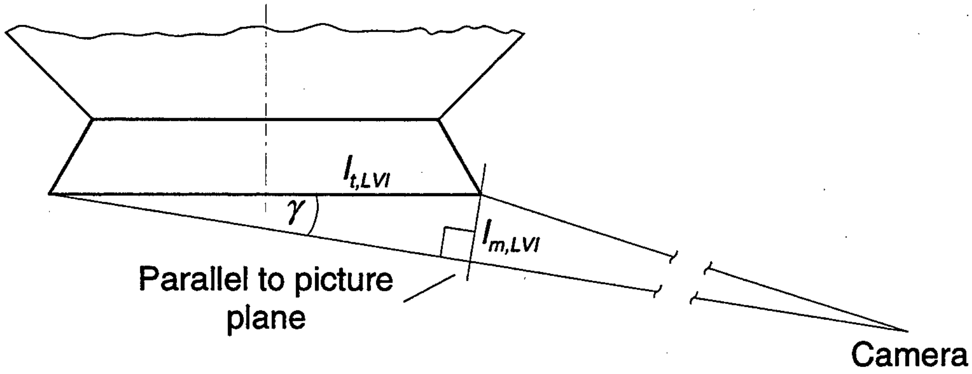


Figure 4.4: Determination of Angular Separation to Plane of Camera

To get more confidence in this result, the ejection velocity is determined from the pictures and compared to the value specified by NASA. The first picture, Figure E.1, shows that PANSAT is at the same height as the camera, while the second picture, Figure E.2, shows that it has gained a certain additional height. It can be determined by observing the LVI, which is shown in Figure 4.4. The angle $\gamma$is

$$\gamma = \arcsin\left(\frac{l_{m,LVI}}{l_{t,LVI}}\right) = \arcsin\left(\frac{0.55 \cdot 6.7}{24.1}\right) = 8.8^\circ. \tag{4.11}$$

Considering the distance of the canister from the end of the cargo bay, where the windows and the observer are located[14], which is given as 15.8 m, the vertical distance PANSAT traveled in two seconds is

$$d = 15.8\,\mathrm{m} \cdot \tan(8.8^\circ) = 2.45\,\mathrm{m},$$

corresponding to an ejection velocity of

$$v = \frac{2.45\,\mathrm{m}}{2\,\mathrm{s}} = 1.22\,\mathrm{m/s} \hat{=} 3.73\,\mathrm{ft/s}.$$

Referring to the technical specifications of the ejection mechanism, it is supposed to provide a nominal ejection velocity of 3.7 ft/s, and a worst-case delta v of 3.3 ft/s should one of the springs fail. Because this did not happen, the relative error of the estimated ejection velocity is about one percent and, therefore, negligible.

Because the camera is not fixed to the orbiter, it is subject to change. Observing the distance of the tail marking the centerline of the orbiter and the symmetric axis of PANSAT, it shows that the camera moved slightly to the port side. This is approximately 4 cm in 2 s. Since the observer is about 16 m away from the spacecraft, this change in the camera's position can be neglected.

Besides the angular spin introduced by the Ejection Mechanism, the orbiter and, thus, PANSAT were rotating as well. The angular velocities of the orbiter with respect to the inertial M50 frame are given as

$$\begin{aligned} \omega_R &= -0.001\,^\circ/\mathrm{s}, \\ \omega_P &= 0.001\,^\circ/\mathrm{s}, \\ \omega_Y &= -0.001\,^\circ/\mathrm{s}. \end{aligned}$$

Even referring these angular velocities to the orbit-plane coordinate system, which superimposes them with the orbit motion, leads only to a slight increase. Hence,

[14] It is supposed that the camera is directly behind the window in the crew compartment of the orbiter.

these angular velocities are negligible for determining the initial roll-rate. However, it should be kept in mind that over long periods of time small angular rates can become noticeable due to torques acting on the spacecraft. Furthermore, it cannot be verified on the pictures if there was a tip-off moment at deployment, which would lead to additional energy. This can be done by examining the telemetry data.

[This page is intentionally left blank.]

V. DYNAMICS OBSERVED FROM SOLAR PANEL DATA

The outer surface of PANSAT is almost fully covered with eighteen solar panels. Seventeen square panels are equipped with silicon solar cells and one gallium-arsenide panel is attached to the LVI at the bottom of the spacecraft. Current sensors to monitor the output current of the panels were designed into eight of them. Furthermore, temperature measurements from all solar panels and the internal components are available.

In this chapter, this data is evaluated in order to observe the spacecraft's dynamics. The solar panel currents give the location of the Sun with respect to the spacecraft. They are examined and used for the determination of the rotational motion and the attitude of the spacecraft.

A. EVALUATION OF STATUS SNAPSHOT DATA

As discussed in Chapter IV, a spin about the minor axis was introduced at deployment. The telemetry data, which is stored in the satellite and downloaded to the ground station, is subject to aliasing. This is because the sampling period of the telemetry data is given as $T_S = 121$ s, while the angular velocity about the axis in question is approximately 5 °/s leading to a period for a complete rotation of $T = 72$ s. The sampling theorem of Shannon states that the sampling frequency must be at least twice as high as the highest frequency in a signal to avoid aliasing. The sampling period T_S and the sampling frequency f_S are linked by the following formula

$$T_S = \frac{1}{f_S} = \frac{1}{2 f_N}, \tag{5.1}$$

where f_N is the Nyquist frequency in Hz, the highest frequency in a signal. Using the above rotation period based on the initial spin, the sampling period needed is determined to be $T_S = 36$ s instead of 121 s. Therefore, the present sampling period generates undersampled telemetry records and cannot be used to observe the rotational motion of the spacecraft.

To accomplish the determination of the roll-rate, the method of using status records is adopted. When the spacecraft is in sight of the ground station, status information at that instant time (also called status "snapshots") can be received on demand. It contains almost the same information as the telemetry records. Taking status records repeatedly during a pass of the satellite, i.e., the spacecraft is in view of the ground station, a complete set of status information for a time window of a few minutes can be obtained. Since PANSAT is a tumbling spacecraft with antennas having a non-uniform radiation pattern, the transmission quality depends strongly on the attitude. This leads to the effect that the status samples are not equidistant as a result of the roll-rate and the attitude with respect to the ground station. The following analysis is based on these status samples.

1. Solar Panel Currents

As a first step, the elementary roll-rate of PANSAT can be observed by looking at the solar panel current curves. The solar panels, which are equipped with current sensors, are as follows: 4, 5, 7, 9, 11, 13, 14, 16 (Figure 1.1).

Figures 5.1 and 5.2 show that the spacecraft is spinning about the geometrical axis of symmetry, though this axis is not directed to the Sun. This is because the cover panel currents – the cover (CV) panels are located at the side of the spacecraft – change in sequence. First, a peak occurs on panel SP7, afterwards SP9, then SP11, and then SP13 and, subsequently, the cycle is repeated. The phase shift from panel to panel can be best seen in Figure 5.1. When the phase shift in the CV panel currents occurring from panel to panel is suppressed, the correspondence in the curves can be seen and understood especially in areas with fewer sample points (Figure 5.3). It should be noted that the currents of the adjacent panels of SP4 and SP13 as well as SP9 and SP14 are in phase showing consistency of the curves. Since the spacecraft is rotating and the vector to the Sun can be assumed to be fixed in space for that short period of time, it is a positive spin about Y'. Observation of Figure 5.2 reveals a period of the spin of about 60 s and, hence, an angular velocity of $360°/60\,\mathrm{s} = 6\,°/\mathrm{s} \mathrel{\hat{=}} 2/3\,\mathrm{rpm}$.

This spin axis is not pointing to the Sun, because the amplitude of the top (TP) panel current is lower than the amplitude of the peaks of the CV panel currents (in such a case, the side panel currents are zero). The angle between the symmetrical axis and the

Sun line can be estimated by comparing the amplitude of the currents of SP9 and SP14. Both panels are adjacent to each other similar to SP4 and SP13 (except that SP4 is on the lower deck of the spacecraft); however, the normal vector of panel SP9 is perpendicular to the axis of symmetry, while the angle between the normal of SP14 and Y' is 45°. Observing the first three samples (Figure 5.2) reveals that the amplitudes of the peak values of I_9 and I_{14} are approximately the same. This is also true for the Sun incidence angle being the angle between the normal to the panel's surface and the solar vector, because its cosine is proportional to the output current of the panel. Therefore, the Sun incidence angle of SP14 is about 22.5° giving an angle of the symmetrical axis and the Sun line being the line from the spacecraft to the Sun of about 67°.

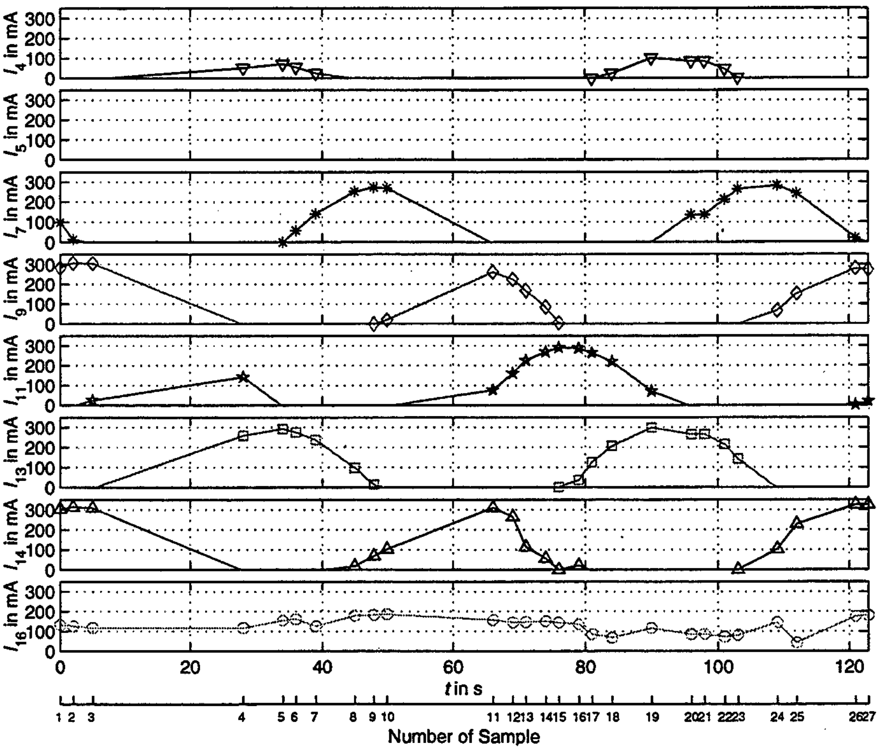


Figure 5.1: Solar Panel Currents (1998-11-17_06-29-17)[15]

Figure 5.2 shows that the axis of symmetry of the spacecraft has shifted with respect to the Sun line. This is indicated by observation of the currents of SP9 and SP14

[15] The format of the date/time stamp is as follows: yyyy-mm-dd_hh-mm-ss; Pacific Standard Time, PST, is used (*UTC* – 8 hours). It belongs to the first sample taken.

at $t \approx 65\,\text{s}$ and $t \approx 120\,\text{s}$, because the Sun incidence angle of SP14 is increased and the one of SP9 decreased. The amplitude of the top panel current for these samples is increased as well compared to samples 2 and 3. Therefore, the angle between the Sun line and the symmetrical axis is lowered on these observed sample points. This angle is increased at $t \approx 35\,\text{s}$ and $t \approx 95\,\text{s}$ indicated by the current curves of SP13 and SP16. The peak currents of the lower deck panel SP4, which is in phase with SP13, may prove that.

The change in this angle mentioned above indicates that the spin is obviously not the only rotational motion of the spacecraft. Since the deviation of the principal axes from the geometrical axes is negligible as a first-order approximation (see Appendix C), nutation occurs implying that there are angular velocity components in the other two axes[16]. Using Figure 5.2, the period of the top panel current is determined to be approximately 80 s, corresponding to an angular velocity of 4.5 °/s. This is the period of the nutation, because the top panel current relative to the CV panel currents provide the location of the minor axis, which is changing with respect to the Sun. Hence, in considering the snapshot data time window, the rotational motion of the spacecraft consists of two major parts: the spin about Y' and the nutation[17].

By investigating the current curves, some random errors seem to be introduced. The top panel current curve does not have a sinusoidal shape. The current curves should be sinusoids because the satellite rotates smoothly in space and following the Law of Cosines, the current of a panel changes as function of the cosine of the Sun incidence angle. The sudden drops in amplitude of one or a few samples are the result of shadowing of solar cells by the four dipole antennas. They are mounted on the triangular sections of the upper deck (see Figure 1.1). Because the angle between the symmetrical

[16] In satellite dynamics, nutation is defined as rotational motion for which the instantaneous rotation axis is not aligned with a principal axis. In the case for which a geometrical axis is not parallel to a principal axis, this misalignment causes the geometrical axis to rotate about the angular momentum vector. This rotation is referred to as coning [Ref. 2].

[17] Besides these mentioned rotations, other rotational motions are caused by the satellite's movement in its orbit and the Earth, which turns around the Sun. In the considered time frame of about 2 minutes, these effects can be neglected.

axis and the Sun line is about 65°, the antennas shade some areas of the top panel (Figures F.1 and F.2 in the appendix). It happens only when a triangle is facing the Sun, i.e., between two peaks of the side panel current. The effect of shadowing is well understood. If one portion of one solar cell is shadowed, the output current is reduced. Since all 32 cells making up an array are connected in series, the output current of all the other non-shadowed cells is limited as well. That is why the loss of area equivalent to the drop in current is usually greater than the real shadowed area by the antenna.

Another visible effect has its origin in the acquisition of the measurements. The values of the panel currents are identical at sample number 20 and 21; therefore, it appears like a systematical error (Figure 5.2). Obviously, the measurements are taken too fast, so the measurement system could not obtain the new values. All measurements[18] are sampled and converted to digital information. This process takes approximately 17.4 ms per A/D sample. Because there are 108 samples per cycle, the system needs 1.9 s to acquire all measurements[19]. This is the reason for the observed effect, because the sampling period of these samples is 2 s and thus in the range of the mentioned time lag.

Considering snapshot data of November 9, 1998 (see Appendix G for data plots), systematical errors occur in the solar panel current curves of the side panels, because there are currents reaching values up to 45 mA on panels opposite of the illuminated one. By observing the peaks of I_9, I_{14} and I_{16}, the angle between the solar vector and the symmetrical axis is between 45° and 67.5°. The pictures in Appendix F show that parts of the antennas reach slightly over the side panel planes. Furthermore, they are bent so that Sun's rays may be reflected onto the surface of a panel pointing to the opposite direction of the Sun.

[18] e.g., battery cell voltages, battery currents, spacecraft bus voltage, solar panel currents, and temperatures of panels and internal components.

[19] The spacecraft bus voltage and the solar panel current measurements, which give the power output of the panel, are taken at the end of the cycle providing a time difference between the first and last measurement of the panel currents of about 240 ms.

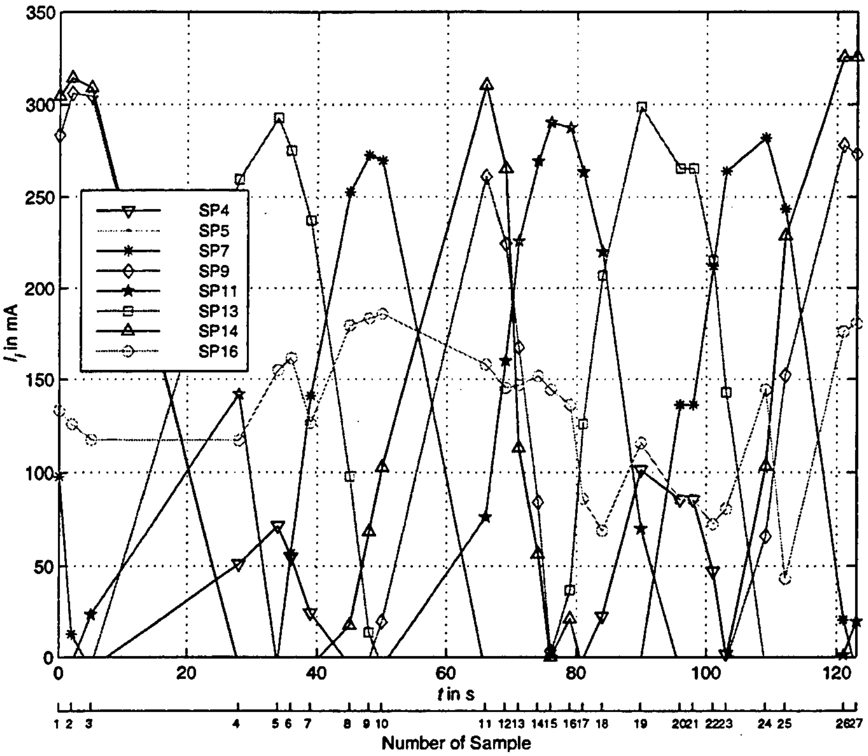


Figure 5.2: Solar Panel Currents (1998-11-17_06-29-17)

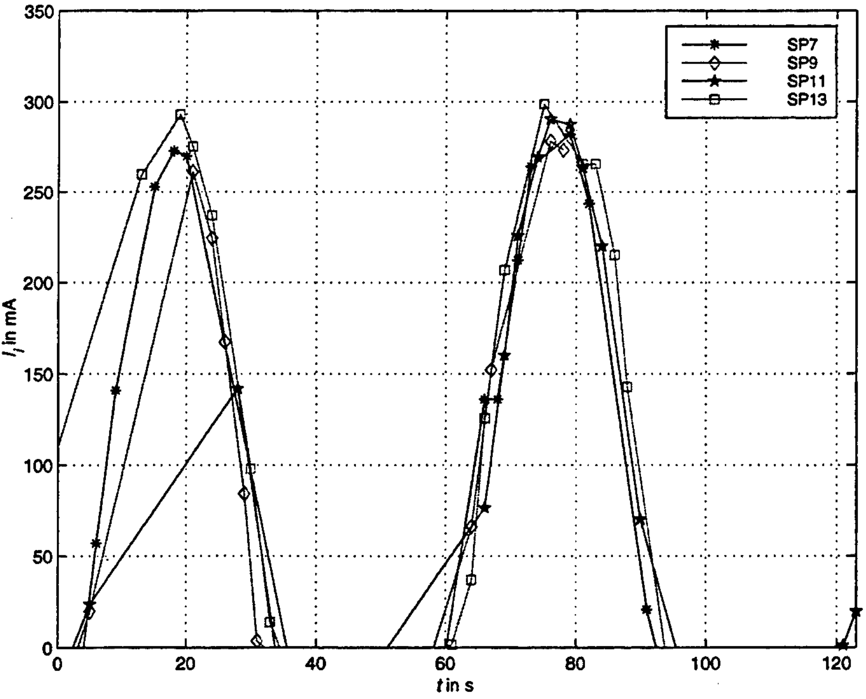


Figure 5.3: CV Panel Currents, Phase Shift Suppressed (1998-11-17_06-29-17)

Reflections caused by the LVI can also be observed on some status records (e.g., December 14, 1998; see Figure G.56 in the appendix). The sides of the LVI are under a 30° angle with respect to the axis of symmetry. Thinking of the attitude of the spacecraft with respect to the Sun, the current peaks of SP9, SP14 and SP16 reveal a Sun line/axis of symmetry angle between 22.5° and 45°. Because SP4 is attached with a 45° angle with respect to the axis of symmetry, it should not receive any illumination producing current. Observing its output current shows that current is generated and suggests that reflections, caused by the LVI, could account for this current.

2. Panel Temperatures

In this section, the solar panel temperatures are examined using data of November 17, 1998, 6:29:17 PST as an example. The panel temperatures are a function by the attitude of the spacecraft with respect to the Sun and, therefore, its rotational motion, the Sun exposure, eclipse cycles as well as conduction and radiation within the satellite.

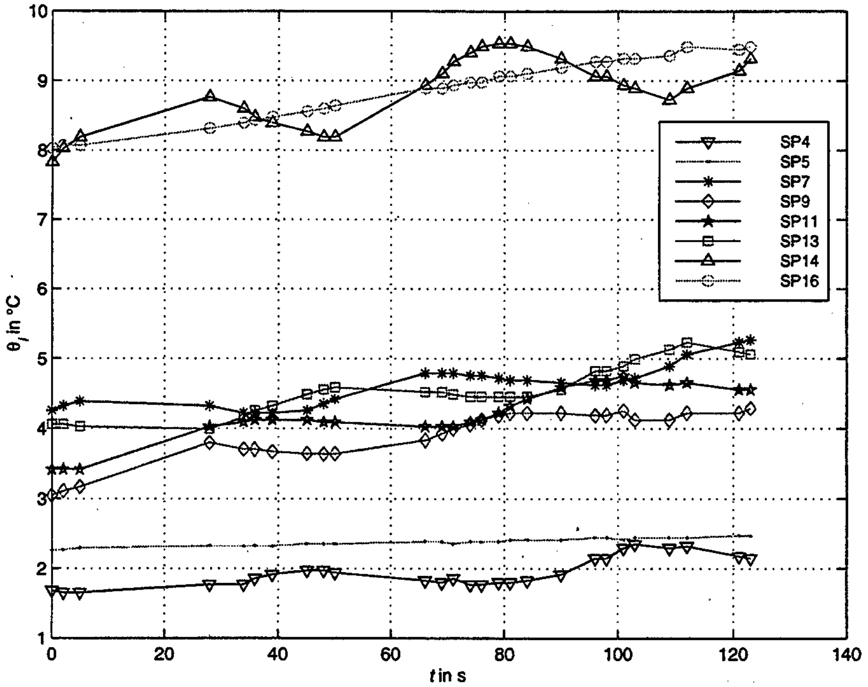


Figure 5.4: Temperatures of Panels Equipped with Current Sensors (1998-11-17_06-29-17)

Figure 5.4 depicts the solar panel temperatures of those panels that are equipped with current sensors. As estimated in the last section, the upper half of the spacecraft is directed toward the Sun. The temperatures are the highest of the panels facing the most area to the Sun. These are the upper deck panels and the top panel. The CV panel temperatures are significantly lower, because they obtain less illumination from the Sun and conduction between the panels is limited due to small cross sectional areas. The lowest temperatures occur on the gallium arsenide panel at the bottom and the lower deck (LDCK) panel SP4. Remarkably, the temperature of SP5 is higher than SP4, although the former panel does not obtain any illumination at all. This is because the bottom panel is part of the LVI block, onto which Sun's rays are shining and where higher conduction of heat takes place. The overall temperature difference between the panels is considerably small (about 9 °C), which indicates that the heat transfer within the satellite is insignificant.

At the time the status snapshots were taken, the Sunrise occurred on the ground. It indicates that the satellite left the penumbra shadow only recently. The steady rise in all temperatures is a result of the eclipse/Sun transition. It can be best seen in Figure 5.5, where the upper deck (UDCK) panel temperatures surround the increasing top panel temperature curve. The local cycles in the UDCK panel temperatures are caused by the spin of the satellite about its geometrical axis of symmetry. Observation of the period of the cycle reveals a value of roughly 60 s corresponding to the value determined from the panel currents. Hence, the thermal time constant and thermal mass are considerably low allowing the use of the temperatures to support the rotational rates determined from the current sensors.

Because the peaks of the UDCK panel temperatures are higher than the corresponding value of the top panel temperature, they are under a smaller angle directed to the Sun than the top panel. This is also proven in the last section, where the currents are examined.

The observed measurement acquisition problem is confirmed by the temperature samples showing the same effect (Figure 5.5).

On some panel temperature plots, the heat transfer from other internal components can be observed. As an example, the snapshot data of 12/14/1998 starting at 16:35:41 PST is taken (plots can be found in Appendix G). Especially the top panel temperature curve shows the influence of peak temperature of the electrical power system (EPS), the mass storage (MS), and the temperature multiplexer (TMUX), which are directly underneath SP16 and mounted to the upper equipment plate. This also suggests that there is good conduction and radiation within the satellite structure.

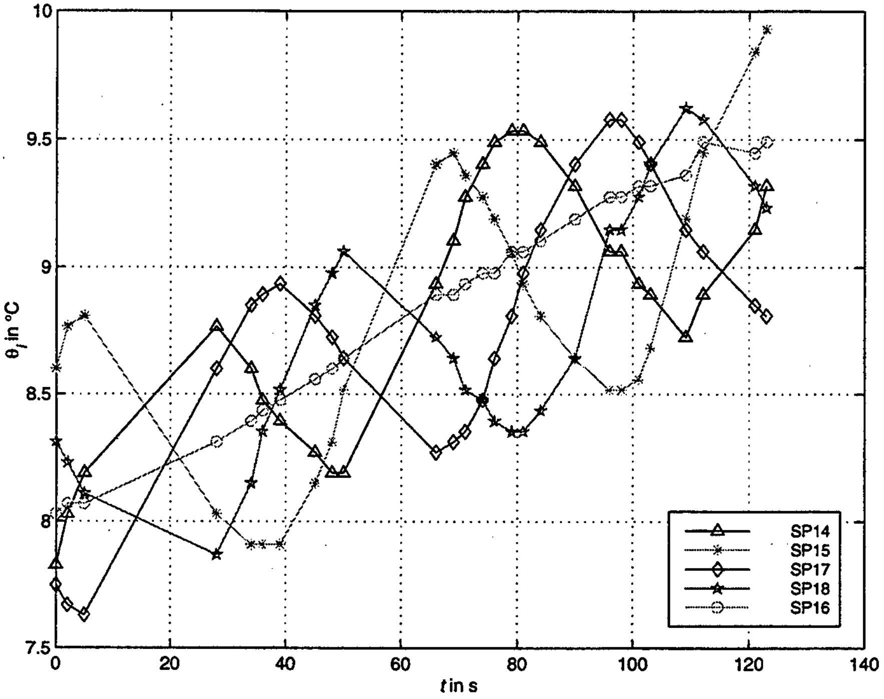


Figure 5.5: Temperatures of UDCK and TP panels (1998-11-17_06-29-17)

B. ANALYSIS OF ROTATIONAL MOTION

The rotational motion of PANSAT is very complex. As observed from the solar panel data, its motion consists of a spin and nutation. In addition, precession of the angular momentum vector occurs because external torques react on the spacecraft.

The precession can be estimated by the angular momentum equation (2.2). The time differentials are approximated by differences, so that the change in angular momentum in scalar form is

$$\Delta L = M \Delta t . \tag{5.2}$$

The gravity gradient torque is the primary torque as determined in Chapter II. Since the gravity gradient torques are smaller about the other body axes and change with the attitude of the spacecraft with respect to the 'center of gravity' of the Earth, a value of 10^{-7} Nm is used rather than the worst-case value determined. Applying Equation (5.2) gives the change in angular momentum during a day as

$$\begin{aligned}\Delta L &= 10^{-7}\,\mathrm{Nm} \cdot 86{,}400\,\mathrm{s} \\ &= 0.01\,\mathrm{Nms}\end{aligned} ,$$

over a week as

$$\begin{aligned}\Delta L &= 10^{-7}\,\mathrm{Nm} \cdot 604{,}800\,\mathrm{s} \\ &= 0.06\,\mathrm{Nms}\end{aligned} ,$$

and over a period of a month as

$$\begin{aligned}\Delta L &= 10^{-7}\,\mathrm{Nm} \cdot 2{,}592{,}000\,\mathrm{s} \\ &= 0.26\,\mathrm{Nms}\end{aligned} .$$

In order to compare these values with the angular momentum at the time of deployment, the following equation is used

$$L = \sqrt{I_1^2 \omega_1^2 + I_2^2 \omega_2^2 + I_3^2 \omega_3^2} . \tag{5.3}$$

Since the spacecraft is rotating with 5 °/s about the geometrical axis of symmetry assumed to be coaligned with the minor axis, the angular momentum is determined to be

$$\begin{aligned}L &= \sqrt{\left(0.98\,\mathrm{kg\,m^2} \cdot -0.087\,\mathrm{rad/s}\right)^2} \\ &= 0.09\,\mathrm{Nms}\end{aligned} .$$

At the time of deployment, the angular momentum vector is aligned with the symmetrical axis with a magnitude as determined above, because the spacecraft is only spinning about this axis.

If a spacecraft is initially spinning about a principal axis, as in the case of PANSAT, a torque applied parallel or antiparallel to the angular momentum vector will cause an increase or a decrease in the magnitude of **L** without affecting its direction. A torque component perpendicular to **L** will cause the direction of **L** to change without altering its magnitude. This change in direction of an angular momentum vector due to an applied torque is called precession.

Assuming that the torque in Equation (5.2) is applied perpendicular to the angular momentum vector, the angle between the initial and the precessed vector over a day is given as

$$\Delta\alpha = \arctan\left(\frac{0.01}{0.09}\right) \approx 6^\circ.$$

Therefore, precession cannot be neglected because gravity gradient torques are acting on PANSAT.

In order to analyze the rotational motion of the spacecraft, the solar vector needs to be determined. To accomplish that, curve-fitting methods are required due to errors in the current curves.

1. Curve Fitting of Panel Currents

As discussed in Section V.A., errors having random or systematical characteristics appear in the solar panel current curves. Furthermore, the samples are not taken equidistant so that curves are weakly validated in some regions. Because this data is used to compute the solar vector, which is done in the following section, the current curves need to be smoothed out.

This can be done by adopting the knowledge about the origin of these curves. They are all sinusoids because they change proportionally with the cosine of the incidence angle. Hence, it is appropriate to fit a sine function to the data given as follows

$$I_f = a_0 + b_0 \sin(\omega t + c_0), \tag{5.4}$$

where a_0, b_0, and c_0 are respectively the vertical shift, amplitude, and phase shift. The radian frequency ω is given by $2\pi/T$, where the period T is determined from the raw curve.

In order to accomplish the goal of fitting Equation (5.4) to the current curves, a weighted least squares method is used [Ref. 25, 27]. This estimation algorithm minimizes the square of the error between the measured value and the fitted curve. The discrete loss function J can be stated as follows

$$J(a_0, b_0, c_0) = \sum_{i=1}^{N} w_i \left[I_{m,i} - a_0 - b_0 \sin\left(\frac{2\pi}{T} + c_0\right)\right]^2 \rightarrow \min, \qquad (5.5)$$

where w_i is the weight for the *ith* sample. The weights are determined manually by observing the current curves, because it is difficult to model them mathematically. $I_{m,i}$ is the *ith* measured value of the current.

The desired parameters are determined by its first partial differential of the loss function, which is the necessary criterion for a minimum[20]. This leads to three equations

$$\frac{\partial J}{\partial a_0} = 0, \frac{\partial J}{\partial b_0} = 0, \frac{\partial J}{\partial c_0} = 0, \qquad (5.6)$$

which must be solved numerically, because the problem is not linear in c_0. This is programmed in MAPLE; the code is given in Appendix J.

Equation (5.4) provides satisfactory results for the estimation of the top panel current curve. Since the side panel currents are not only influenced by the spin about the minor axis but also by the nutation, above equation is not appropriate in this case. Because the current curve of SP16 is a function of the nutation, it is added to the estimation equation using another sine function

$$I_f = a_0 + b_0 \sin(\omega t + c_0) + d_0 + e_0 \sin(\omega_{16} t + f_0). \qquad (5.7)$$

Hence, the loss function is extended as follows

$$J(b_0, d_0, e_0) = \sum_{i=1}^{N} w_i \left[I_{m,i} - a_0 - b_0 \sin\left(\frac{2\pi}{T} + c_0\right) - d_0 - e_0 \sin\left(\frac{2\pi}{T_{16}} + f_0\right)\right]^2 \rightarrow \min.$$

$$(5.8)$$

The partial differential equations are given as

$$\frac{\partial J}{\partial b_0} = 0, \frac{\partial J}{\partial d_0} = 0, \frac{\partial J}{\partial e_0} = 0. \qquad (5.9)$$

[20] The sufficient criteria for a minimum $\frac{\partial^2 J}{\partial a_0^2} > 0, \frac{\partial^2 J}{\partial b_0^2} > 0, \frac{\partial^2 J}{\partial c_0^2} > 0$ are not determined explicitly; the results show that it is fulfilled.

The loss function, Equation (5.8), is only solved for the parameters b_0, d_0, and e_0, while the other variables are determined by using Equations (5.4) – (5.6). Applying them to the particular side panel current leads to a_0 and c_0, and applying them to SP16 yields f_0. It can be done because these variables are not affected by superposition of the second sine function. Otherwise, it is almost impossible to obtain reasonable results.

The determined values of these parameters of the trigonometric functions are summarized in Appendix H. Additionally, the plots of the fitted current curves for each snapshot data are provided there.

2. Solar Vector from Currents

In order to calculate the vector from the spacecraft to the Sun, the incidence angle for a particular panel needs to be determined first. The following power equation follows from the first law of thermodynamics applied specifically to a solar panel

$$U_{Out}\, I_{Out} = \eta\ I_d\, A\, S \cos\Gamma, \tag{5.10}$$

where the left side in Equation (5.10) is the electrical power output being the product of the output panel current and the output voltage, while the power input by radiation from the Sun is on the right side degraded by the efficiency η and the inherent degradation I_d. The factor I_d contains effects caused by the design and the assembly of single solar cells into complete arrays, the shadowing of cells, and the operating temperature of the panel [Ref. 1]. The radiation input of the Sun on a surface normal to the Sun's rays is given by S, the solar constant at 1 AU, and the active panel area is given by A. Finally, $\cos\Gamma$ is referred to as cosine loss, where Γ is the Sun incidence angle. Life degradation effects caused by radiation, thermal eclipse cycling and micrometeoroid strikes are neglected in Equation (5.10).

Above equation reveals that the output current of an array is proportional to the Sun incidence angle. This relationship is used in the last section. It can be expressed as follows

$$\cos\Gamma_{SP} = C_{SP}\, I_{SP}\,, \tag{5.11a}$$

where C_{SP} is the proportional factor defined as

$$C_{SP} = \frac{U_{SCB}}{S\,A_{SP}\,\eta_{SP}\,I_{d,SP}}. \qquad (5.11b)$$

Because all panels are connected in parallel, their output voltage, denoted by the spacecraft bus voltage, is the same for each. With the exception of the gallium-arsenide panel[21], all other panels have identical dimensions making the active cell area equal. Furthermore, the efficiency factor and the inherent degradation can be assumed to be the same for different panels, since they are made of silicon with an efficiency of approximately 14%. The temperature differences between the panels used to determine the solar vector is small; shadowing is neglected. Therefore, the proportional factor C_{SP} can be assumed as the same at a sample point for different panels.

For the determination of the solar vector, Sun incidence angles of three solar panels are required, as shown in Figure 5.6. The solar vector is referred to the geometrical system, because the normal vectors of the solar panels are aligned with its axes. If needed, a transformation of the solar vector to the principal axes, the body axes of PANSAT, can be carried out using the direction cosine matrix given by Equation (C.4) in the appendix.

[21] The gallium-arsenide panel has a different efficiency and active cell area. It is designed to obtain some power if the bottom of the spacecraft is facing the Sun. Since it only produces current when the spacecraft bus voltage drops below 15.7 V, this case has never been observed yet. Thus, it does not cause problems in the above application.

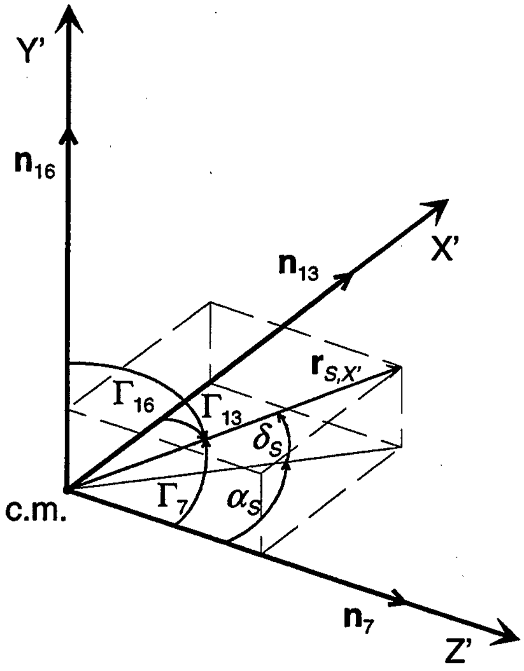


Figure 5.6: Solar Vector in the Geometrical Coordinate System

From observation of Figure 5.6, the solar vector can be calculated by the following equation

$$\mathbf{r}_{S,X'} = \left|\mathbf{r}_{S,X'}\right| \left[\cos\Gamma_{13} \quad \cos\Gamma_{16} \quad \cos\Gamma_7\right]^T . \tag{5.12}$$

Applying Equation (5.11a) on Equation (5.12) and normalizing the vector gives

$$\mathbf{r}_{S,X'} = \frac{\left[C_{13}\, I_{13} \quad C_{16}\, I_{16} \quad C_7\, I_7\right]^T}{\sqrt{(C_{13}\, I_{13})^2 + (C_{16}\, I_{16})^2 + (C_7\, I_7)^2}} . \tag{5.13}$$

As discussed earlier, the factors C_{SP} are considered to be the same at a sample point, so that the final result can be reached

$$\mathbf{r}_{S,X'} = \frac{\left[I_{13} \quad I_{16} \quad I_7\right]^T}{\sqrt{I_{13}^{\ 2} + I_{16}^{\ 2} + I_7^{\ 2}}} . \tag{5.14}$$

Equation (5.14) allows calculating the solar vector directly from the currents, which are taken as estimates from the least squares algorithm of Section V.B.1. For the calculation of the solar vector in different quadrants of the geometrical coordinate system, Equation (5.12) given for the eight quadrants can be found in Table 5.1. Equation (5.14) can be easily derived from Equation (5.12) as shown above.

The solar vector is computed using the estimated panel currents provided by the least squares fit. The two highest current values are taken leaving only two choices for the quadrant of the solar vector, by inspecting Table 5.1. SP4 and SP14 are ignored. The most likely quadrant is the one with the higher remaining current value. These three currents are applied on Equation (5.14) under consideration of Table 5.1. This computation of the solar vector is carried out in MATLAB; the code is provided in Appendix J.

Table 5.1: Calculation of Solar Vector from Sun Incidence Angles[22]

Quadrant	*Solar Vector* $\mathbf{r}_{S,X'}$
I, above	$\mathbf{r}_{S,X'} = \lvert\mathbf{r}_{S,X'}\rvert[\cos\Gamma_{13} \quad \cos\Gamma_{16} \quad \cos\Gamma_7]^T$
I, below	$\mathbf{r}_{S,X'} = \lvert\mathbf{r}_{S,X'}\rvert[\cos\Gamma_{13} \quad \cos\Gamma_{16} \quad -\cos\Gamma_{11}]^T$
II, above	$\mathbf{r}_{S,X'} = \lvert\mathbf{r}_{S,X'}\rvert[-\cos\Gamma_9 \quad \cos\Gamma_{16} \quad \cos\Gamma_7]^T$
II, below	$\mathbf{r}_{S,X'} = \lvert\mathbf{r}_{S,X'}\rvert[-\cos\Gamma_9 \quad \cos\Gamma_{16} \quad -\cos\Gamma_{11}]^T$
III, above	$\mathbf{r}_{S,X'} = \lvert\mathbf{r}_{S,X'}\rvert[-\cos\Gamma_9 \quad -\cos\Gamma_5 \quad \cos\Gamma_7]^T$
III, below	$\mathbf{r}_{S,X'} = \lvert\mathbf{r}_{S,X'}\rvert[-\cos\Gamma_9 \quad -\cos\Gamma_5 \quad -\cos\Gamma_{11}]^T$
IV, above	$\mathbf{r}_{S,X'} = \lvert\mathbf{r}_{S,X'}\rvert[\cos\Gamma_{13} \quad -\cos\Gamma_5 \quad \cos\Gamma_7]^T$
IV, below	$\mathbf{r}_{S,X'} = \lvert\mathbf{r}_{S,X'}\rvert[\cos\Gamma_{13} \quad -\cos\Gamma_5 \quad -\cos\Gamma_{11}]^T$

3. Energy considerations

The energy of a rigid body consists of potential energy and kinetic energy, which is made up of translational and rotational energy. When this body moves under the action of conservative forces[23], the sum of the kinetic and the potential energy remains constant [Ref. 19]. Hence, it may be stated as follows

$$E = T + U, \tag{5.15}$$

where E is the total energy, T is the kinetic energy and U the potential energy.

[22] The following convention is used for the classification of quadrants: the quadrant I is defined by the positive directions of the X', Y', and Z' axes. Quadrants II, III, and IV are given by turning counterclockwise about Y' starting at quadrant I. 'Above' and 'below' is referred to the X' - Z' plane.

The kinetic energy of a rigid body in scalar form is given as

$$T = \frac{1}{2} m v^2 + \frac{1}{2}\left(I_1 \omega_1{}^2 + I_2 \omega_2{}^2 + I_3 \omega_3{}^2\right), \tag{5.16}$$

where the first term is the translational energy and the second the rotational energy. In this equation, v is the velocity of the center of mass, ω_i denotes the angular velocities, m the mass of the rigid body, and I_i the principal moments of inertia.

It is shown in Chapter I, that the translational and rotational motion can be considered uncoupled since the distance from the Earth's center to the center of mass of PANSAT is large compared to the spacecraft's dimensions. Therefore, the rotational energy can be assumed as constant as long as energy dissipation is negligible. This approximation is true specifically for PANSAT, because the structure can be assumed as rigid, magnetic fields are insignificant and the antennas are small. As a result, the sum of the translational and potential energy should remain constant.

The potential energy can be calculated from the following equation taking the center of the Earth as a reference [Ref. 8]

$$U = m \cdot g \cdot r_E \left(1 - \frac{r_E}{\sigma}\right) \tag{5.17}$$

where g is the acceleration at the Earth's surface.

For the calculation of the translational energy, the orbit velocity is required. The so-called vis viva equation for an elliptical orbit is given as [Ref. 2]

$$v = \sqrt{K\left(\frac{2}{\sigma} - \frac{1}{a}\right)} \tag{5.18}$$

To get an idea how large the energies involved in the motion of this spacecraft are, the potential and kinetic energies are calculated. For November, 17, 14:29 *UTC*, the following required orbital elements obtained from STK are given

$$a = 6938.3\,\text{km},$$
$$e = 0.0011,$$
$$\tau = 338.2°.$$

[23] A conservative force satisfies the statement that the work of the force is independent of the path followed and is equal to (minus) the change in potential energy [Ref. 19].

Using Equation (3.1) with these parameters, the distance from the spacecraft's 'center of mass' to the 'center of Earth' is calculated as 6931.2 km. The potential energy using Equation (5.17) is

$$\begin{aligned} U_1 &= 56.9\,\mathrm{kg}\cdot 9.81\frac{\mathrm{m}}{\mathrm{s}^2}\cdot 6378.1\cdot 10^3\mathrm{m}\left(1-\frac{6378.1}{6931.2}\right). \\ &= 284.1\,\mathrm{MJ} \end{aligned}$$

Applying Equation (5.18), the velocity of the satellite in its orbit is

$$\begin{aligned} v_1 &= \sqrt{3.986\cdot 10^{14}\frac{\mathrm{m}^3}{\mathrm{s}^2}\left(\frac{2}{6931.2}-\frac{1}{6938.3}\right)\cdot 10^{-3}\mathrm{m}^{-1}} \\ &= 7587.3\frac{\mathrm{m}}{\mathrm{s}} \end{aligned}.$$

The translational energy is determined from the first term in Equation (5.16)

$$\begin{aligned} T_{trans,1} &= \frac{1}{2}56.9\,\mathrm{kg}\left(7587.3\frac{\mathrm{m}}{\mathrm{s}}\right)^2 \\ &= 1637.8\,\mathrm{MJ} \end{aligned}.$$

Finally, the sum of the potential and the translational energy.is calculated as

$$E' = 1921.9\,\mathrm{MJ}.$$

This value is subject to change, because the atmospheric drag, unlike other perturbation forces caused by the Earth's oblateness and third-body interactions, is a nonconservative force and takes energy continuously away from the orbiting satellite. Thus, the semi-major axis and the period are gradually decreasing. The orbit velocity is increasing, however, because Kepler's third law[24] must be satisfied [Ref. 6].

The rotational energy at this time is determined from the angular velocities of the spin and the nutation as observed in Section V.A.1. It is assumed that the angular velocity of the nutation occurs about both X' and Z'. Hence, the rotational term of Equation (5.16) provides the result

[24] Kepler's third law of planetary motion states that the squares of the periodic times of the planets are proportional to the cubes of the semi-major axes of their orbits [Ref. 19].

$$\begin{aligned} T_{rot} &= \frac{1}{2}\left[(1.42+1.30)\mathrm{kg\,m^2}\cdot(0.079\,\mathrm{rad/s})^2+0.98\,\mathrm{kg\,m^2}\cdot(0.105\,\mathrm{rad/s})^2\right] \\ &= 27.8\,\mathrm{mJ} \end{aligned},$$

giving a difference of 11 orders of magnitude with respect to the translational energy. This suggests that the assumption of considering the rotational energy being independently from the orbital motion is appropriate.

At the deployment of PANSAT, a spin of 5 °/s about the symmetrical axis is observed, which gives the rotational energy as

$$\begin{aligned} T_{rot} &= \frac{1}{2}0.98\,\mathrm{kg\,m^2}\cdot(0.087\,\mathrm{rad/s})^2 \\ &= 3.7\,\mathrm{mJ} \end{aligned}.$$

Comparing this value to the rotational energy 18 days later, it shows an increase in energy that violates laws of physics. It also indicates that the above assumption taking the angular velocity of the nutation for both body axes 1 and 2 is not appropriate.

The only explanation for an increase in the rotational energy in this quantity is a tip-off moment at deployment of PANSAT, i.e., the impulsive force of the spring was not directed through the 'center of mass' of the spacecraft causing a non-central impact. The delta translational energy of the satellite caused by the spring is calculated by using the separation velocity as determined in Chapter IV

$$\begin{aligned} \Delta T_{trans} &= \frac{1}{2}56.9\,\mathrm{kg}\cdot 1.2^2\,\frac{\mathrm{m}^2}{\mathrm{s}^2} \\ &= 41.0\,\mathrm{J} \end{aligned}.$$

Even this value is three orders of magnitude higher than the rotational energy indicating that a small misalignment of the separation vector can cause an increased rotational energy. Since the force was supposedly aligned with the axis of symmetry of the satellite and the 'center of mass' is some millimeters off, the impact was not central. Furthermore, the release of the spacecraft by the moment clamp could also provoke tip-off rates. Such an amount of energy becomes noticeable after a longer period of time as observed from the status snapshot data.

Figure 5.7 shows that the angular velocities of both spin and nutation increased after deployment and have dropped to about half the magnitude after reaching a maximum on about the 20th day. A decline in this quantity cannot be caused by an energy dissipation with this particular spacecraft configuration. It rather suggests that there is an angular velocity about another axis, which cannot be observed from the solar panel currents. This is discussed in detail in the following section.

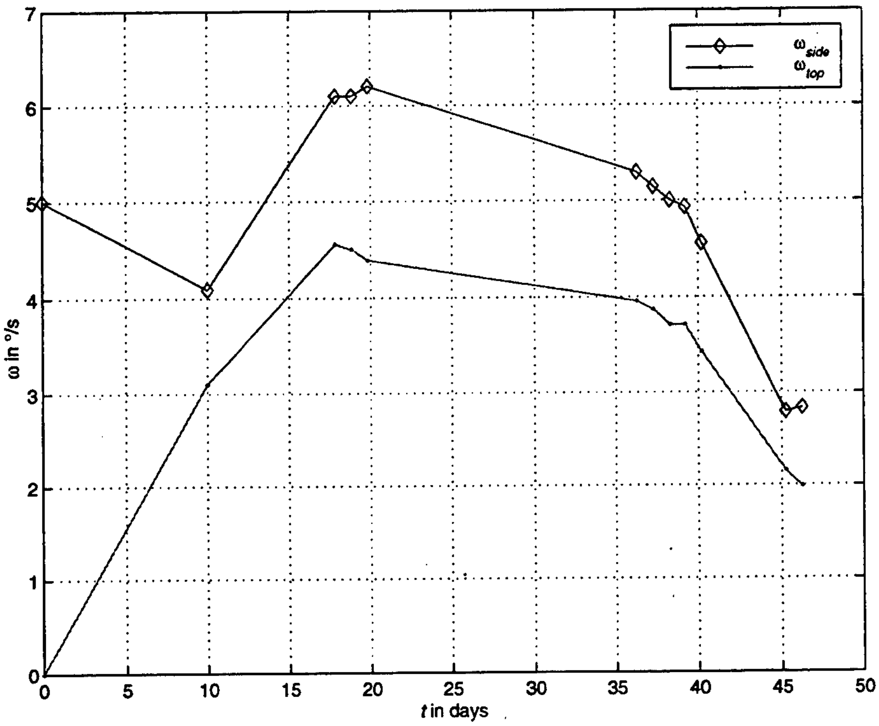


Figure 5.7: Angular Velocities Determined from Panel Currents

4. Examination of Angular Velocities

As observed in the last sections, cases may appear when the angular velocity cannot be determined unambiguously. It is always true when the solar vector referred to the spacecraft's geometrical axes is parallel to a normal vector of a solar panel's surface and a rotation about this axis occurs. However, this case is unlikely since PANSAT is a tumbling satellite changing its attitude with respect to the Sun constantly.

One component of the angular velocity cannot be determined from the solar panel data: the component of the angular velocity which is parallel to the Sun line. This is also true when the instantaneous axis of rotation is aligned with the Sun vector. But this is more questionable because the rotational motion consists of a spin, nutation and precession provoking the angular momentum vector to change not only in magnitude but also in direction.

Summarizing these facts, methods of determining the missing angular velocity component need to be discussed. The missing component is the major problem in analyzing the state of rotational motion.

Assuming an arbitrary angular velocity vector $\boldsymbol{\omega}$, it can be transformed into the geometrical system as follows

$$\begin{bmatrix} \omega_{X'} \\ \omega_{Y'} \\ \omega_{Z'} \end{bmatrix} = |\boldsymbol{\omega}| \begin{bmatrix} \cos\delta_\omega \sin\alpha_\omega \\ \sin\delta_\omega \\ \cos\delta_\omega \cos\alpha_\omega \end{bmatrix}, \tag{5.19}$$

where α_ω is the azimuth in the X', Z'-plane starting at the Z' axis and δ_ω is the elevation angle from this plane to the vector. Transforming these components into a Sun directed coordinate system by another azimuth and elevation angle so that the transformed Z' axis is aligned with the solar vector (see Figure 5.6), gives the following direction cosine matrix

$$\begin{aligned} \begin{bmatrix} \omega_{X',S} \\ \omega_{Y',S} \\ \omega_{Z',S} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\delta_S) & \sin(-\delta_S) \\ 0 & -\sin(-\delta_S) & \cos(-\delta_S) \end{bmatrix} \begin{bmatrix} \cos\alpha_S & 0 & -\sin\alpha_S \\ 0 & 1 & 0 \\ \sin\alpha_S & 0 & \cos\alpha_S \end{bmatrix} \begin{bmatrix} \omega_{X'} \\ \omega_{Y'} \\ \omega_{Z'} \end{bmatrix} \\ &= \begin{bmatrix} \cos\alpha_S & 0 & -\sin\alpha_S \\ -\sin\delta_S \sin\alpha_S & \cos\delta_S & -\sin\delta_S \cos\alpha_S \\ \cos\delta_S \sin\alpha_S & \sin\delta_S & \cos\delta_S \cos\alpha_S \end{bmatrix} \begin{bmatrix} \omega_{X'} \\ \omega_{Y'} \\ \omega_{Z'} \end{bmatrix} \end{aligned}. \tag{5.20}$$

The azimuth for this transformation can be determined from the solar vector as follows

$$\alpha_S = \arctan\left[\frac{\mathbf{r}_{SX'}(1)}{\mathbf{r}_{SX'}(3)}\right], \tag{5.21}$$

and the elevation angle as

$$\delta_S = \arcsin[\mathbf{r}_{SX'}(2)]. \tag{5.22}$$

The angular velocity about X' of the Sun directed system, can be calculated from

$$\omega_{X',S} = -\frac{\Delta\delta_S}{T_S}, \tag{5.23}$$

where $\Delta\delta_S$ is the change in the elevation angle between two samples and T_S is the sampling period. The negative sign is added because in fact the spacecraft rotates with respect to the Sun and not the Sun with respect to the geometrical system as the definitions imply.

The difference of the azimuth referred to the sampling period leads to another angular velocity

$$\omega_{Y'} = -\frac{\Delta\alpha_S}{T_S}, \tag{5.24}$$

which is referred to the geometrical frame.

As discussed earlier, the angular velocity about the Z' axis in either system undetectable.

The introduced angular velocity vector $\boldsymbol{\omega}$ is defined by the three unknown parameters ω, α_ω, and δ_ω. If they can be determined, the angular velocity vector will be known and will allow calculating of the missing component.

To accomplish this, there are only two equations to determine these three unknown parameters. These are

$$\omega_{X',S} = \omega\left(\cos\alpha_S \cos\delta_\omega \sin\alpha_\omega - \sin\alpha_S \cos\delta_\omega \cos\alpha_\omega\right), \tag{5.25}$$

and

$$\omega_{Y'} = \omega \sin\delta_\omega. \tag{5.26}$$

As discussed, the component about the X' axis is referred to the system directed to the Sun and the one about Y' to the geometrical axes. A transformation of each of them to the other system is impossible, because the component about the third axis is needed in the equation.

For November, 17, 1998 at 6:29:17 PST, the azimuth and elevation are determined from the solar panel currents using Equations (5.21) and (5.22), respectively. The currents are based on the curve fitted data, which is provided in Appendix H. The angles are shown in Figure (5.8). The azimuth is decreasing continually, which is essentially a positive spin about the systematical axis. The roughly sinusoidal change in the elevation angle is caused by the nutation. This angle changes about 10°. Therefore, it is not caused by a misalignment of the geometrical and the principal axes known as coning, because those angles are much smaller (see Appendix C). These results meet the observations from the last sections.

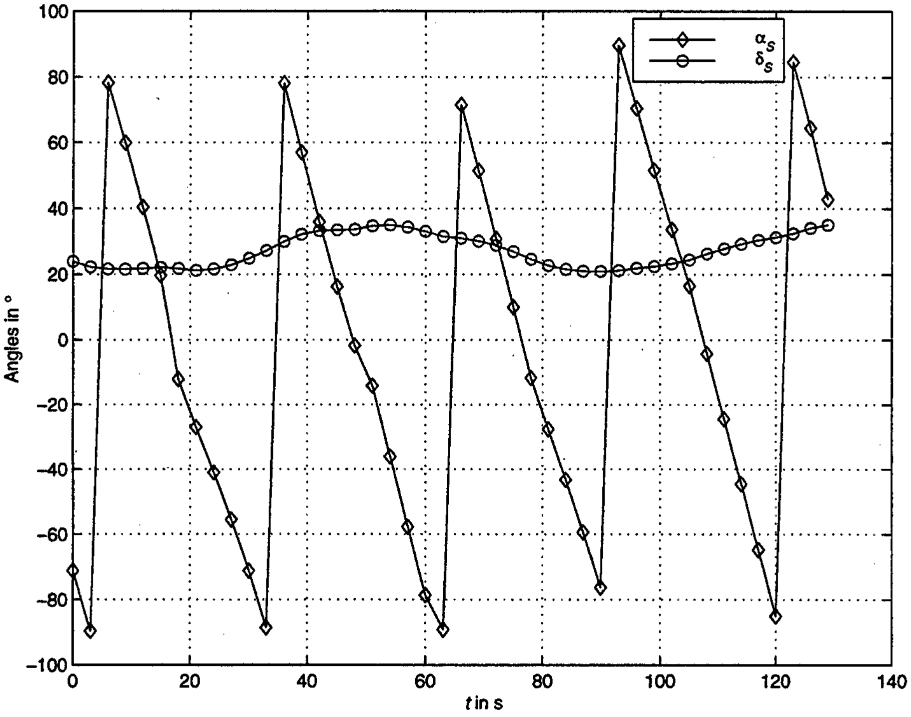


Figure 5.8: Azimuth and Elevation of Sun in Geometrical System (1998-11-17_06-29-17)

Figure (5.9) depicts the angular velocities $\omega_{X',S}$[25] and $\omega_{Y'}$ as computed from Equations (5.23) and (5.24). They are basically affected by the spin and the nutation, because precession can be neglected for this short amount of time. It can be seen that they are not constant. This is why the desired parameters cannot be determined by

[25] Every time the solar vector leaves a quadrant of the geometrical system for another one, the angular velocity, which is calculated from two samples in two different quadrants in this case, is ignored. This is because the error in the current is increased for lower magnitudes and, therefore, the curve fitting provides an overlap or a gap in the curves leading two a wrong angular velocity value (see Figure H.2).

applying a least squares algorithm on Equations (5.25) and (5.26), which requires these parameters as being constant.

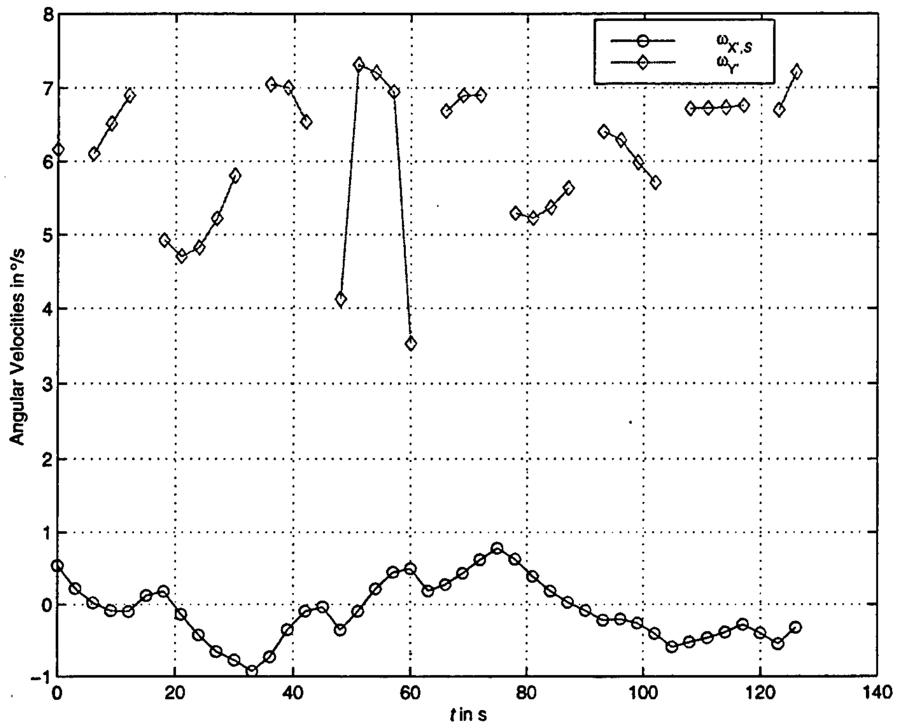


Figure 5.9: Angular Velocities of Spacecraft (1998-11-17_06-29-17)

The equations of motion, Equation (2.12), reveal that the angular velocities are only constant when the motion can be assumed torque-free and the moments of inertia are symmetric, i.e., $I_{X'X'} = I_{Z'Z'}$ (respectively, $I_2 = I_1$) causing an angular momentum vector fixed is space. Since the moments of inertia are not symmetric and gravity gradient torques are acting on the spacecraft, this assumption is not appropriate in this case. The moments of inertia about the X' and Y' axes are approximately nine percent off and, therefore, cannot be modeled symmetrically. However, Appendix C shows that the moments of inertia $I_{X'Y'}$, $I_{X'Z'}$, and $I_{Y'Z'}$ are fully negligible. The change in the angular momentum vector as estimated above also suggests that the torque cannot be neglected.

It is very difficult to determine the missing angular velocity component. Since the initial impact is unknown and, therefore, the entire amount of rotational energy at deployment as well, it cannot be used to calculate the desired component using energy equations. Technical specifications of the Ejection Mechanism could help to estimate the tip-off moment in favor of determining the entire rotational energy and the state of motion.

For this complex motion, it is impossible to resolve the equations of rotational motion symbolically. This is because the nonlinear terms must be kept in these equations due to the unsymmetrical moments of inertia and the presence of external torques being a driving moment for the spin axis. However, they could be solved for two special cases. The first takes the external torque into account (and eventually assume them constant) and sets the moments of inertia equal, $I_{X'X'} = I_{Z'Z'}$, and the second neglects the torque and takes the unsymmetrical moments of inertia into account. The combination of both calculations would lead to the involved motion of PANSAT. Numerical computations using Runge-Kutta algorithms could resolve the required transformation angles, α_ω and δ_ω, and the magnitude of the angular velocity ω, which change with time, in order to determine the missing angular velocity component.

C. ATTITUDE DETERMINATION

The determination of the attitude is an important goal to reach because it does not only influence the power state and battery charging of the satellite and, therefore, the lifetime, but it also affects the transmissions to the ground station due to a disturbed antenna pattern. Knowing the absolute attitude of the spacecraft would allow displaying it with respect to other reference frames. Additionally, it would allow analyzing the attitude with regard to stability aspects. An unstable attitude of the spacecraft is preferred, because a stable one could not be favorable concerning the communications. Last but not least, the attitude could be predicted using estimation algorithms in order to plan ahead communications with the satellite.

Since the sensors allow only tracking of one celestial body, the Sun, a three-axis attitude determination is impossible. It can only be determined with respect to the Sun and even then, the attitude is not solved unambiguously, because the orientation angle about the solar vector cannot be resolved.

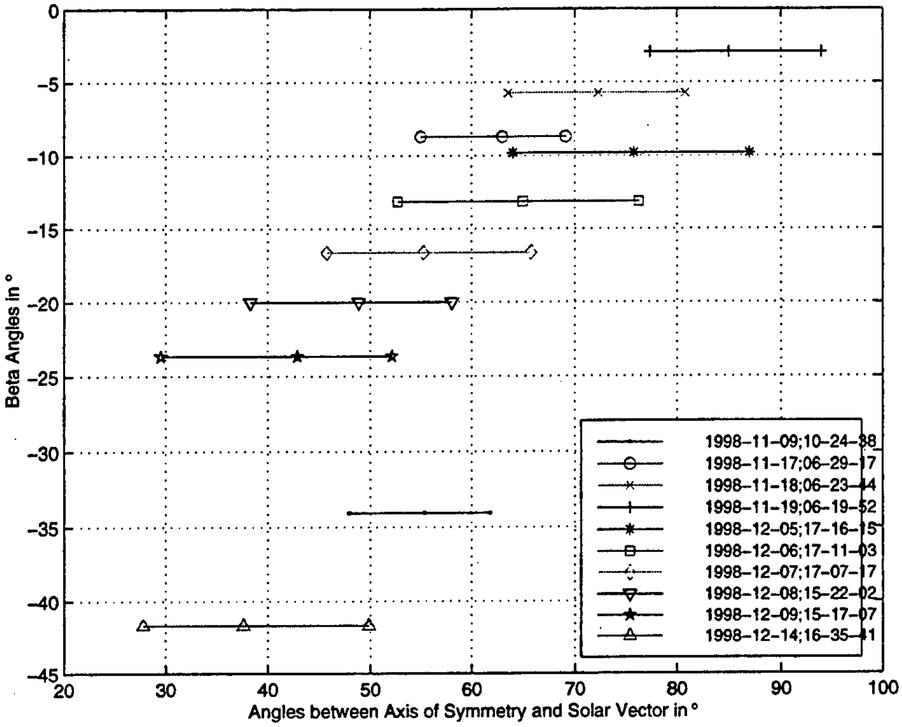


Figure 5.10: Angles Regarding Attitude with respect to the Sun

Figure (5.10) shows the angles between the geometrical axis of symmetry of the spacecraft and the Sun line vs. the beta angle. The former angle can be determined from $\pi/2 - \delta_S$, where the elevation angle is computed by Equation (5.22), while the beta angle on the vertical axis is calculated by Equation (3.29) from orbital tracking data determined by STK using TLEs. It links the data to a reference on the orbital plane, the orbit-plane axes system. However, it is clear from above discussion that the orientation of the symmetrical axis with respect to this system cannot be determined unambiguously. Having both angles for one set of status snapshots as shown in Figure 5.10 defines a cone in the orbit-plane system where the Y' axis is located.

Observing these angles obtained from snapshots in a daily sequence shows that they change almost linearly suggesting that the symmetrical axis stays in roughly the same orientation for a period over a few days. However, the effect of precession can be seen as a shift in the angles over longer periods of time.

The gradual increase in the delta of the angle between the symmetrical axis and the Sun line from November, 9, until November, 14, indicates that the effect of nutation, i.e., the nutation angle, has risen. It may verify a transfer of rotational energy from the minor axis to the major axis, which could lead to a stable attitude. Figure 5.7 also seems to verify this because the spin has declined after reaching a maximum on November, 19. Since the rotational energy can be assumed conserved, the non-detectable angular velocity must have increased since then.

More status data will provide a clearer picture and could help finding a correlation between the spacecraft based and the orbit-plane based Sun location in order to determine the attitude of the spacecraft. This can be achieved by calculating the solar vector by Equation (3.28) and determining the symmetrical axis/Sun line angle, which defines a cone of the Y' axis about the solar vector with respect to the orbit-plane system.

A single-axis attitude determination estimating the solar vector could be used to obtain and predict the attitude of PANSAT referred to the Sun. From that, a rough orientation of the satellite with respect to the observer, the ground station, could be used to evaluate communications with the satellite. A cone of possible positions of the axis of symmetry would be obtained, similar to that with respect to the orbit-plane system as described earlier. This could be done by using a Kalman filter. However, it requires more data to reach reasonable results.

Furthermore, the knowledge of the complete state of initial conditions at deployment would allow solving the equations of motion. To accomplish this the tip-off moment needs to be determined as described in the last section. The equations of motion provide the state of motion and the attitude of the spacecraft, which could be correlated to the data from the status snapshots. However, it must be noted, that errors in the initial conditions build up making it impossible to achieve reasonable results.

[This page is intentionally left blank.]

VI. RESULTS AND CONCLUSION

The objective of this report is to examine the attitude dynamics of PANSAT from current telemetry data. To accomplish this, a dynamics model is provided describing the rotational motion of the satellite under the action of environmental torques. The estimation of the external torques reveals that the gravity gradient torque is dominant being approximately one order of magnitude higher than the aerodynamic torque, while further torques can be neglected due to the orbit altitude and spacecraft configuration. Therefore, the gravity torque enters the equations of rotational motion.

Two high-resolution pictures taken at deployment show a positive spin about the geometrical axis of symmetry being (almost) aligned with the minor axis of the spacecraft. Observation reveals an angular velocity of 5 °/s, which originated from the Ejection Mechanism. This result is verified by the separation velocity determined to be close to NASA specification. Further examination of angular velocities from status data indicates a tip-off moment introduced during separation.

Visualization tools have been developed to display the attitude of PANSAT, e.g., at deployment with respect to the orbit-plane axes. This attitude is calculated from data provided by NASA. In addition, the roll-rate of the spacecraft can be shown relative to the Sun.

The rotational motion of the satellite is very complex and is composed of a spin, nutation, and precession. Spin and nutation can be clearly observed from solar panel data providing the location of the Sun, while the change in the angular momentum vector is estimated by investigating the dominant external torque. The spin of the spacecraft is manifested by panel temperature measurements as well.

A disadvantage of using the solar panels as a sensor to determine the angular velocities is that the component of the angular velocity aligned to the Sun line cannot be determined. Specifically for PANSAT, the rotational energy is assumed constant in the absence of energy dissipation. However, inspection of angular velocities determined from panel currents shows increases and major decreases in energy violating basic laws of physics. This confirms the missing angular velocity component and suggests a tip-off rate at deployment.

Analytical methods to determine this component fail because the angular velocities are not constant due to the presence of gravity gradient torques. Furthermore, the principal moments of inertia are unsymmetrical, i.e., they are all different. However, the principal axis directions can be assumed as aligned with the geometrical axes of the spacecraft. Due to the two mentioned reasons the equations of motion cannot be simplified.

The concept of attitude determination from the Sun location allows only resolving the attitude of the spacecraft relative to the Sun.. The initial attitude of the spacecraft at deployment is the only reference attitude allowing transformations to other coordinate systems. Since the mission life depends significantly on the attitude of the satellite to the Sun and, additionally, the transmissions to the ground station are affected by the attitude as well, it is desired to resolve the attitude problem.

This could be done by determining the rotational state of motion in order to obtain the missing angular velocity component and the tip-off moment. To accomplish this the equations of motions could be solved for special cases and the solutions could be combined to meet the involved motion of PANSAT. Using the complete state of initial motion and attitude, the equations of motion could be solved numerically in order to obtain the unambiguous rotational motion and the attitude. These simulations could be correlated to the data obtained from the solar panel currents.

Furthermore, one-axis attitude determination methods using the solar vector as the estimated parameter would be helpful in predicting the attitude with respect to the Sun, because an orientation directing the LVI to the Sun could cause serious power generation problems. Knowledge of the attitude and roll-rate is also useful to resolve the stability problem, because a stable attitude could be unfavorable regarding the communications.

Summarizing the results, the motion of the satellite in space can be completely described qualitatively. A quantitative determination of the attitude and complete state of rotational motion is mathematically complex and requires more data. The concept of using the Sun location as parameter in determining the attitude and roll-rate has limited applications for the purpose of attitude control; however, it has been demonstrated that using solar panel data yield meaningful information in the field of spacecraft dynamics.

[This page is intentionally left blank.]

APPENDIX A. TIME CONSIDERATIONS

Julian Dates

Julian dates (*JD*) are a convenient method in astrodynamics to preserve the date and time information in one variable. It is the continuous amount of time measured in days from the epoch January 1, 4713 B.C., 1200.

To find the Julian date from a known date and time, the day number of the year and the fraction of day from 0 *UT* must be added to the *JD* for January 0.0 *UT* of that year listed in the upper half of Table A.2. Julian Dates can also be found in current editions of "The Astronomical Almanac" issued by the U.S. Naval Observatory in the United States or by the Royal Greenwich Observatory in Great Britain, respectively.

Table A.1: Numbers of Days a Month to Determine Day of Year[26]

Jan	*Feb*	*Mar*	*Apr*	*May*	*Jun*	*Jul*	*Aug*	*Sep*	*Oct*	*Nov*	*Dec*
31	28	31	30	31	30	31	31	30	31	30	31

Table A.2: Julian Date at the Beginning of Year until 2000 and Relevant Standard Epochs

Date and Time of Epoch	*Julian Date*
1998, Jan 0.0 *UT*	2,450,813.5
1999, Jan 0.0 *UT*	2,451,178.5
2000, Jan 0.0 *UT*	2,451,543.5
Standard epoch, 1900 Jan 0.5 *UT*	2,415,020.0
Standard epoch, B1950.0 = 1950 Jan 0.923	2,433,282.423
Standard epoch, J2000.0 = 2000 Jan 1.5	2,451,545.0

Sidereal Time

Sidereal time is based on the rotation of the Earth relative to the stars. It is defined as the hour angle of the vernal equinox relative to the local meridian, which is measured positively in the counter-clockwise direction when viewed from the North Pole. Hence, sidereal time is needed to relate a reference in inertial space, the vernal equinox,

[26] In the case of a leap year occurring every four years, the February has 29 days instead of 28. To prove a leap year, it must be divisible by 4, while centuries (year ending with two zeros) need to be evenly divisible by 400. Therefore, the year 2000 will be a leap year.

to a point fixed on Earth as required for a coordinate transformation between the geocentric inertial system and the topocentric horizon system in Chapter III.

Following the definition of the sidereal time[27], the sidereal time associated with the Greenwich meridian is termed Greenwich Sidereal Time, θ_{GST} or GST. The sidereal time at a particular longitude is called Local Sidereal Time, θ_{LST} or LST. The following relationship is used to convert between GST and LST at a particular longitude λ[28], i.e., the longitude of the observer,

$$\theta_{LST} = \theta_{GST} + \lambda . \tag{A.1}$$

In order to determine the location of the Greenwich meridian with respect to vernal equinox, the Greenwich mean sidereal time at midnight θ_{GST0}, (0 h 0 min 0 s) $UT1$, using the epoch of J2000 as a basis, can be calculated using the following empirical formula [Ref. 3]

$$\theta_{GST0} = 1.753369 + 628.331971\, T_{UT1} + 6.8 \cdot 10^{-6}\, T_{UT1}{}^{2}, \tag{A.2}$$

where θ_{GST0} is in radian measure and T_{UT1} is the number of Julian centuries elapsed from the epoch J2000. The quantity of θ_{GST0} is reduced to a result within the range of 0 to 2π. The Julian centuries are calculated using the following relationship

$$T_{UT1} = \frac{JD_0 - 2{,}451{,}545.0}{36{,}525} . \tag{A.3}$$

JD_0 is the Julian day number for the date of interest, i.e., the JD at 0 h 0 min 0 s of the day.

To complete the calculation, the elapsed $UT1$ time on the day of interest must be added [Ref. 3]

$$\theta_{GST} = \theta_{GST0} + \omega_E\, UT1 \tag{A.4}$$

[27] Because of inaccuracies with the concept of sidereal time (precession of the equinox, change in apparent locations of the stars, and polar motion), the mean sidereal time refers to a mean equinox that moves only with secular motion (precession). The apparent sidereal time, which refers to the true equinox and is connected to the mean equinox by the equation of equinoxes, is only needed for high accuracy calculations. Thus, sidereal time (LST, GST) is considered to be mean sidereal time in this report.

[28] Longitudes are regarded as positive east of Greenwich and as negative west of the Greenwich meridian.

where ω_E is the Earth's mean angular rotation in radians per solar second (Appendix D) and $UT1$ is the universal time in solar seconds.

Solar Time and Universal Time

Solar time[29] is based on the interval between successive transits of the Sun over a local meridian, which establishes the basic solar day. Introduction of a Fictitious Mean Sun avoiding the variation in the Sun's apparent motion along the celestial equator leads to the mean solar time, which is used as primary reference for timekeeping. Mean solar time refers to an equinox that has only secular motions [Ref. 3].

The mean solar time at Greenwich is defined as universal time UT. Because the Fictitious Mean Sun is defined mathematically as a function of sidereal time, UT is derived from sidereal time. Correction of $UT0$ for polar motion making time independent of station location proceeds to $UT1$.

The most commonly used time system is coordinated universal time UTC, which is derived from atomic time. UTC is the basis of civil time keeping systems and is designed to follow $UT1$ within ±0.9 s. Thus, meeting a sufficient precision, differences between $UT1$ and UTC can be neglected in this report.

[29] Since the solar day is measured relative to the Sun, an observer on Earth has to revolve once *and* observe the Sun at the same location. Therefore, a sidereal day is slightly shorter than a solar day because the stars remain essentially in the same location over time.

[This page is intentionally left blank.]

APPENDIX B. TRANSFORMATION OF STANDARD EPOCHS

In this appendix, a brief discussion follows on the B1950 and J2000 systems and the conversion between them.

The geocentric inertial system is "fixed" in inertial space by using the vernal equinox as a reference and the Earth's equator as fundamental plane. However, both are subject to change due to perturbations by the Moon, the Sun and the planets. The equinox precesses as a result of the change in the ecliptic's orientation caused by the planets. Furthermore, the Earth is not spherical producing a precession of the polar axis and, therefore, the equatorial plane due to the gravitational fields of the Sun and the Moon. In addition, the Moon is mainly responsible for the small oscillations in the Earth's rotation axis, called nutation. Hence, the geocentric frame is not inertial unless it is fixed to a specified time.

The B1950 system relies on the Besselian year and is no longer considered precise. The J2000 system is referred to the J2000 epoch and accounts for the equinox's and equator's motion precisely. Therefore, it is considered as the best realization of an ideal, inertial frame at a fixed epoch.

Many modern books and tracking programs provide orbit information in the J2000 epoch; thus, a transformation from B1950 to J2000 is given using a matrix operation. It is usable for any position or velocity vectors $\mathbf{X}$ [Ref. 3]

$$\mathbf{T}_{J2000,B1950} = \begin{bmatrix} 0.999926 & -0.011181 & -0.004859 \\ 0.011181 & 0.999937 & -0.000027 \\ 0.004859 & -0.000027 & 0.999988 \end{bmatrix}, \qquad \text{(B.1a)}$$

$$\mathbf{X}_{J2000} = \mathbf{T}_{J2000,B1950}\,\mathbf{X}_{B1950}\,. \qquad \text{(B.1b)}$$

[This page is intentionally left blank.]

APPENDIX C. PRINCIPAL AXES COORDINATE SYSTEM

For studying the dynamics of the satellite, a principal axes coordinate system is required. Its center coincides with the center of mass of the spacecraft and the axis directions are given by the eigenaxes. The transformation from the geometrical axes of PANSAT as introduced in Section III.A.3. to those principal axes is carried out below. Furthermore, the data required is provided.

The mass of the satellite and the location of the 'center of mass' were measured at NASA Goddard Space Flight Center, GSFC, Greenbelt, MD. The mass is determined to be 56.9 kg and the coordinates of the 'center of mass' referred to the X, Y, Z-system[30] are as follows

$$x^{\mathrm{c.m.}} = 0.1\,\mathrm{cm},$$
$$y^{\mathrm{c.m.}} = 23.2\,\mathrm{cm},$$
$$z^{\mathrm{c.m.}} = -0.2\,\mathrm{cm}.$$

Unfortunately, only two of six moments of inertia were determined at GSFC (referred to c.m.)

$$I_{X'X'} = 1.299\,\mathrm{kg\,m^2},$$
$$I_{Y'Y'} = 0.976\,\mathrm{kg\,m^2}.$$

Comparing these measurements to the data based on the CAD model of the satellite reveals significant differences. This is because the masses of cabling, screws, and the added mass in the support cylinder are not taken into account in the CAD model in addition to the error resulting from discretization of geometrical objects and their masses. The mass based on that model is given as 45.5 kg being significantly lower than the measured value.

The moments of inertia referred to the 'center of mass' determined from the CAD model are as follows

[30] This body-fixed system is shown in Figure 1.1.

$$I_{X'X'} = 1.066\,\mathrm{kgm^2},$$
$$I_{Y'Y'} = 0.792\,\mathrm{kgm^2},$$
$$I_{Z'Z'} = 1.160\,\mathrm{kgm^2},$$
$$I_{X'Y'} = -0.001\,\mathrm{kgm^2},$$
$$I_{Y'Z'} = 0.003\,\mathrm{kgm^2},$$
$$I_{X'Z'} = 0.003\,\mathrm{kgm^2}.$$

Considering the relative error of the measured moments of inertia and the CAD values, it is calculated by the following formula [Ref. 26]

$$\frac{\Delta x}{x} = \frac{x_m - x_t}{x_m}, \tag{C.1}$$

where $\Delta x/x$ is the relative error, x_m is the measured value, and x_t is the true value. By using above data, they are determined to[31]

$$\left|\frac{\Delta I_{X'X'}}{I_{X'X'}}\right| = 0.219 = 21.9\%,$$
$$\left|\frac{\Delta I_{Y'Y'}}{I_{Y'Y'}}\right| = 0.232 = 23.2\%.$$

Both errors are close suggesting that the screws and cables, which are not taken into account in the CAD-based calculation of the moments of inertia, can be taken as uniformly distributed. Hence, one can assume that the missing moments of inertia differ about the same value. This leads to an estimate about the Z' axis of $I_{Z'Z'} = 1.42\,\mathrm{kgm^2}$ assuming a mean relative error derived from above values of 0.225, while the impact on $I_{X'Y'}$, $I_{Y'Z'}$, $I_{X'Z'}$ can be neglected due to their small magnitude.

In order to determine the principal moments of inertia by solving the eigenvalue problem, it is useful to compile the moments of inertia into an inertia tensor **I'**

$$\mathbf{I'} = \begin{bmatrix} I_{X'X'} & I_{X'Y'} & I_{X'Z'} \\ I_{Y'X'} & I_{Y'Y'} & I_{Y'Z'} \\ I_{Z'X'} & I_{Z'Y'} & I_{Z'Z'} \end{bmatrix}, \tag{C.2}$$

[31] The moments of inertia determined at GSFC are assumed as the true values, while the CAD based values are taken as measured in Equation (C.1).

where $I_{X'Y'} = I_{Y'X'}$, $I_{X'Z'} = I_{Z'X'}$, and $I_{Y'Z'} = I_{Z'Y'}$. The eigenvalue problem is solved by the following equation

$$|\mathbf{I}' - \lambda\,\mathbf{E}| = \begin{vmatrix} I_{X'X'} - \lambda & -I_{X'Y'} & -I_{X'Z'} \\ -I_{Y'X'} & I_{Y'Y'} - \lambda & -I_{Y'Z'} \\ -I_{Z'X'} & -I_{Z'Y'} & I_{Z'Z'} - \lambda \end{vmatrix} = 0, \tag{C.3}$$

where $\mathbf{E}$ is the identity matrix. The eigenvalues λ represent the desired principal moments of inertia, while the eigenvectors $\mathbf{e}_i$ define the principal axes. The calculation is carried out with MAPLE; the source code is part of Appendix J.

Applying Equation (C.3) to the approximated data given above and arranging the eigenvalues in order of quantity, the principal moments of inertia are as follows

$$I_1 = 1.42\ \mathrm{kgm}^2,$$
$$I_2 = 1.30\ \mathrm{kgm}^2,$$
$$I_3 = 0.98\ \mathrm{kgm}^2,$$

and the corresponding eigenvectors are

$$\mathbf{e}_1 = [-0.025 \quad -0.007 \quad 1.000]^T,$$
$$\mathbf{e}_2 = [-1.000 \quad -0.003 \quad -0.025]^T,$$
$$\mathbf{e}_3 = [0.003 \quad -1.000 \quad -0.007]^T.$$

Eigenvector $\mathbf{e}_1$ represents axis B_1, $\mathbf{e}_2$ represents B_2, and $\mathbf{e}_3$ represents B_3. The coordinate transformation leads to a change in the direction of the coordinate axes. Axis B_1 points in the positive direction of Z', B_2 in the negative direction of X', and B_3 in the negative direction of Y' (Figure D.1; O marks the center of the X, Y, Z-system). The axes B_1, B_2, and B_3 form a right-hand Cartesian system. The rotation angles between these principal axes and the axes of the X', Y', Z'-system are as follows

$$\eta_{X'} = 0.007\ \mathrm{rad} = 0.4°,$$
$$\eta_{Y'} = 0.024\ \mathrm{rad} = 1.4°,$$
$$\eta_{Z'} = 0.003\ \mathrm{rad} = 0.20°.$$

As a first-order approximation, these values can be neglected and, therefore, the body axes taken as aligned with the geometrical axes.

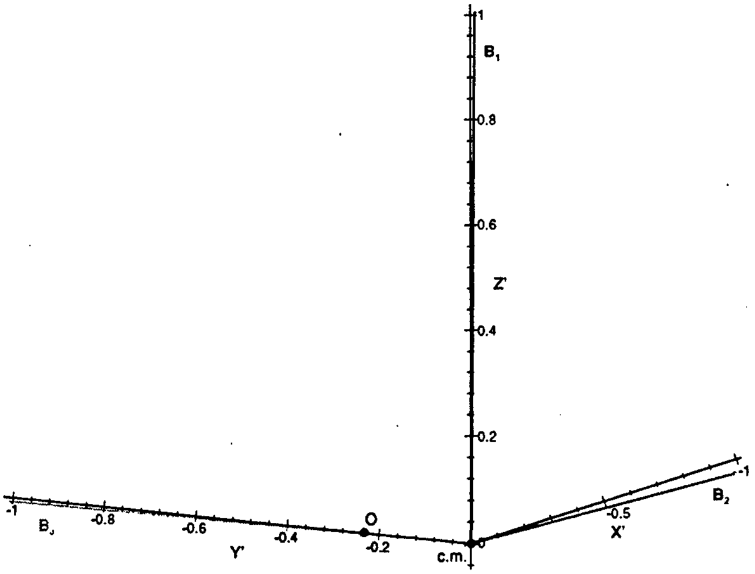


Figure C.1: Principal Axes B_1, B_2, B_3 of PANSAT

For completeness, the transformation matrix from the geometrical axes X', Y', Z' to the body axes B_1, B_2, B_3 is determined by applying a 3-2-1 rotation sequence with the Euler angles $\eta_{X'}$, $\eta_{Y'}$, and $\eta_{Z'}$ (Table 3.1). Due to the change in the axis directions, another matrix is added leading to the desired DCM

$$\mathbf{T}_{BX'} = \begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cs\eta_{X'} & -sn\eta_{X'} \\ 0 & -sn\eta_{X'} & cs\eta_{X'} \end{bmatrix} \begin{bmatrix} cs\eta_{Y'} & 0 & -sn\eta_{Y'} \\ 0 & 1 & 0 \\ sn\eta_{Y'} & 0 & cs\eta_{Y'} \end{bmatrix} \begin{bmatrix} cs\eta_{Z'} & sn\eta_{Z'} & 0 \\ -sn\eta_{Z'} & cs\eta_{Z'} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.4a}$$

$$\mathbf{B} = \mathbf{T}_{BX'}\,\mathbf{X'} \tag{C.4b}$$

APPENDIX D. SPACEFLIGHT CONSTANTS

Spaceflight constants and the geographic location of Monterey required by observer-based coordinate transformations are presented in this appendix.

Table D.1: Spaceflight Constants

Geocentric gravitational constant, K	$3.986\ 10^{14}$	m^3/s^2
Equatorial radius, r_E	$6378.14\ 10^3$	m
Astronomical Unit, 1 AU	$1.496\ 10^{11}$	m
Solar constant, S	1358	W/m^2 at 1 AU
Earth's mean angular rotation, ω_E	$7.292116\ 10^{-5}$	rad/solar s
Acceleration of gravity at the Earth's surface, g	9.81	m/s^2

Table D.2: Location of Monterey, CA

Geodetic latitude, φ_0	36.60° N
Longitude, λ	121.88° W

[This page is intentionally left blank.]

APPENDIX E. PICTURES OF DEPLOYMENT

This appendix provides the pictures that are used to determine the initial state of motion of the spacecraft at deployment.

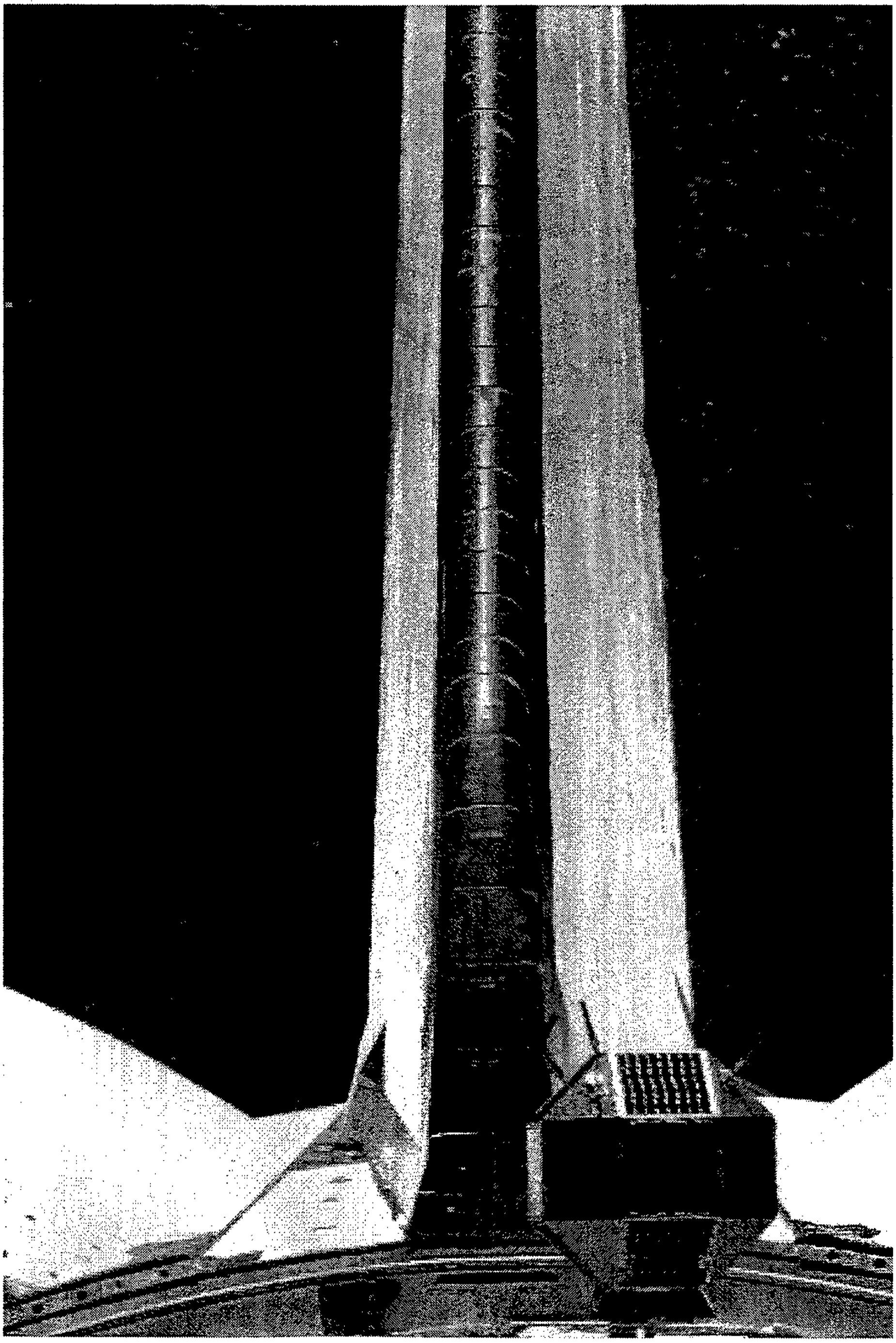


Figure E.1: PANSAT at Deployment, $t = t_1$ (#S95E5039)

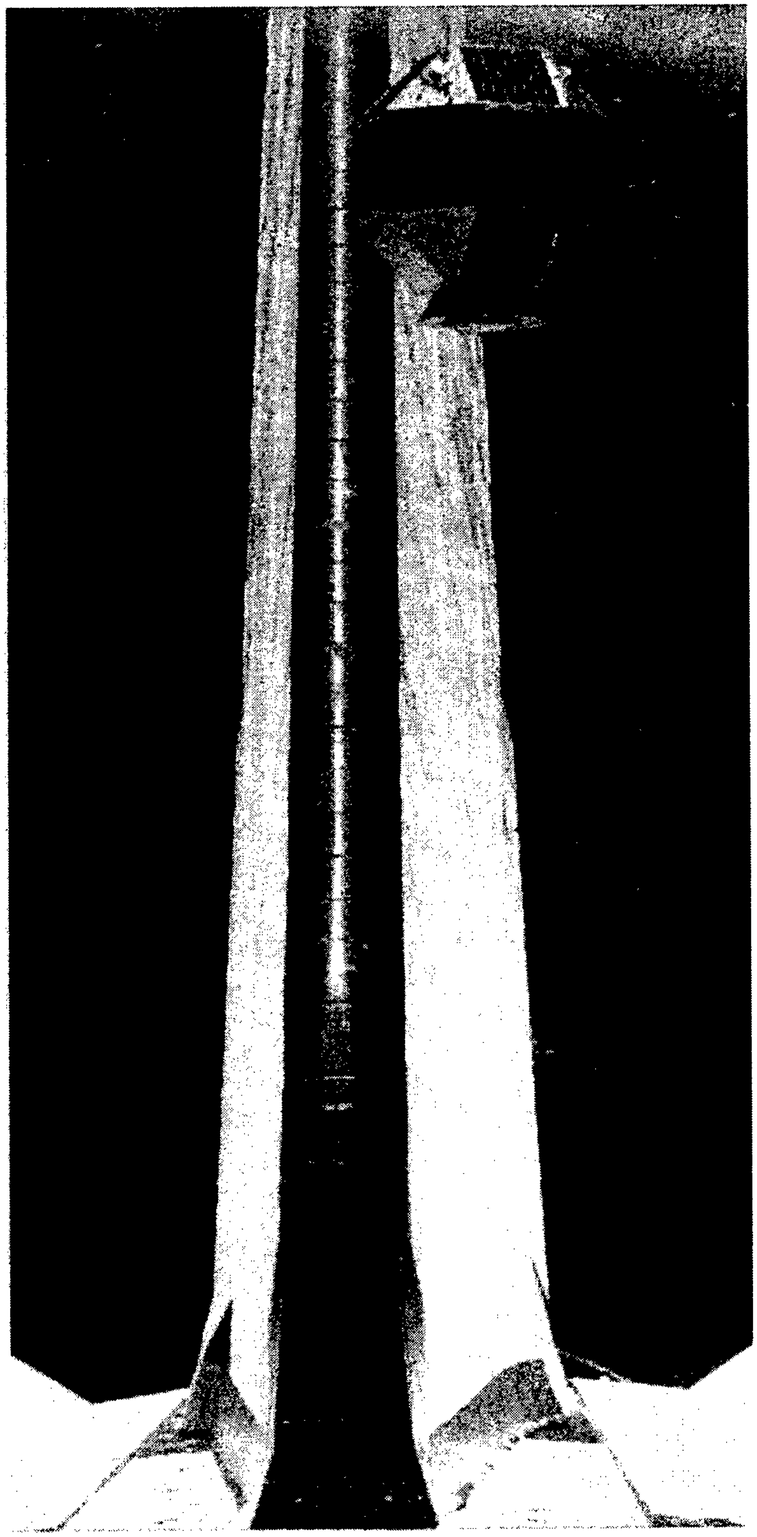


Figure E.2: PANSAT at Deployment, $t = t_1 + 2$ s (#S95E5040)

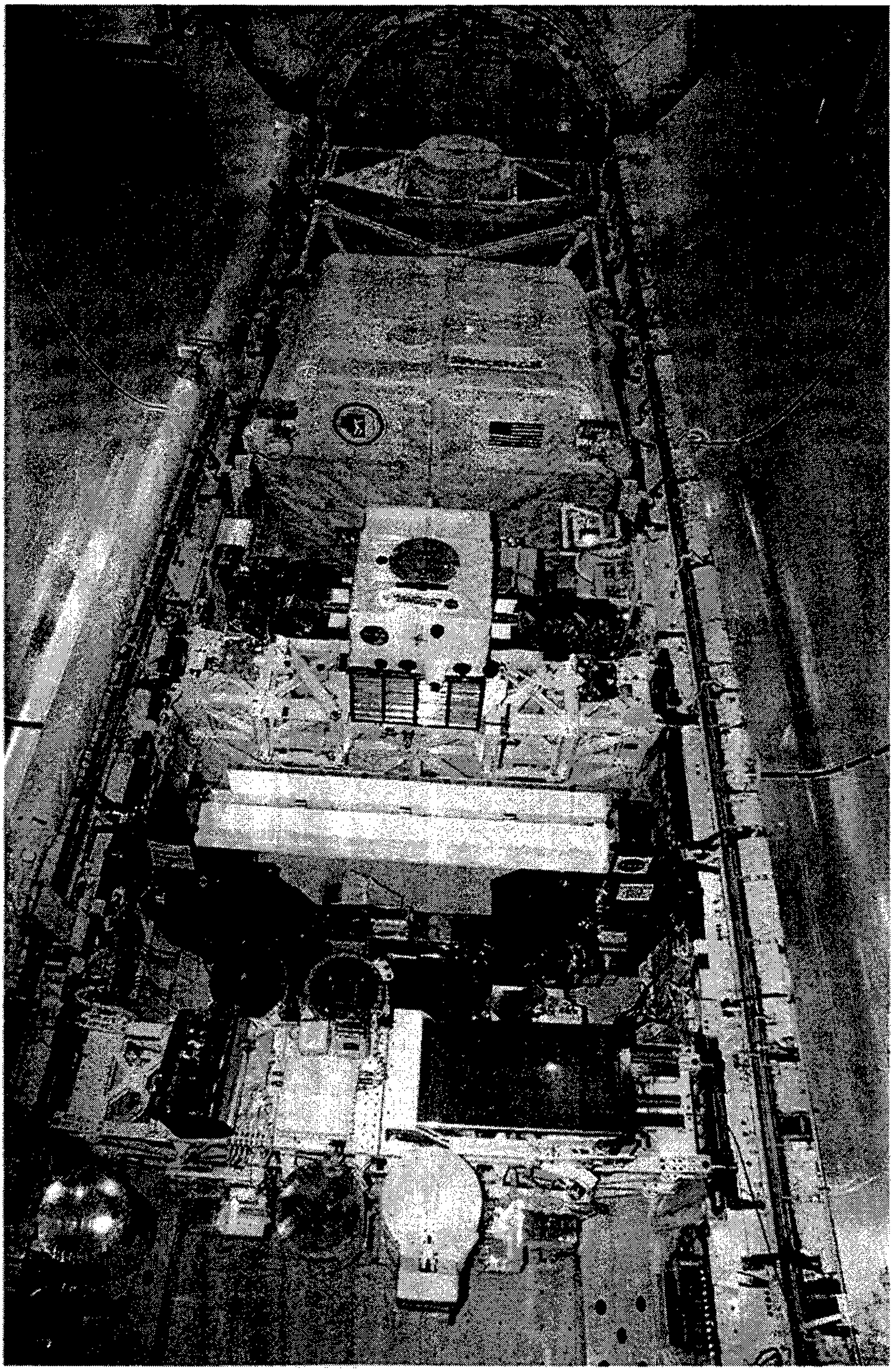


Figure E.3: Payload of Orbiter Containing Canister with PANSAT

[This page is intentionally left blank.]

APPENDIX F. PICTURES OF PANSAT AT INTEGRATION

The following pictures were taken at GSFC as the fully assembled spacecraft was integrated into the canister. They show the antennas after being bent for safety reasons.

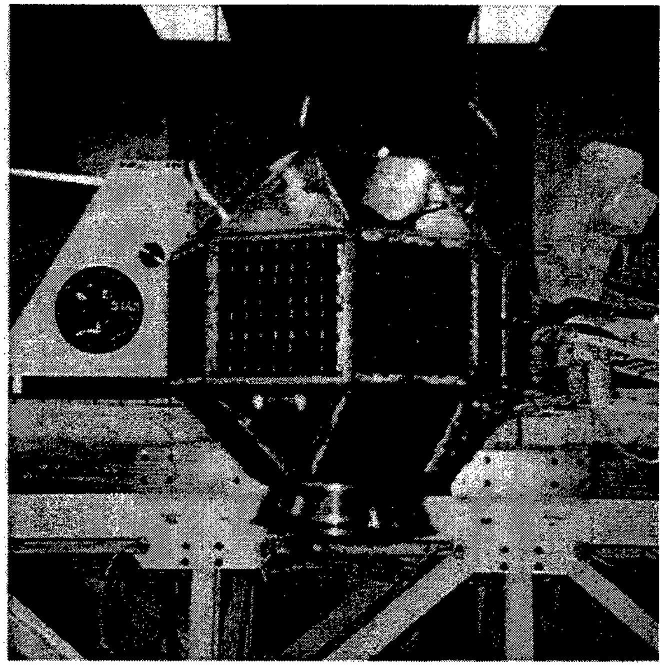


Figure F.1: Side View of PANSAT

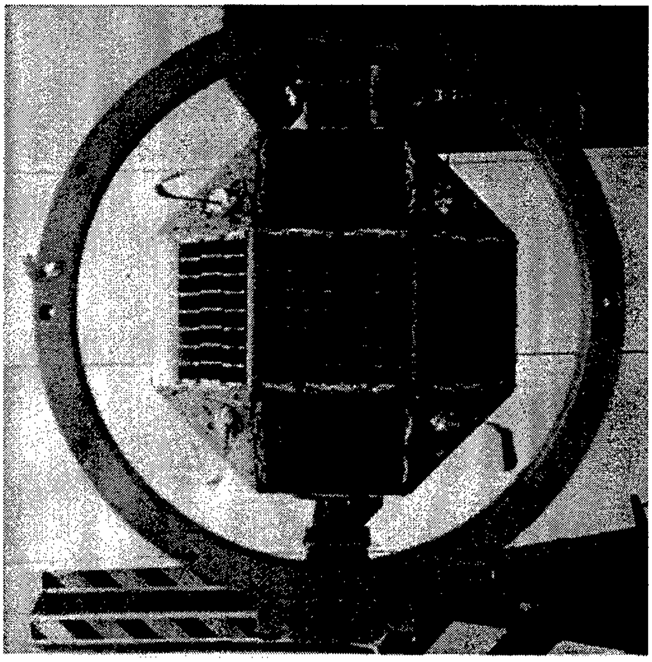


Figure F.2: Top View of Assembled Spacecraft

[This page is intentionally left blank.]

APPENDIX G. SOLAR PANEL SNAPSHOT DATA

Solar panel data based on status snapshots are provided in this appendix. It is used to observe the spacecraft's attitude and rotational motion with respect to the Sun (see Chapter V).

Plots of solar panel currents and temperatures are presented in the following. An overview of the snapshot data available is given in Table G.1.

Table G.1: Date and Time of Provided Snapshot Data

Index	*Date*	*Time (PST)*	*Number of*	*Time Window*
i	*of First Sample*		*Samples N*	$N \cdot T_s$ *in* s
1	11/09/1998	10:24:38	27	179
2	11/17/1998	06:29:17	27	123
3	11/18/1998	06:23:44	20	190
4	11/19/1998	06:19:52	14	135
5	12/05/1998	17:16:15	49	258
6	12/06/1998	17:11:03	26	223
7	12/07/1998	17:07:17	21	161
8	12/08/1998	15:22:02	24	126
9	12/09/1998	15:17:07	34	209
10	12/14/1998	16:35:41	32	211

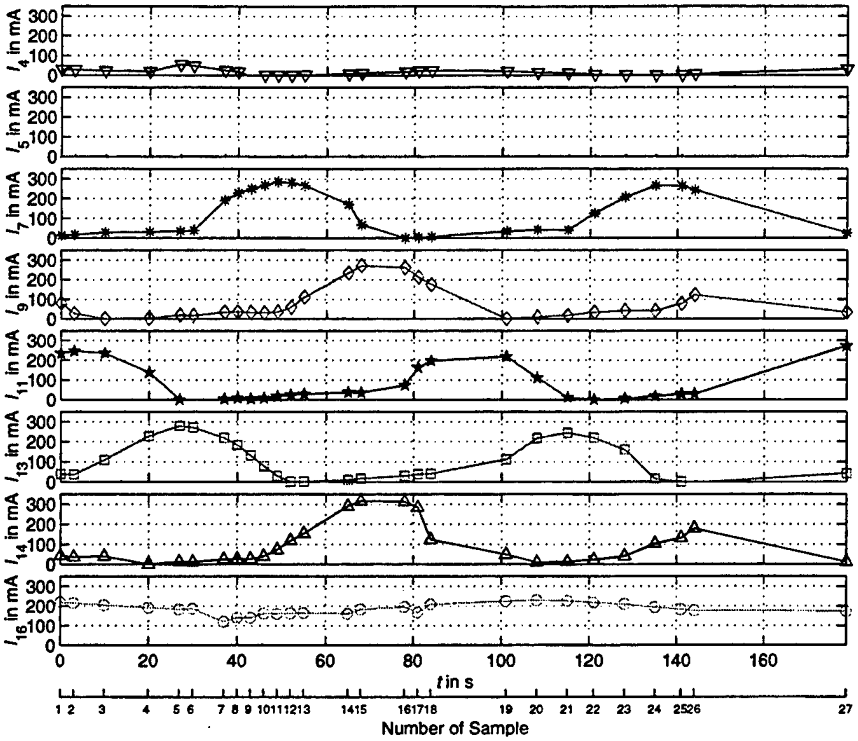


Figure G.1: Solar Panel Currents (i = 1, see Table G.1)

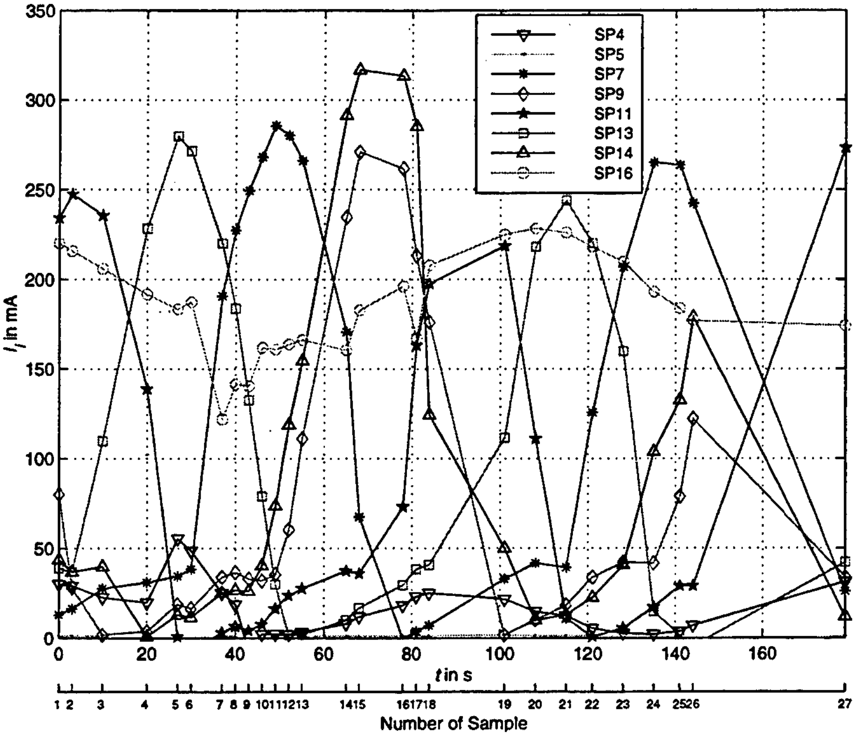


Figure G.2: Solar Panel Currents, in one Plot (i = 1)

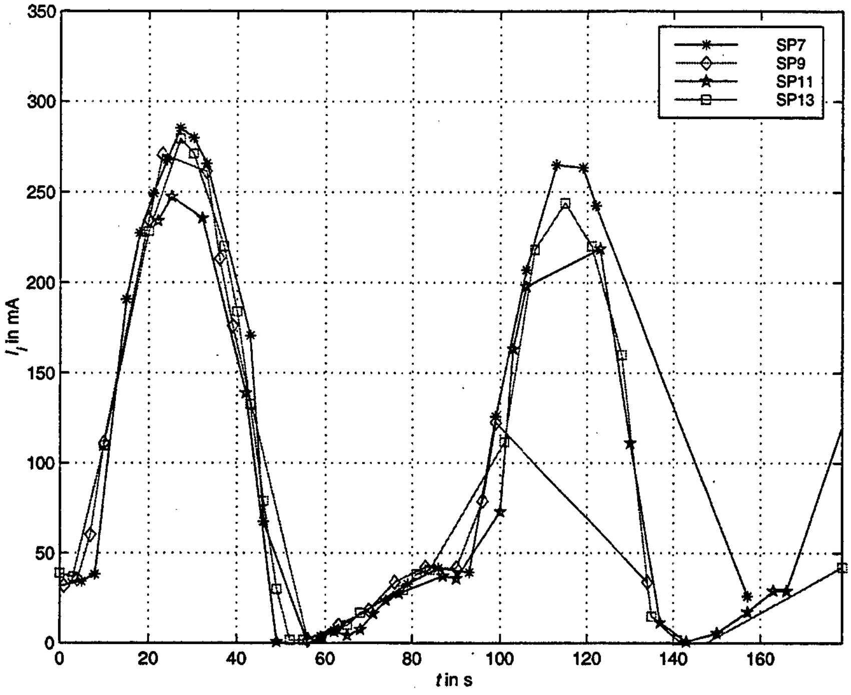


Figure G.3: CV Panel Currents, Phase Shift Suppressed (i = 1)

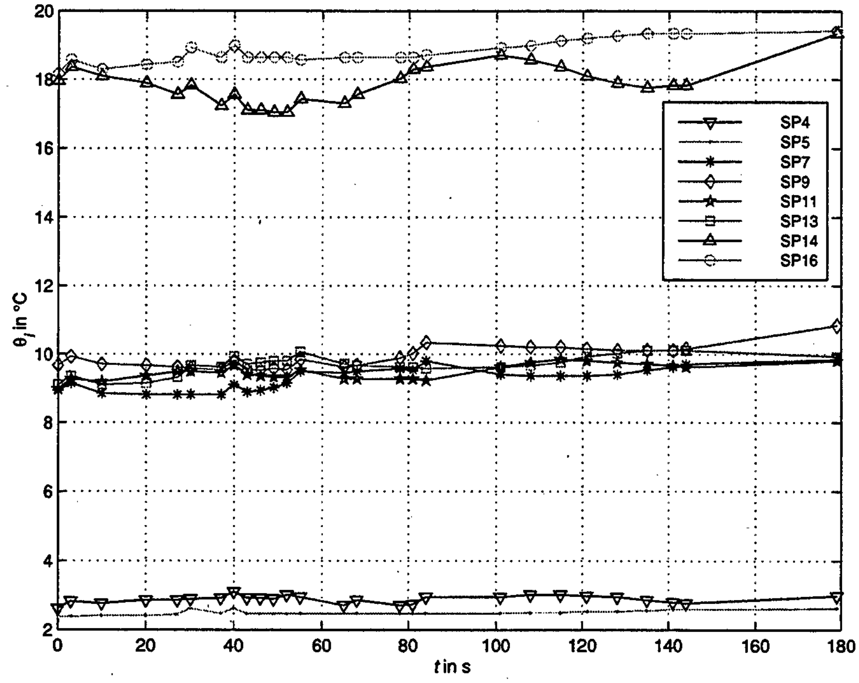


Figure G.4: Temperatures of Panels Equipped with Current Sensors (i = 1)

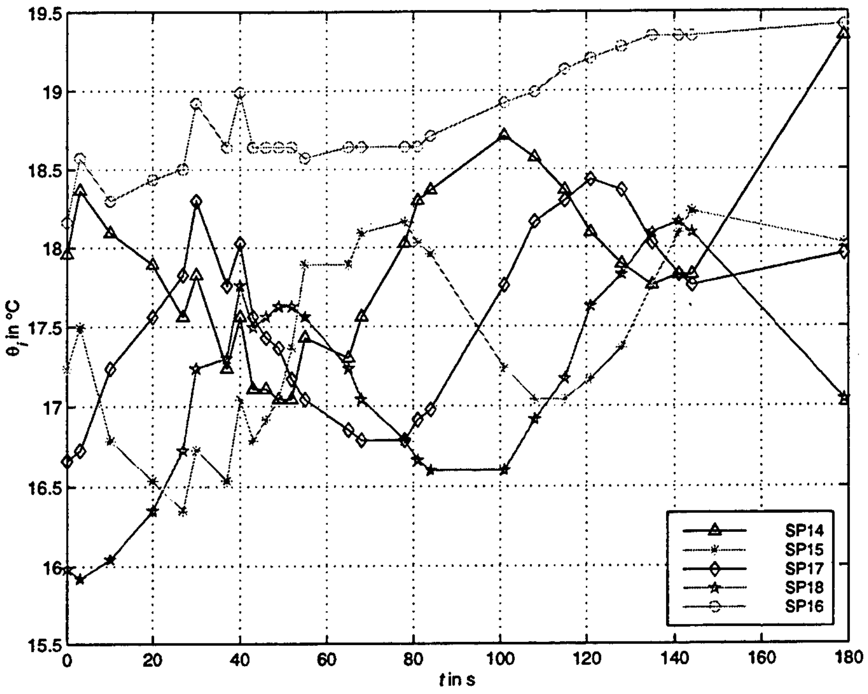


Figure G.5: Temperatures of UDCK and TP panels (*i* = 1)

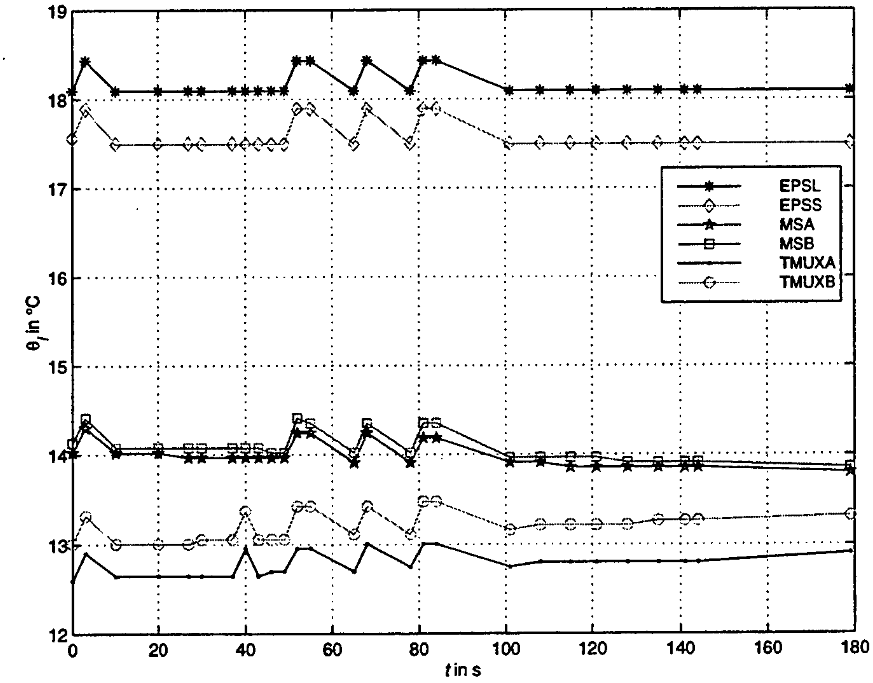


Figure G.6: Temperatures of Components in the Upper Deck (*i* = 1)

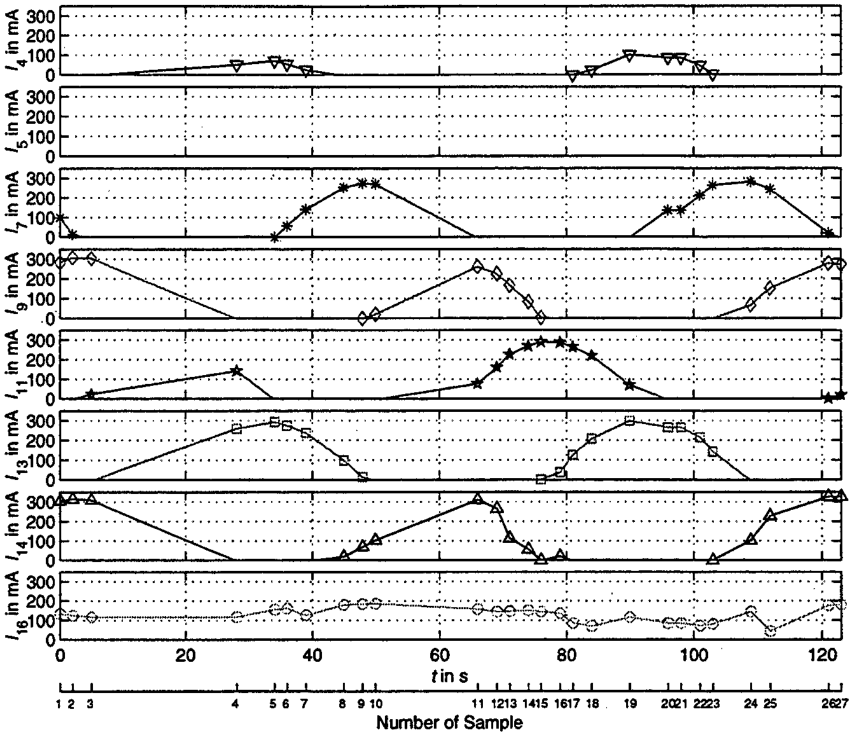


Figure G.7: Solar Panel Currents (*i* = 2, see Table G.1)

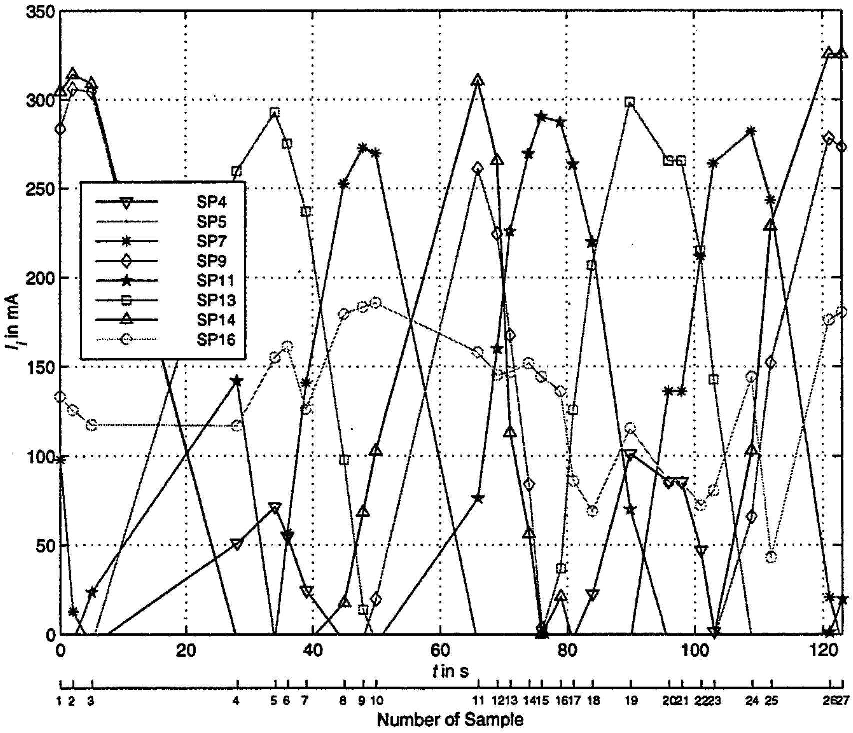


Figure G.8: Solar Panel Currents, in one Plot (*i* = 2)

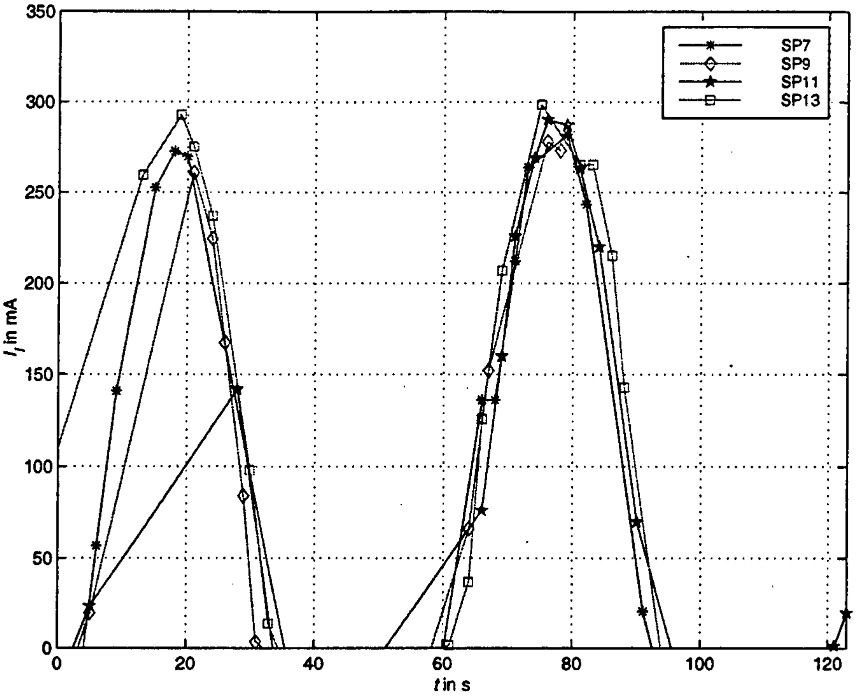


Figure G.9: CV Panel Currents, Phase Shift Suppressed (i = 2)

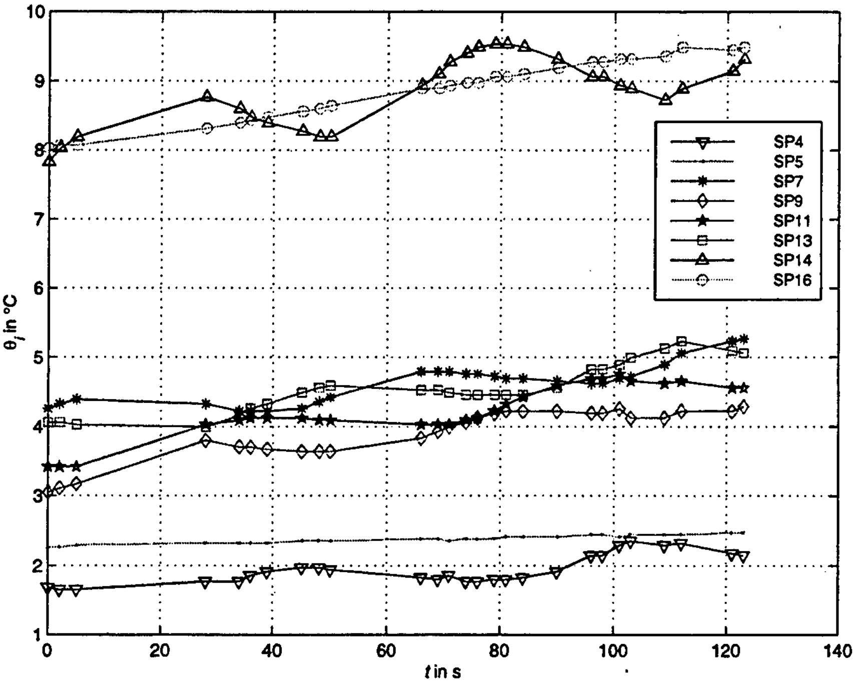


Figure G.10: Temperatures of Panels Equipped with Current Sensors (i = 2)

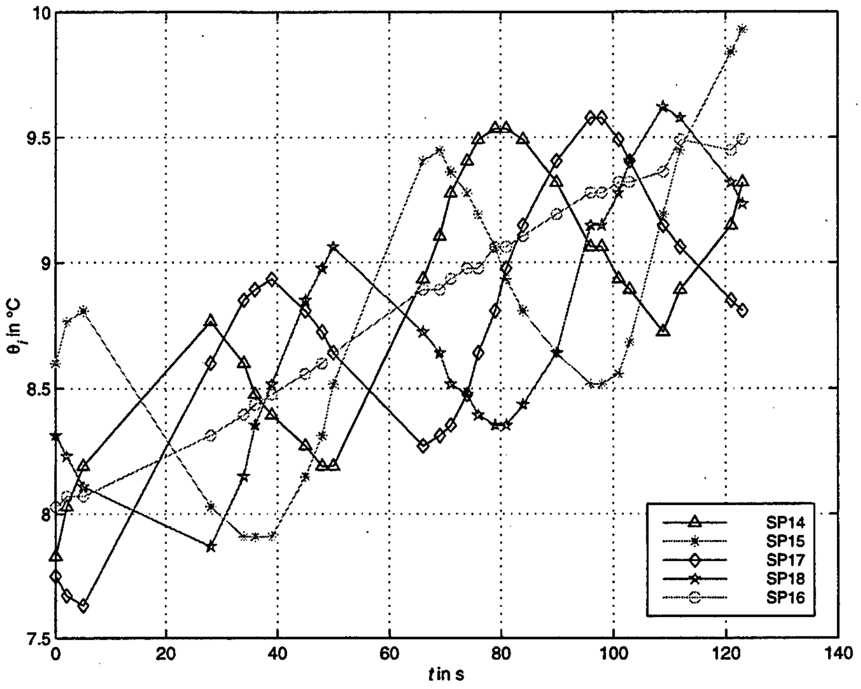


Figure G.11: Temperatures of UDCK and TP panels ($i = 2$)

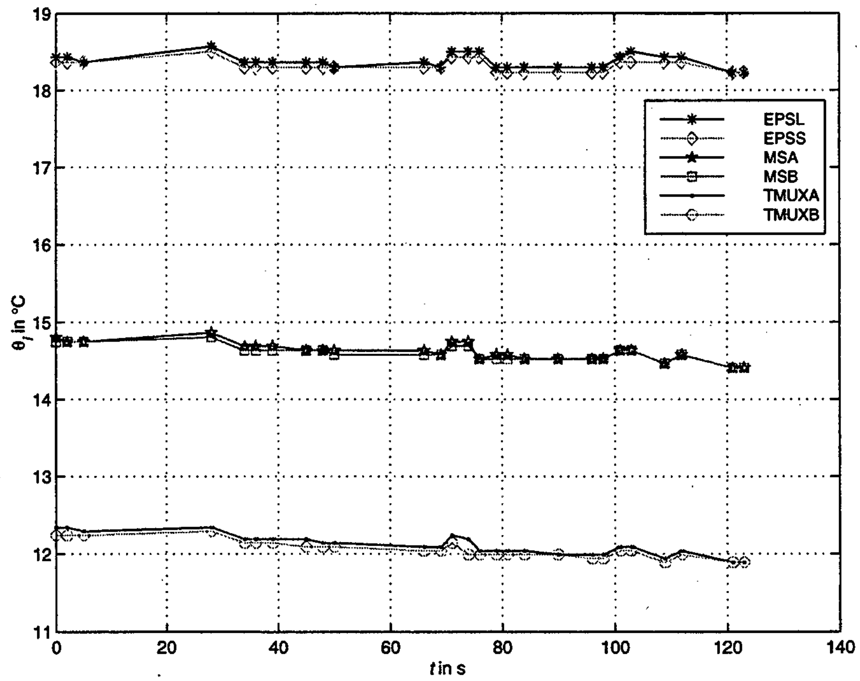


Figure G.12: Temperatures of Components in the Upper Deck ($i = 2$)

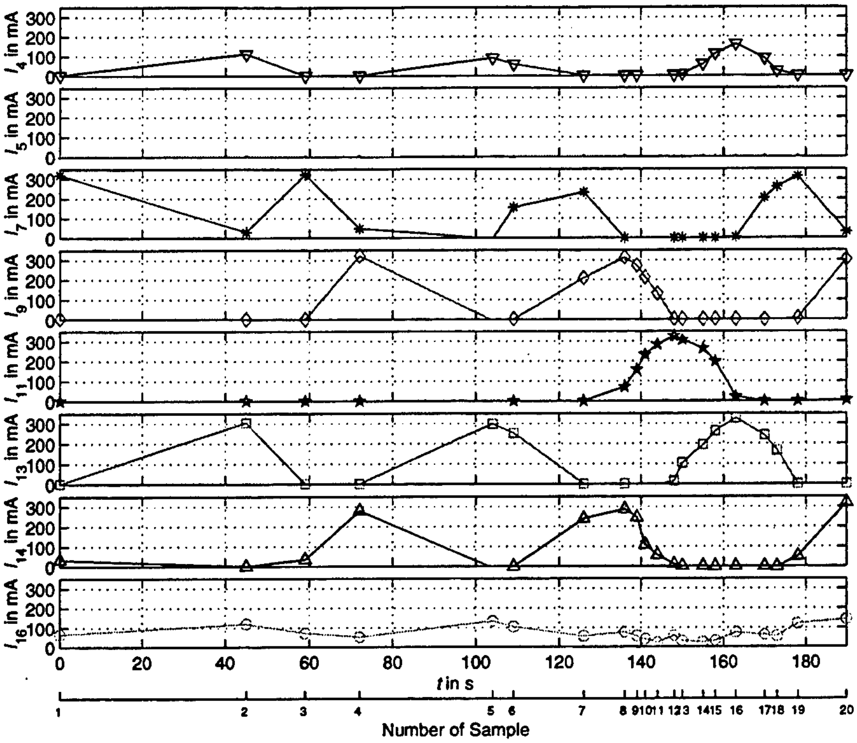


Figure G.13: Solar Panel Currents (i = 3, see Table G.1)

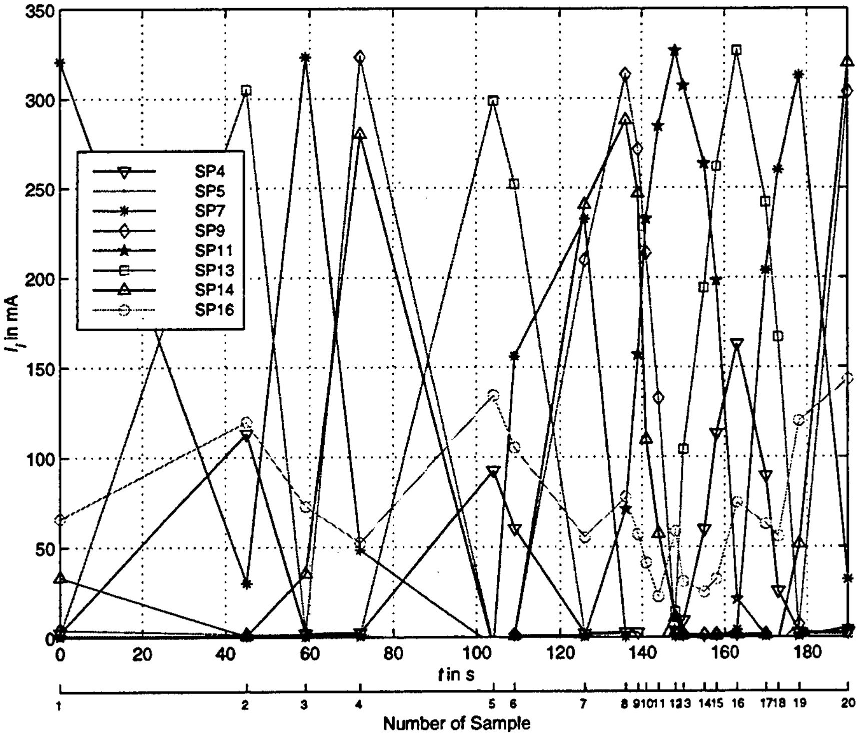


Figure G.14: Solar Panel Currents, in one Plot (i = 3)

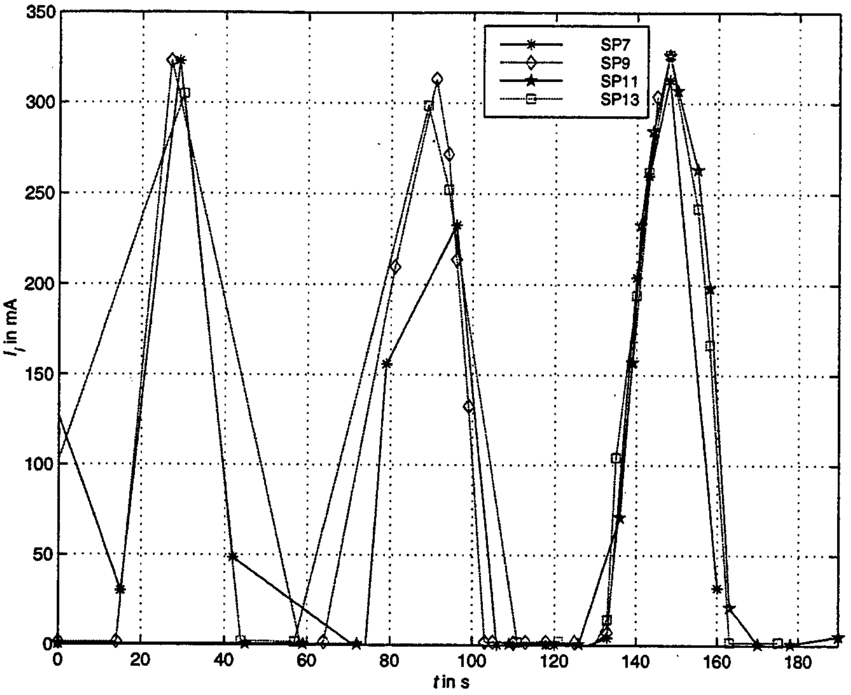


Figure G.15: CV Panel Currents, Phase Shift Suppressed (i = 3)

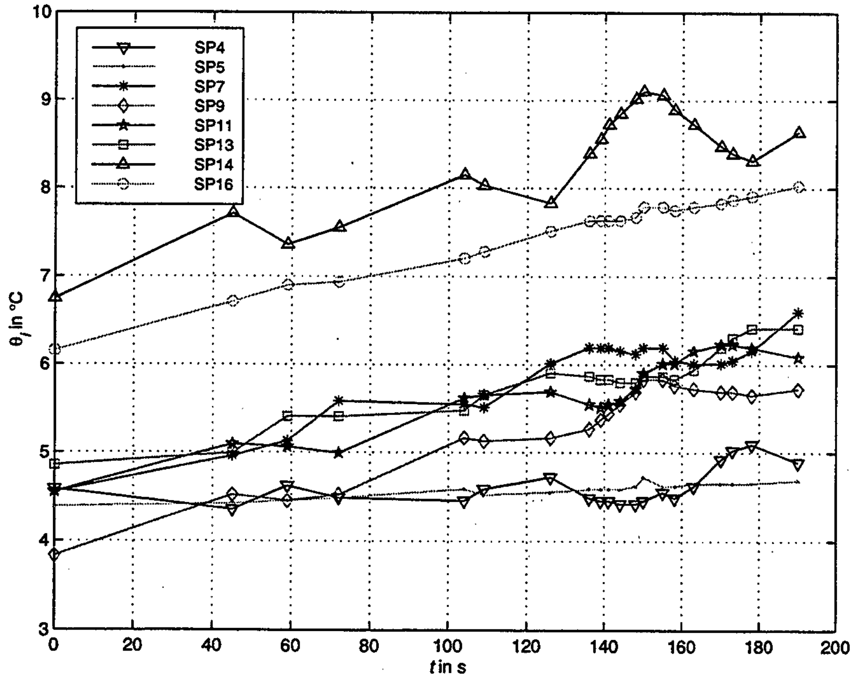


Figure G.16: Temperatures of Panels Equipped with Current Sensors (i = 3)

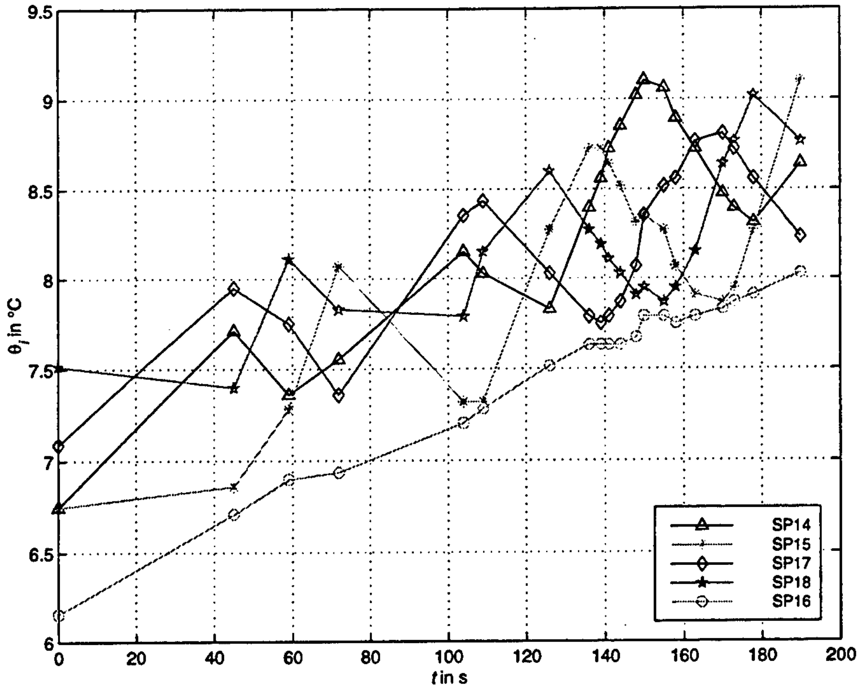


Figure G.17: Temperatures of UDCK and TP panels (i = 3)

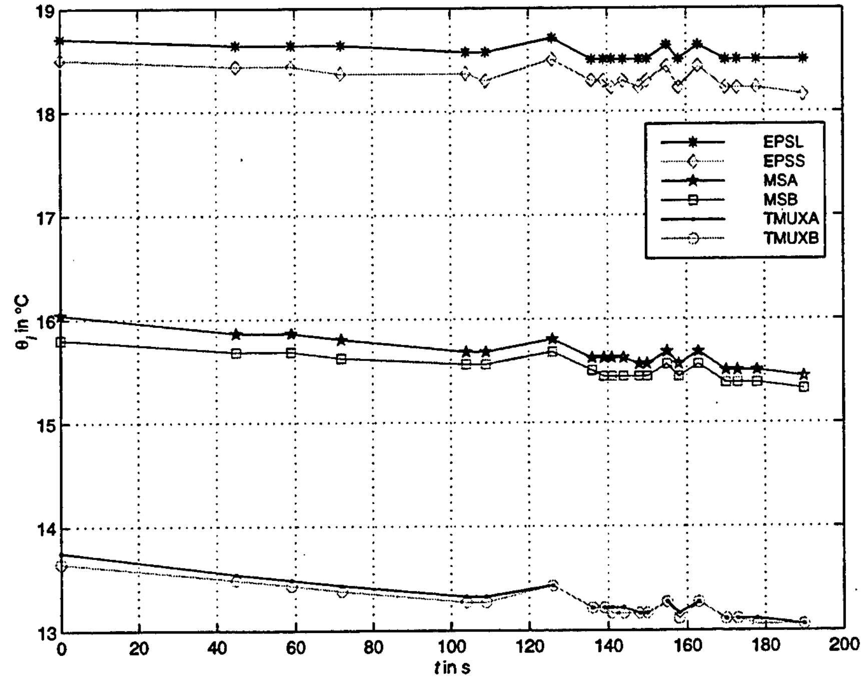


Figure G.18: Temperatures of Components in the Upper Deck (i = 3)

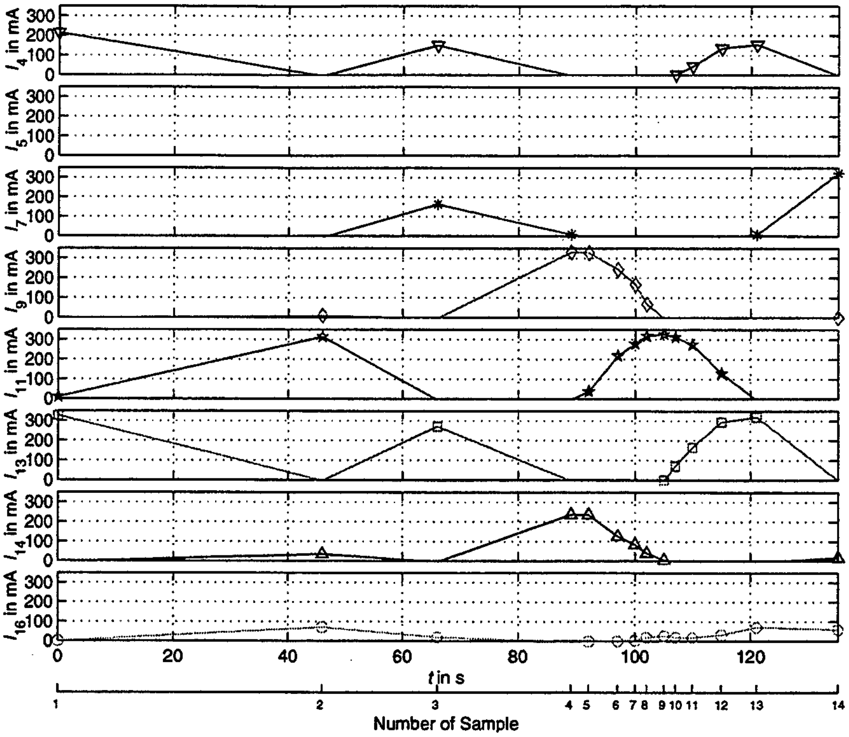


Figure G.19: Solar Panel Currents (i = 4, see Table G.1)

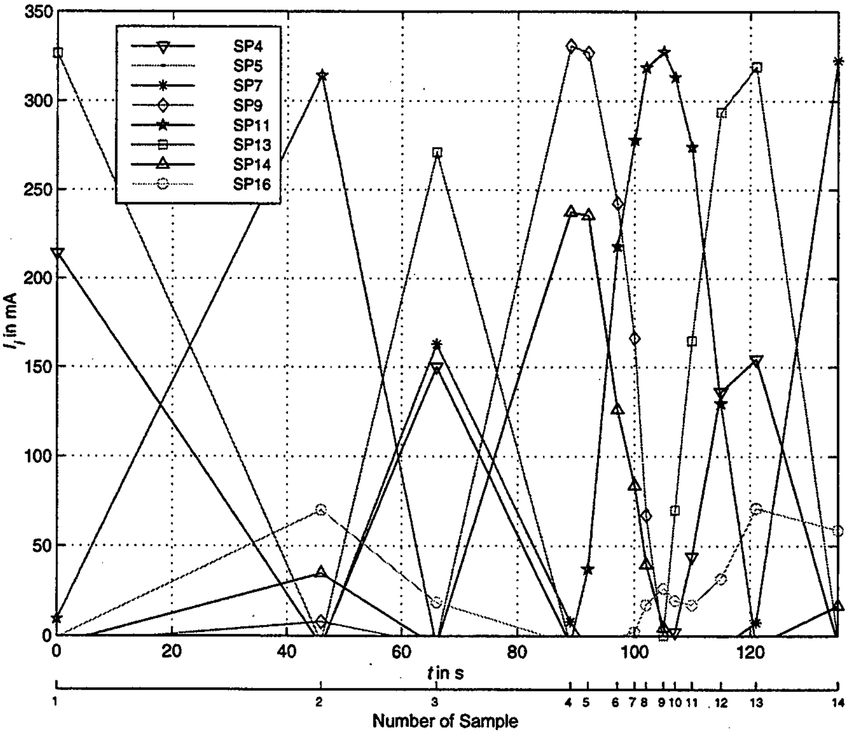


Figure G.20: Solar Panel Currents, in one Plot (i = 4)

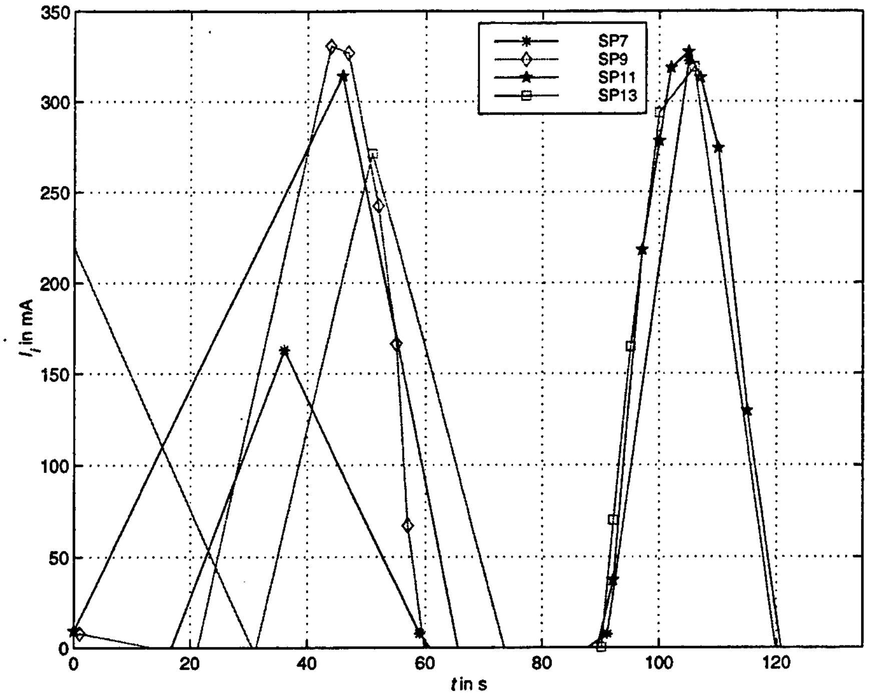


Figure G.21: CV Panel Currents, Phase Shift Suppressed (i = 4)

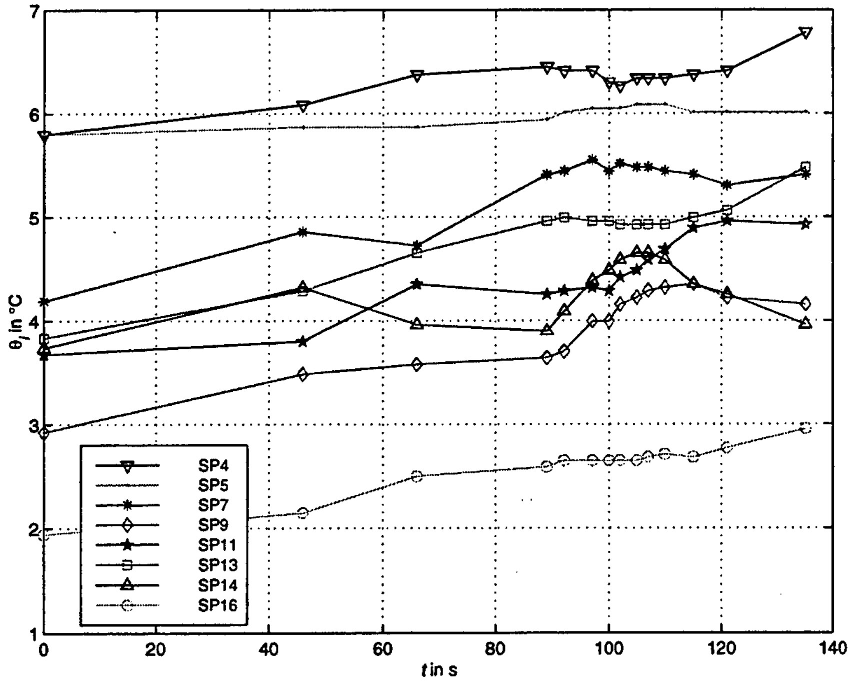


Figure G.22: Temperatures of Panels Equipped with Current Sensors (i = 4)

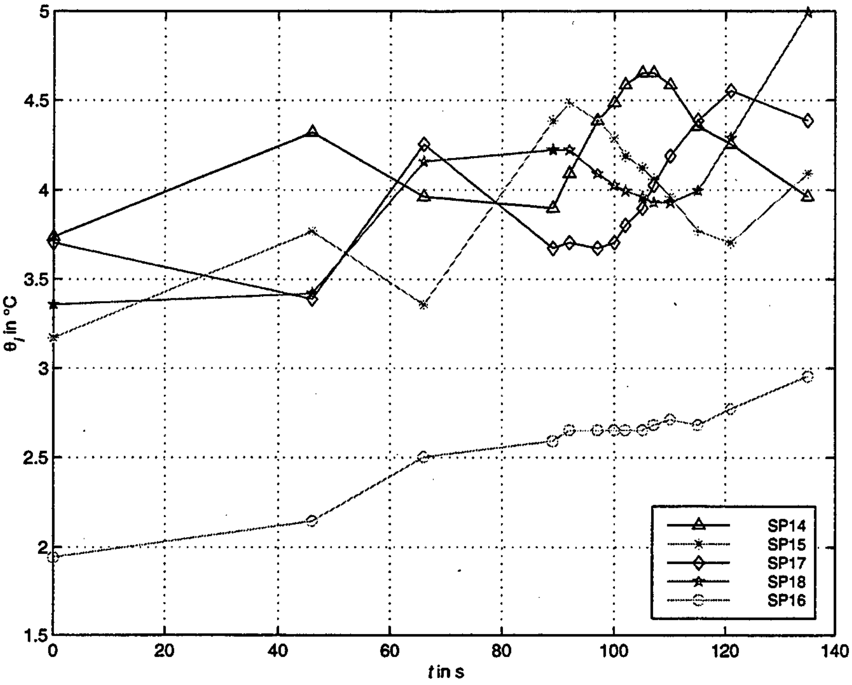


Figure G.23: Temperatures of UDCK and TP panels ($i = 4$)

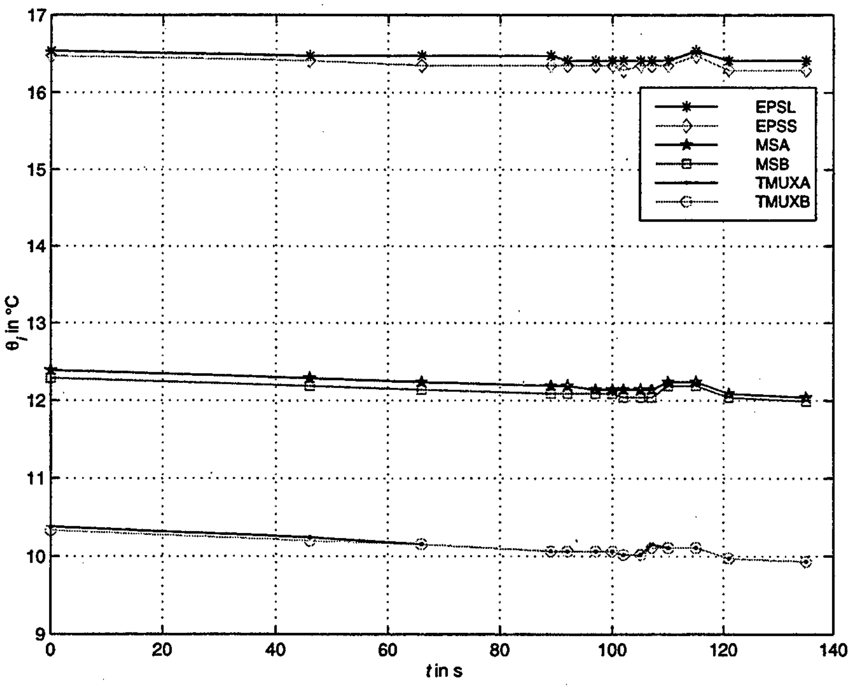


Figure G.24: Temperatures of Components in the Upper Deck ($i = 4$)

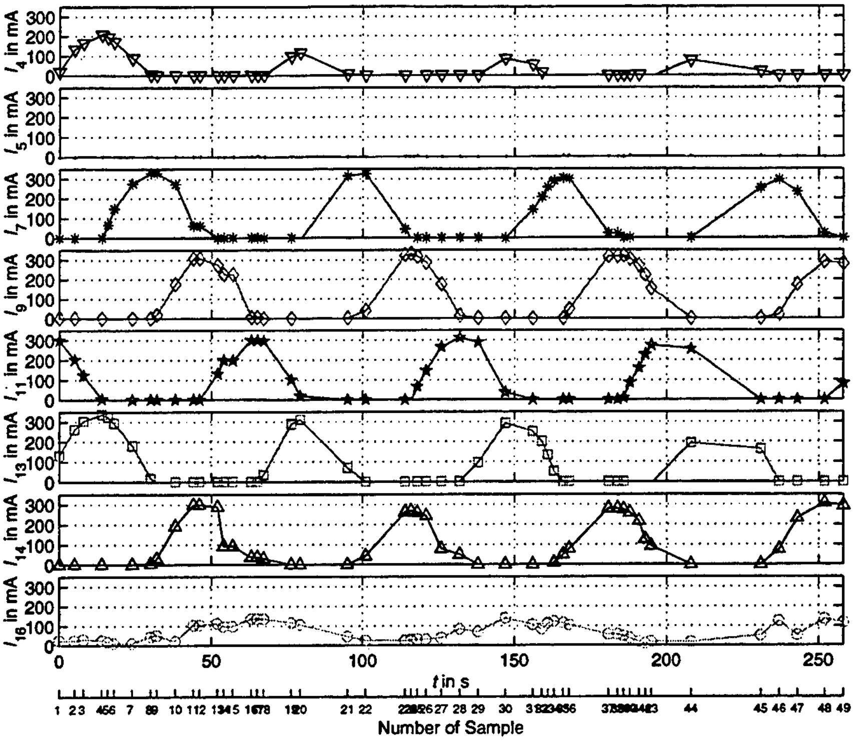


Figure G.25: Solar Panel Currents (i = 5, see Table G.1)

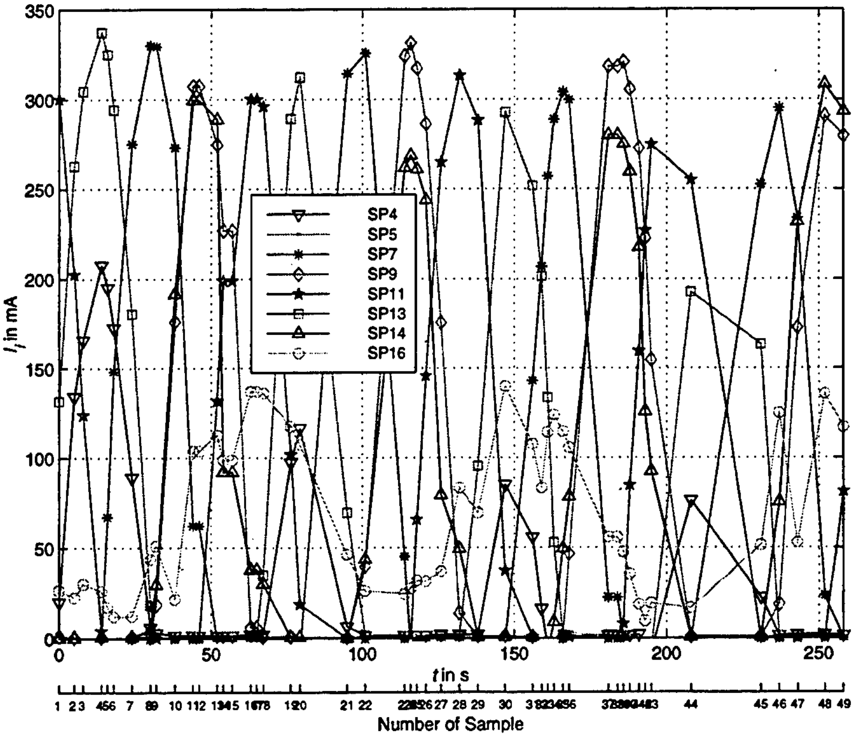


Figure G.26: Solar Panel Currents, in one Plot (i = 5)

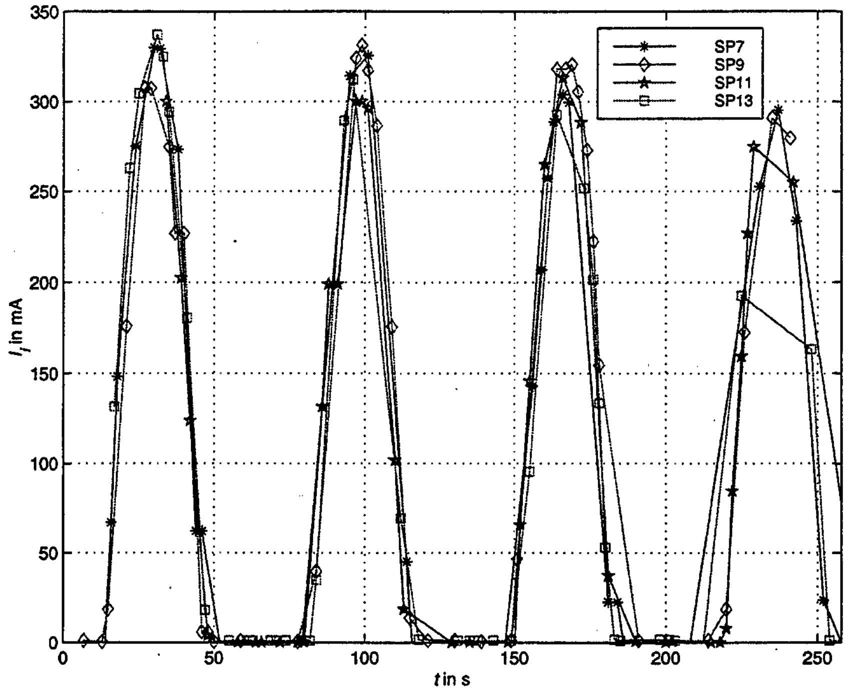


Figure G.27: CV Panel Currents, Phase Shift Suppressed (i = 5)

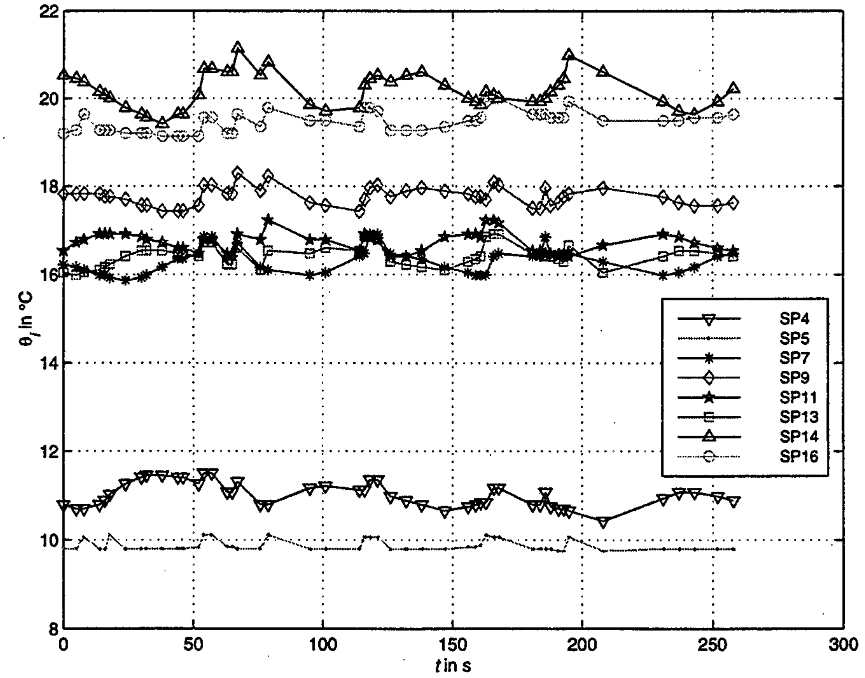


Figure G.28: Temperatures of Panels Equipped with Current Sensors (i = 5)

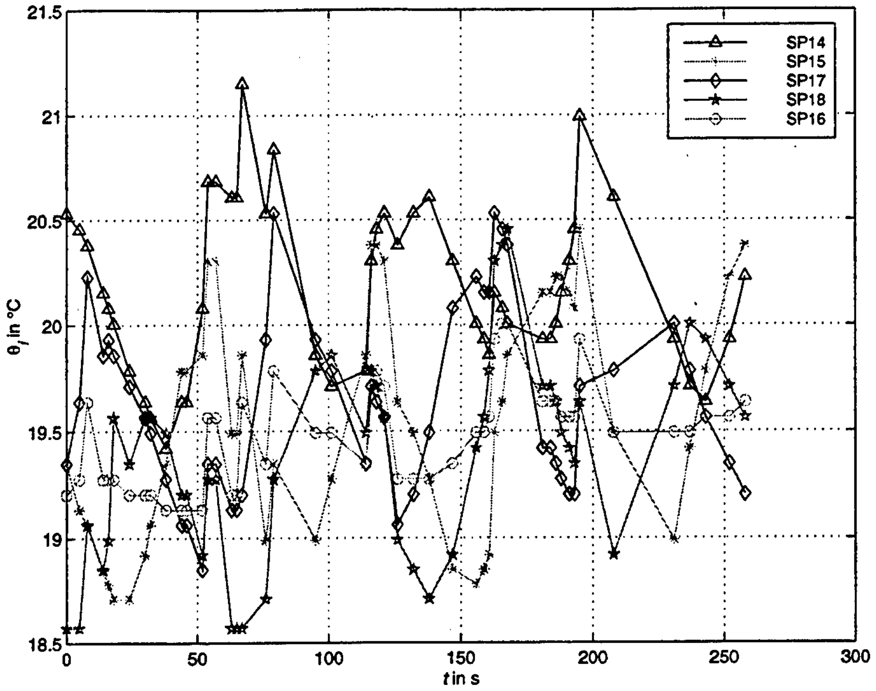


Figure G.29: Temperatures of UDCK and TP panels ($i = 5$)

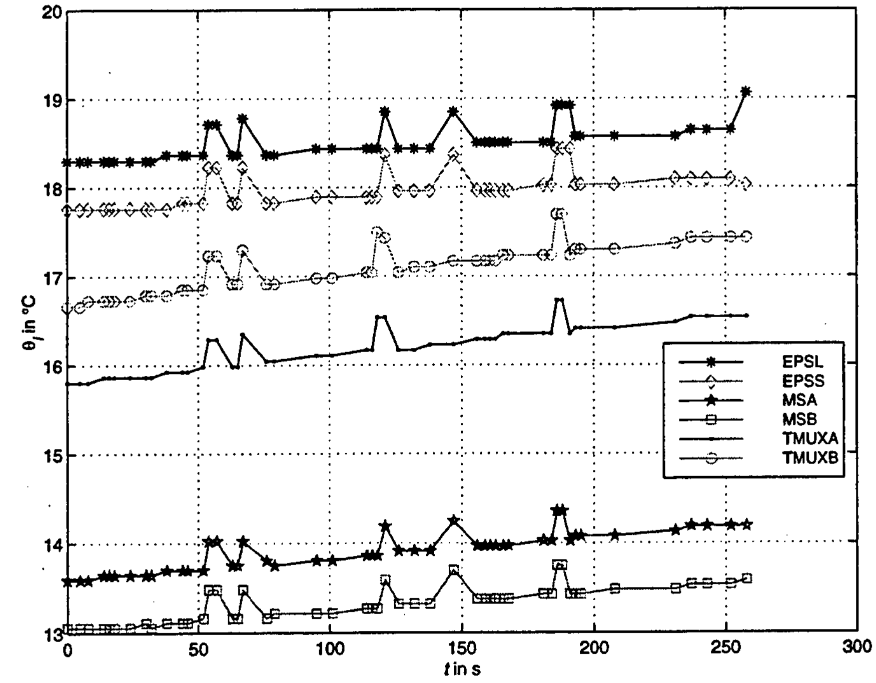


Figure G.30: Temperatures of Components in the Upper Deck ($i = 5$)

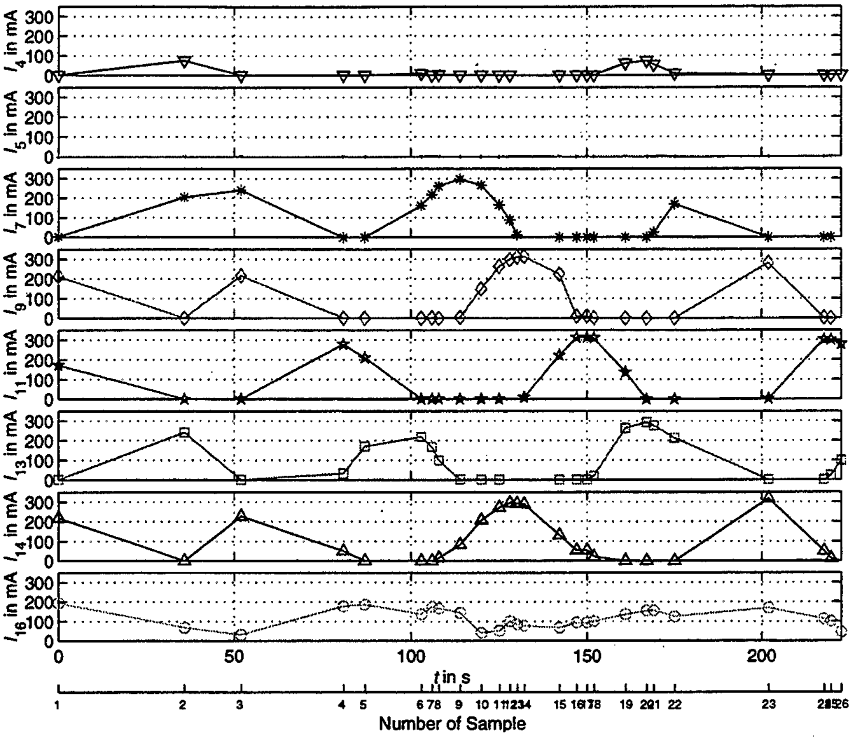


Figure G.31: Solar Panel Currents ($i = 6$, see Table G.1)

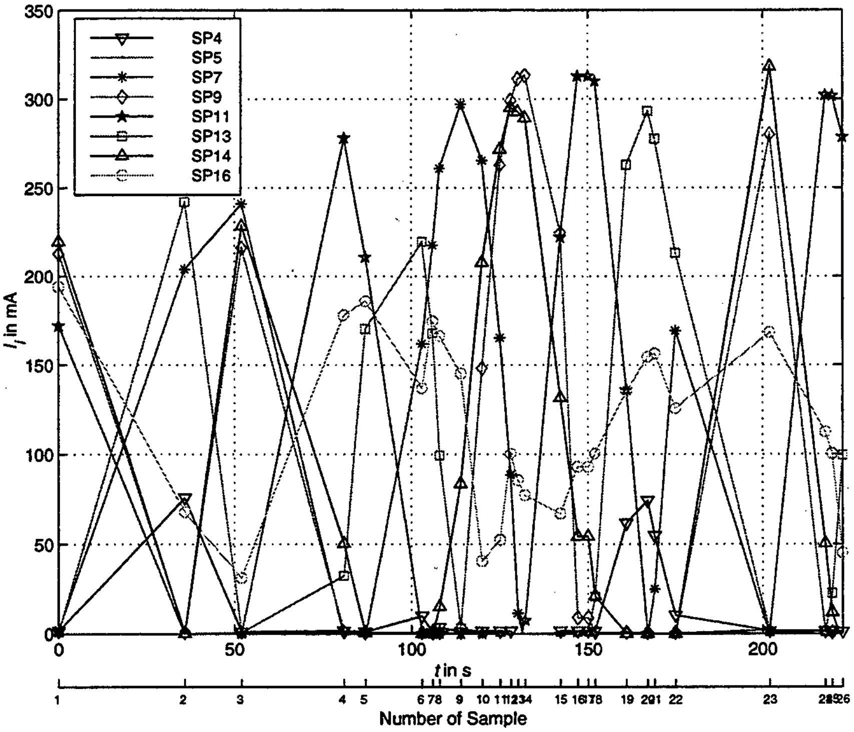


Figure G.32: Solar Panel Currents, in one Plot ($i = 6$)

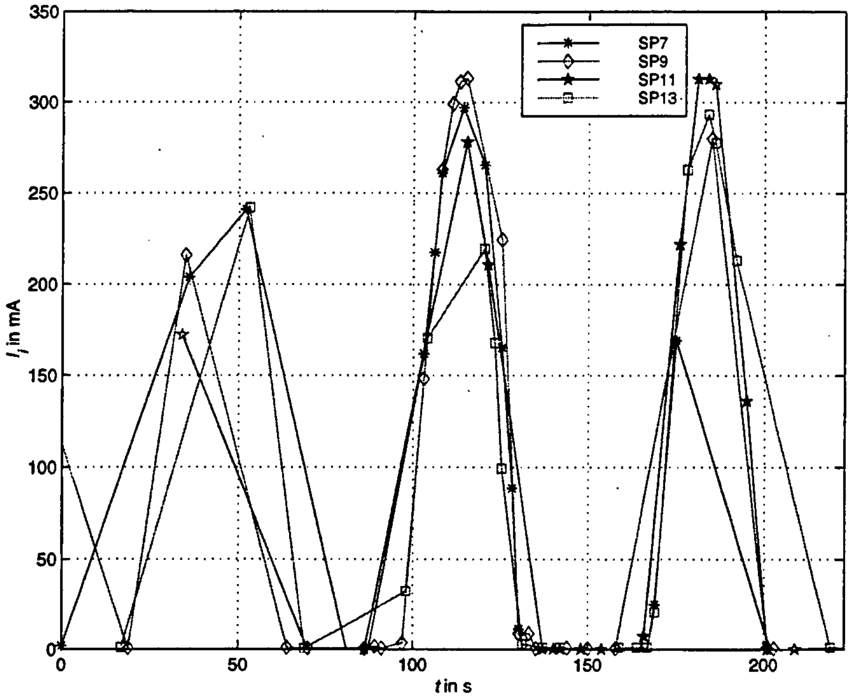


Figure G.33: CV Panel Currents, Phase Shift Suppressed (i = 6)

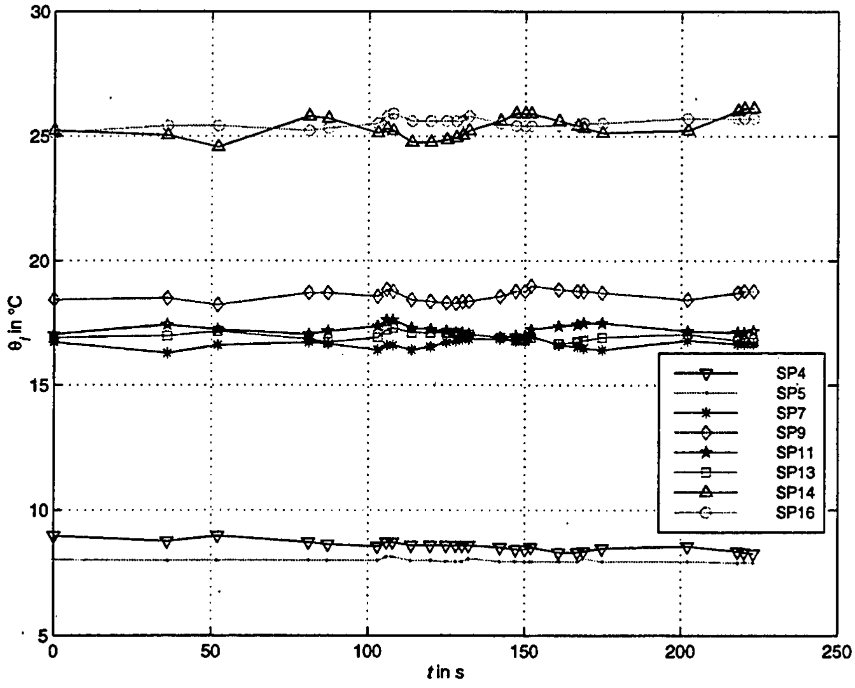


Figure G.34: Temperatures of Panels Equipped with Current Sensors (i = 6)

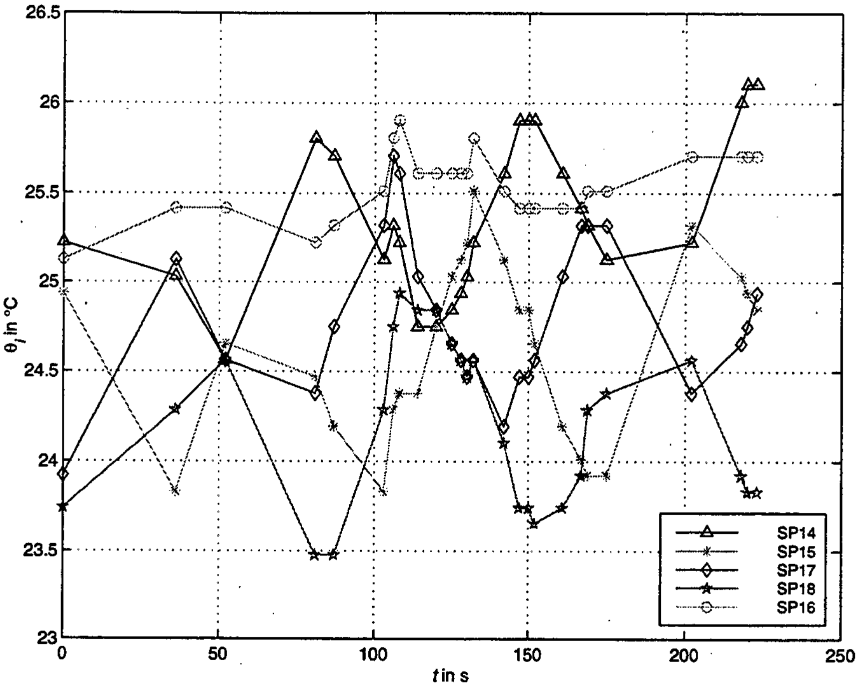


Figure G.35: Temperatures of UDCK and TP panels (*i* = 6)

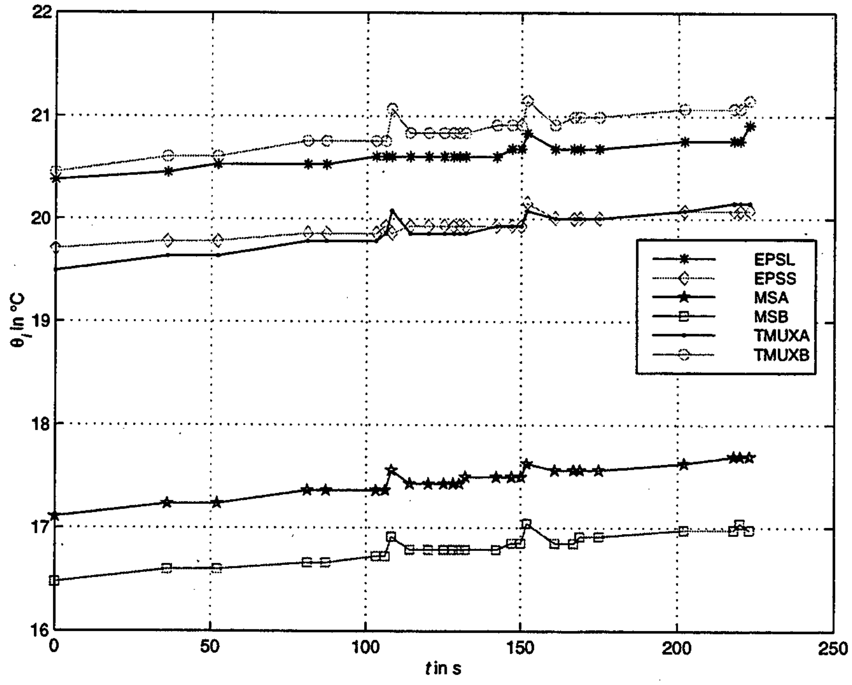


Figure G.36: Temperatures of Components in the Upper Deck (*i* = 6)

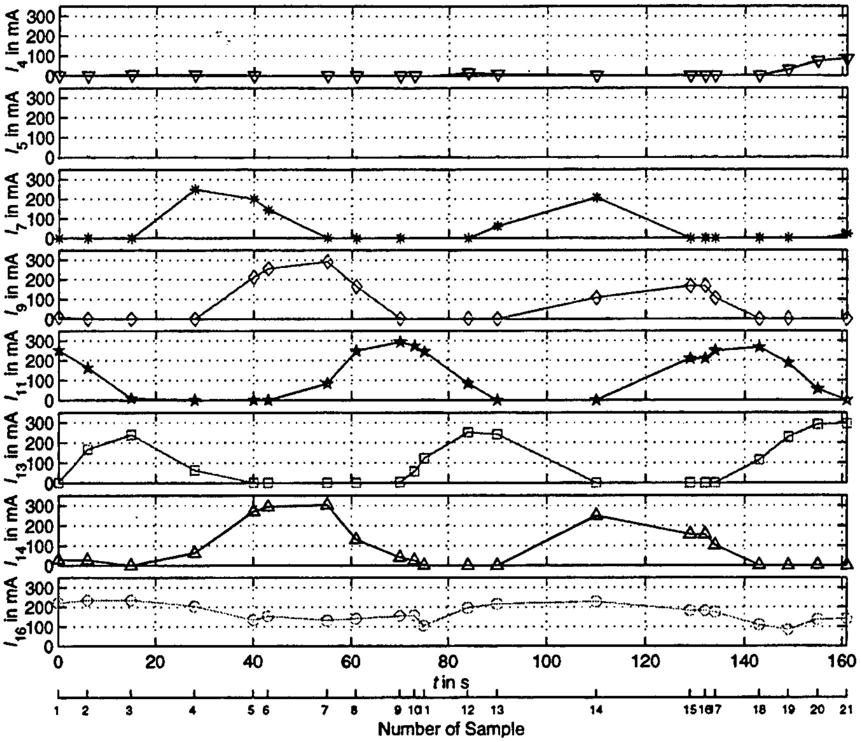


Figure G.37: Solar Panel Currents (i = 7, see Table G.1)

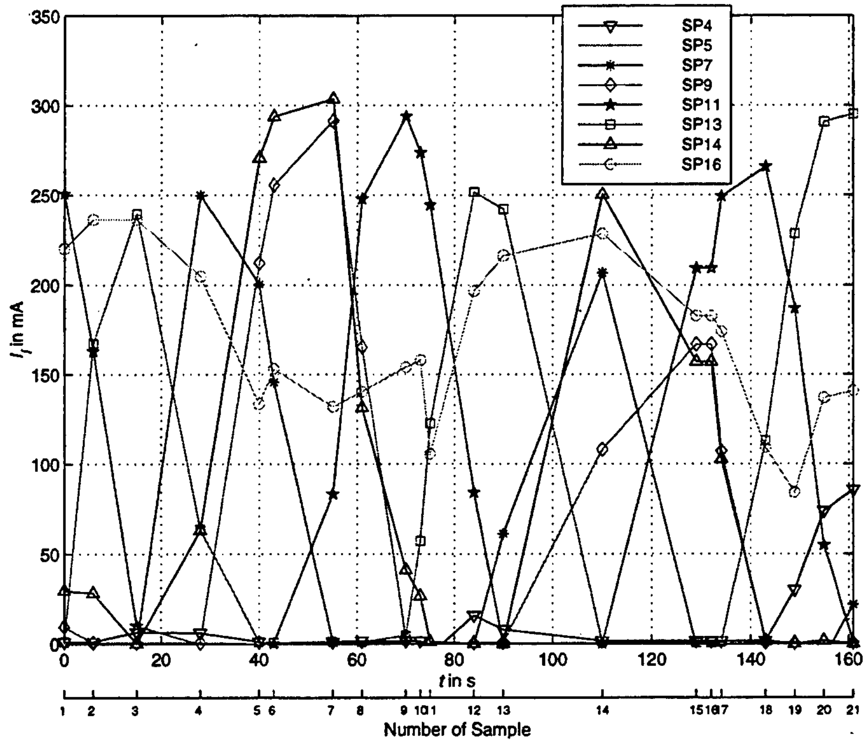


Figure G.38: Solar Panel Currents, in one Plot (i = 7)

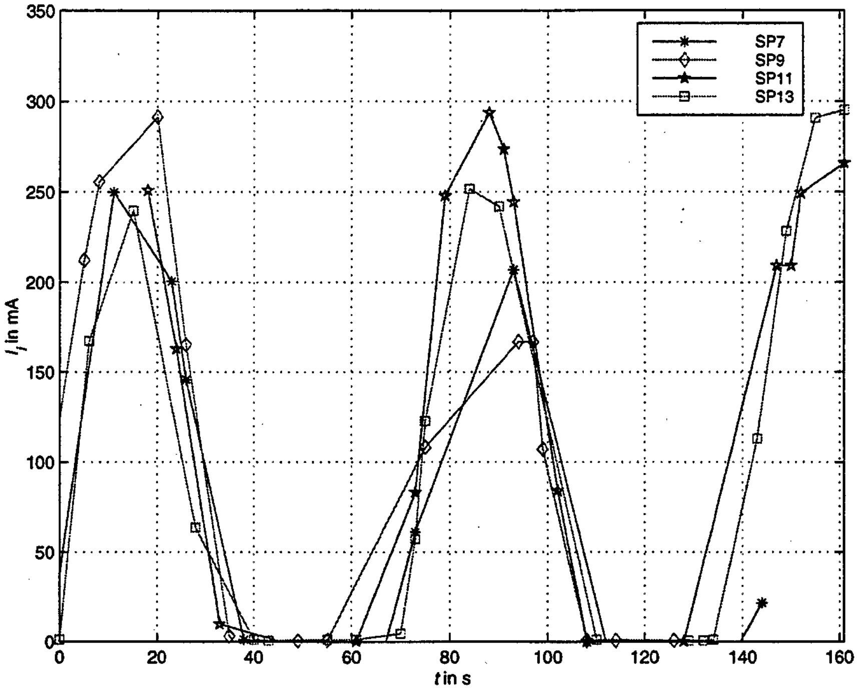


Figure G.39: CV Panel Currents, Phase Shift Suppressed (i = 7)

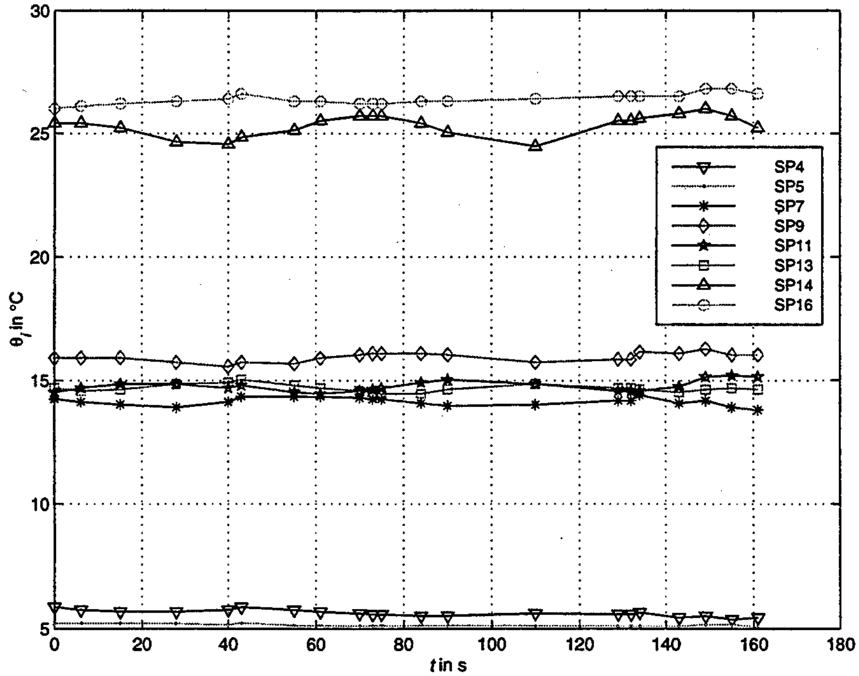


Figure G.40: Temperatures of Panels Equipped with Current Sensors (i = 7)

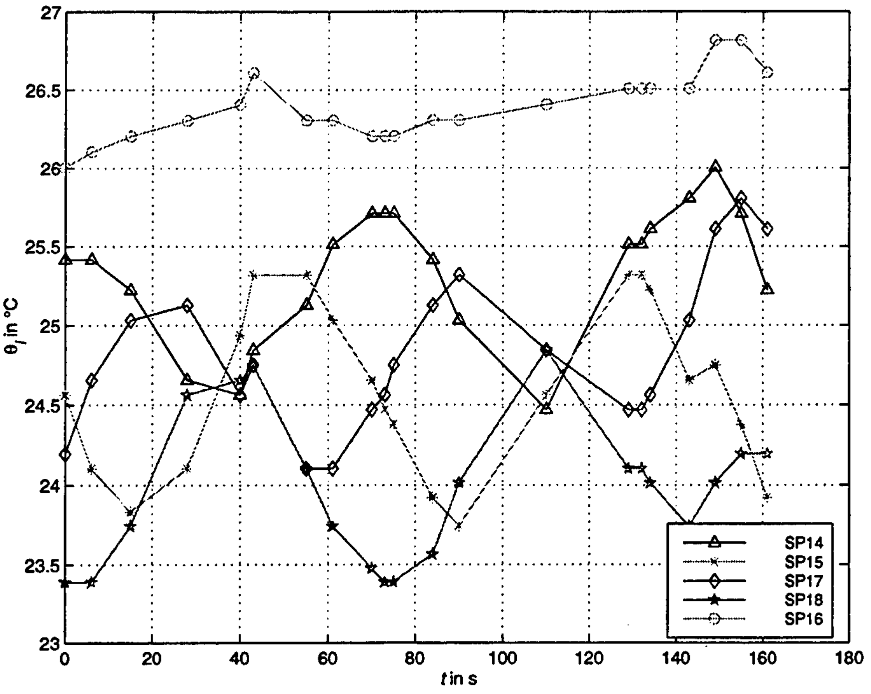


Figure G.41: Temperatures of UDCK and TP panels ($i = 7$)

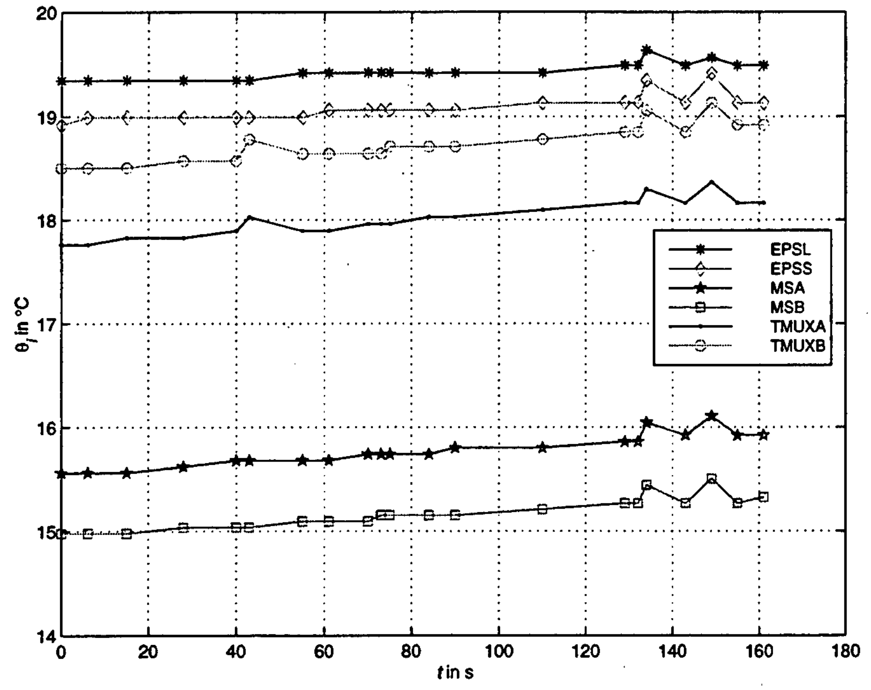


Figure G.42: Temperatures of Components in the Upper Deck ($i = 7$)

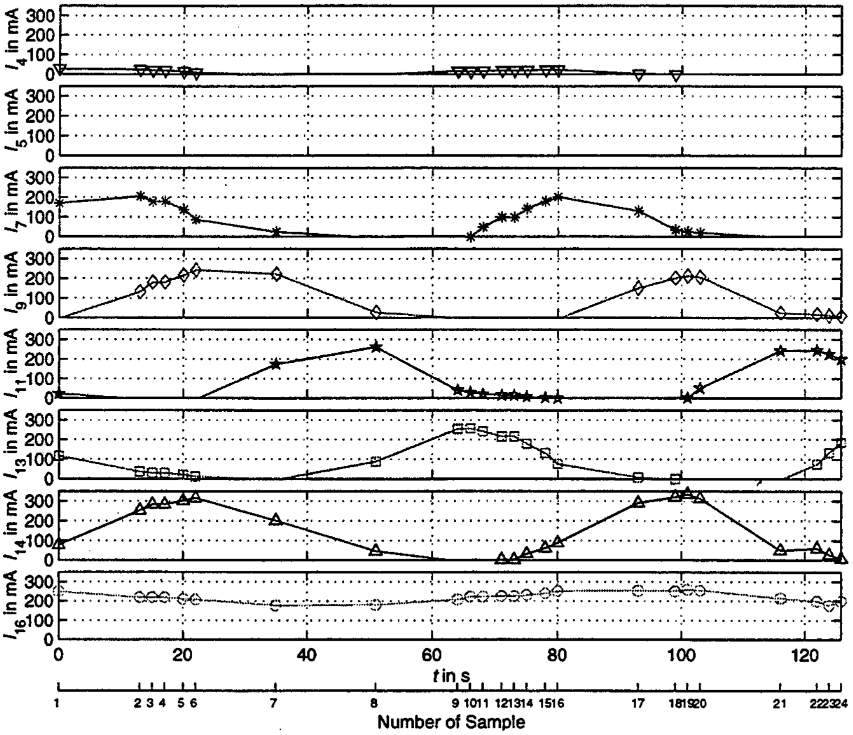


Figure G.43: Solar Panel Currents (i = 8, see Table G.1)

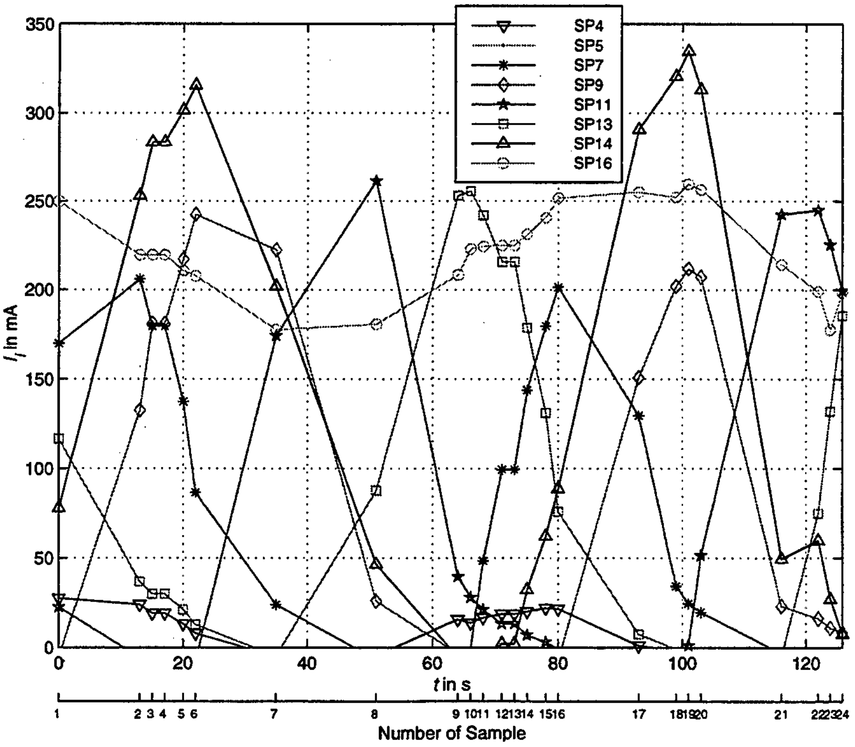


Figure G.44: Solar Panel Currents, in one Plot (i = 8)

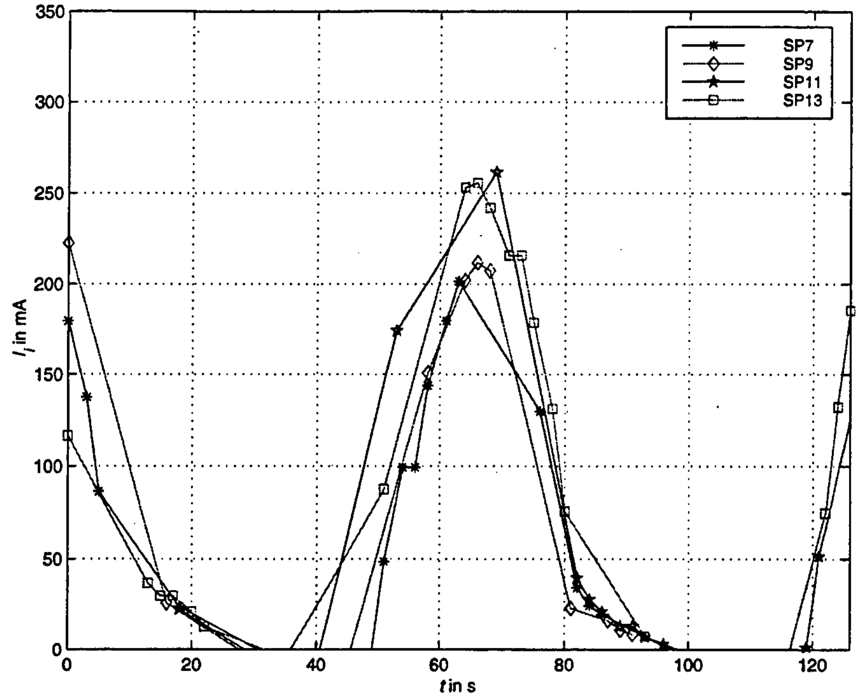


Figure G.45: CV Panel Currents, Phase Shift Suppressed (i = 8)

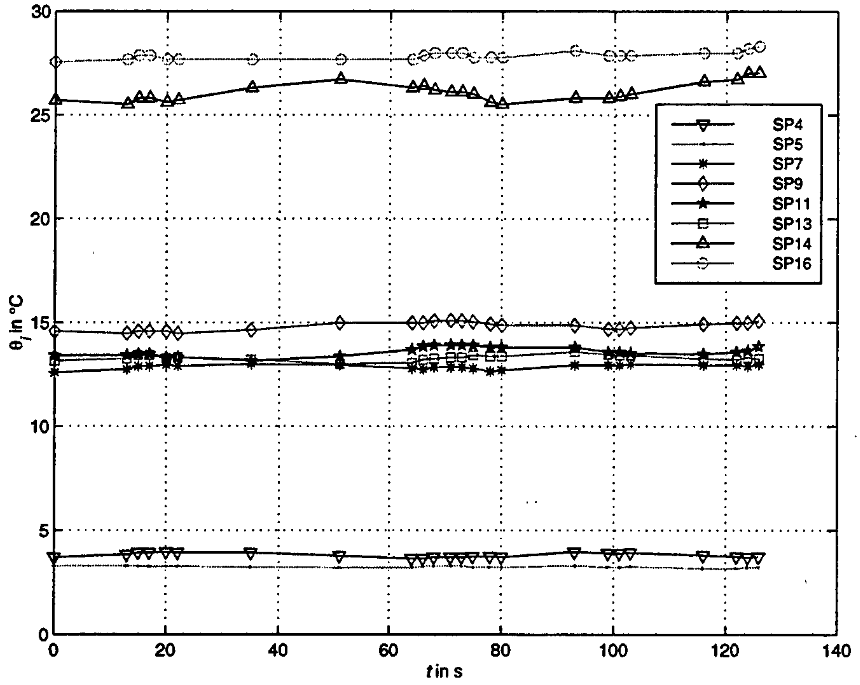


Figure G.46: Temperatures of Panels Equipped with Current Sensors (i = 8)

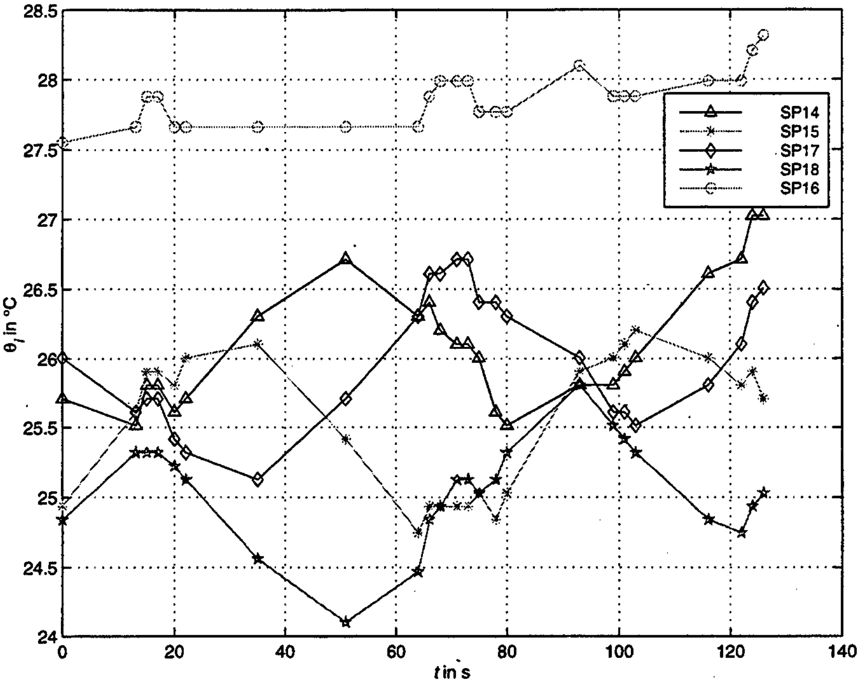


Figure G.47: Temperatures of UDCK and TP panels ($i = 8$)

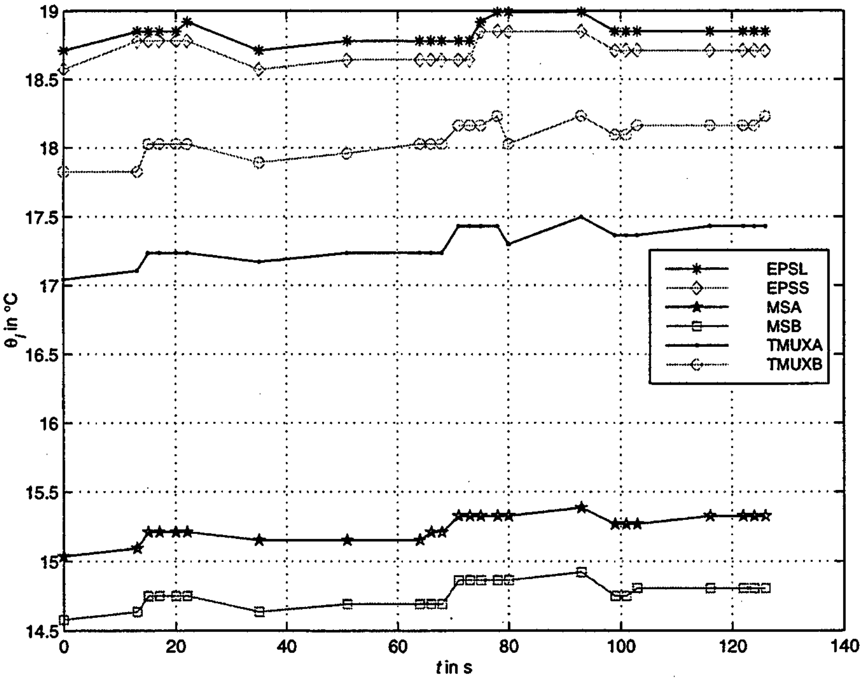


Figure G.48: Temperatures of Components in the Upper Deck ($i = 8$)

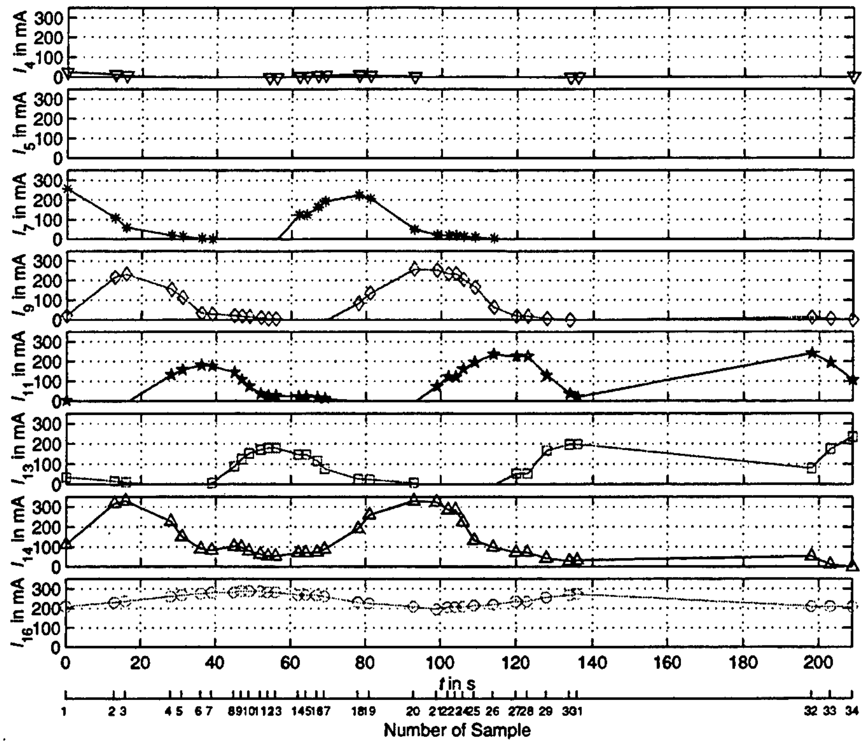


Figure G.49: Solar Panel Currents ($i = 9$, see Table G.1)

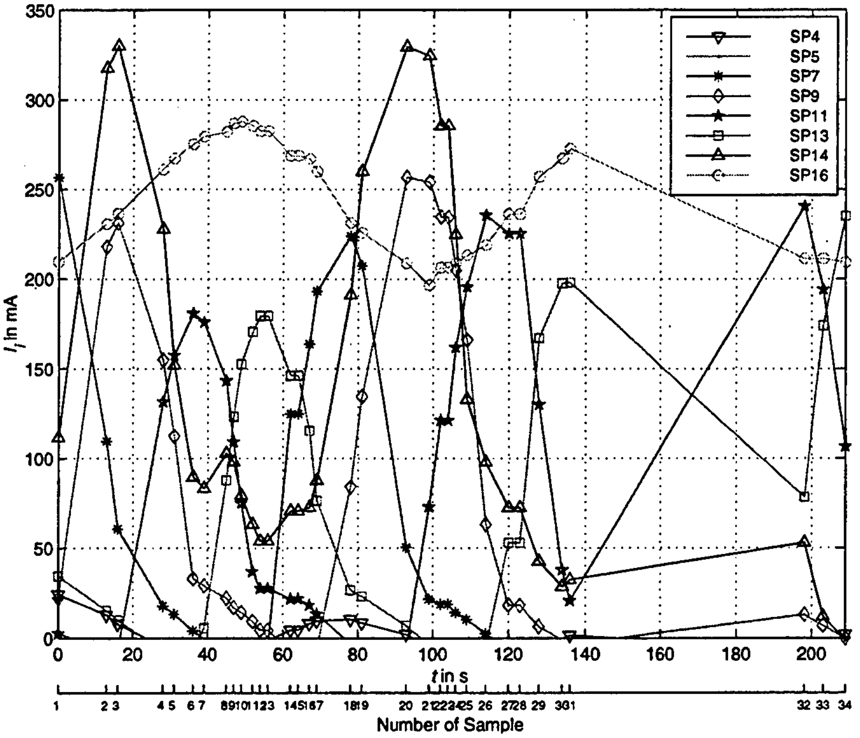


Figure G.50: Solar Panel Currents, in one Plot ($i = 9$)

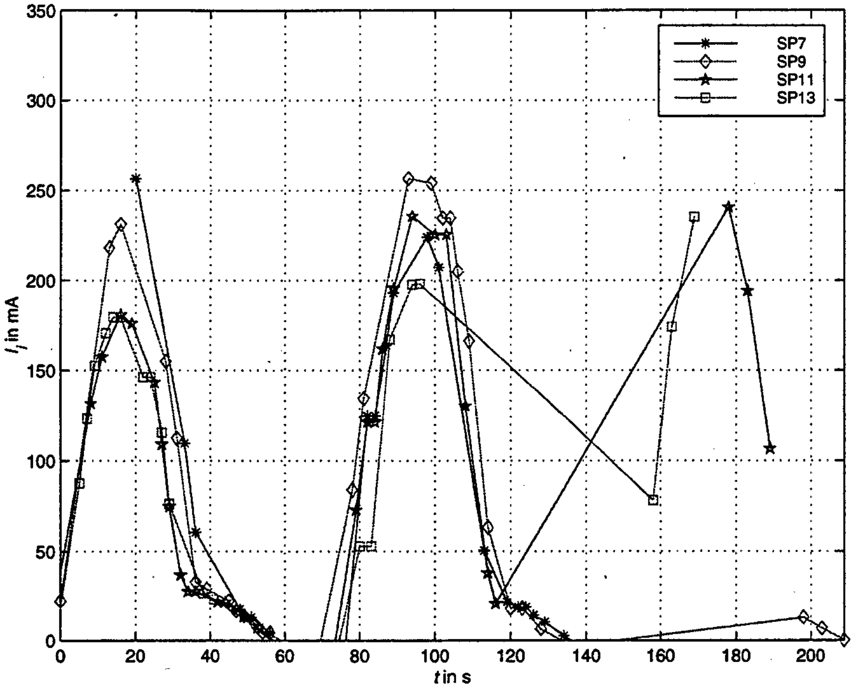


Figure G.51: CV Panel Currents, Phase Shift Suppressed (i = 9)

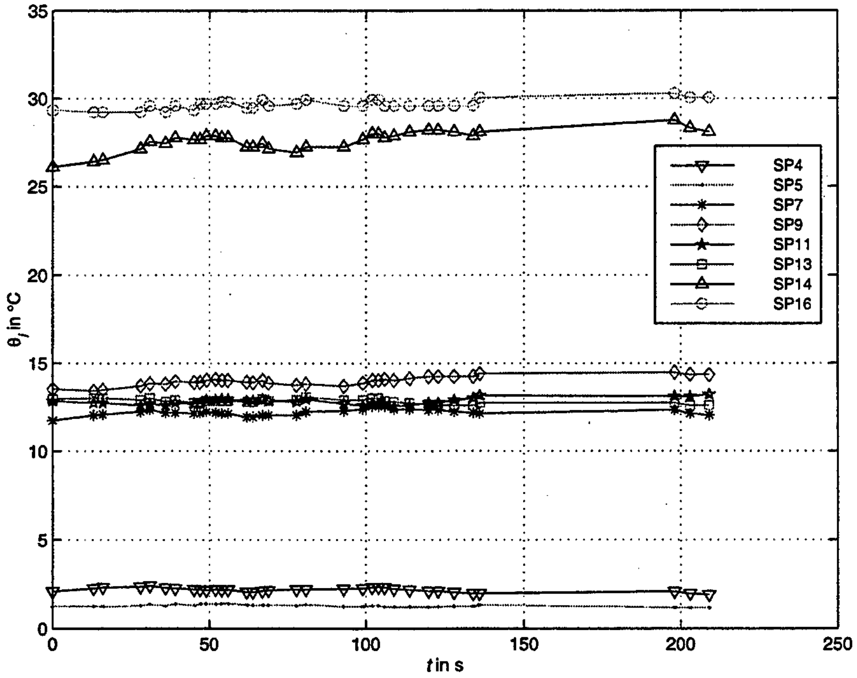


Figure G.52: Temperatures of Panels Equipped with Current Sensors (i = 9)

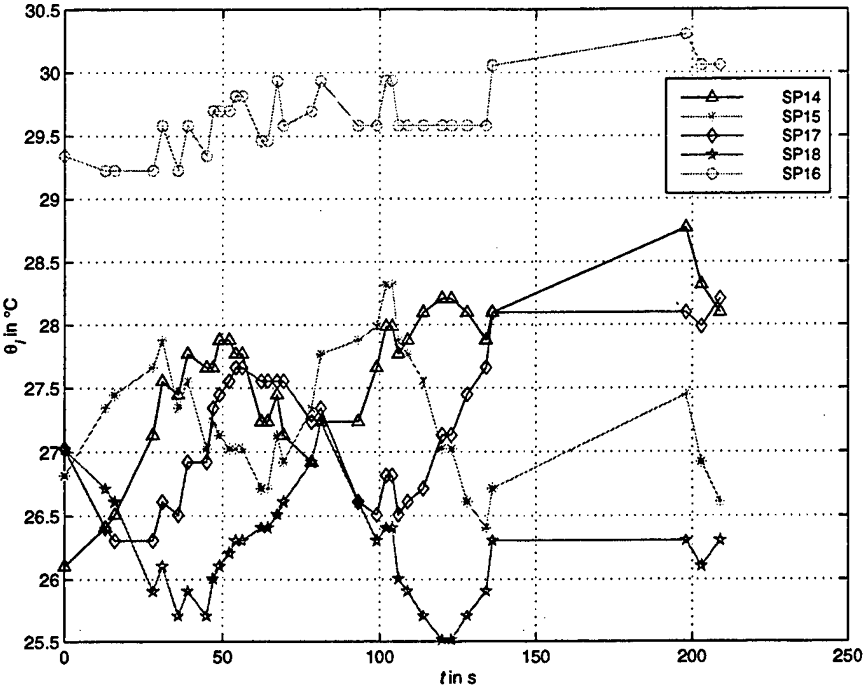


Figure G.53: Temperatures of UDCK and TP panels (i = 9)

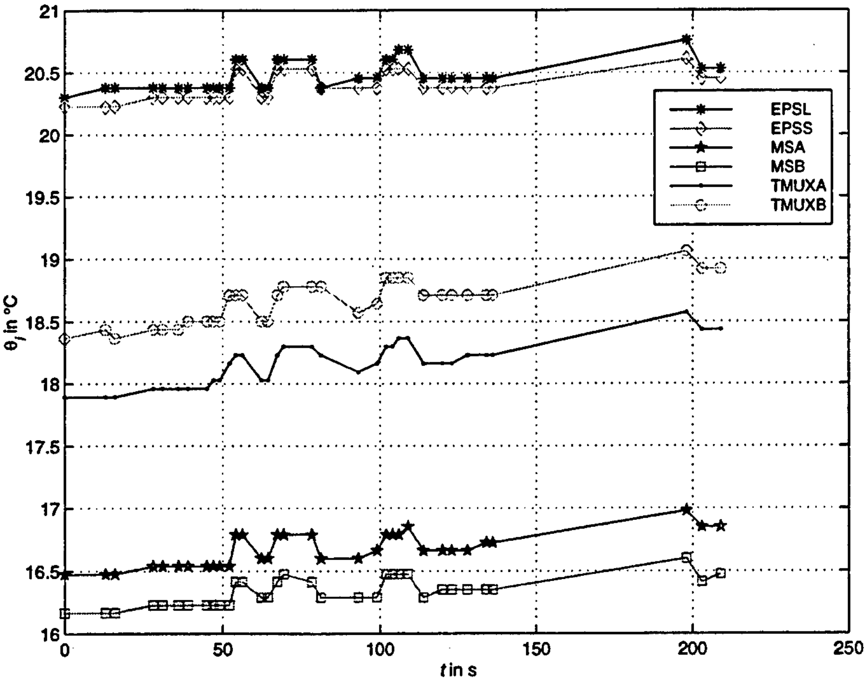


Figure G.54: Temperatures of Components in the Upper Deck (i = 9)

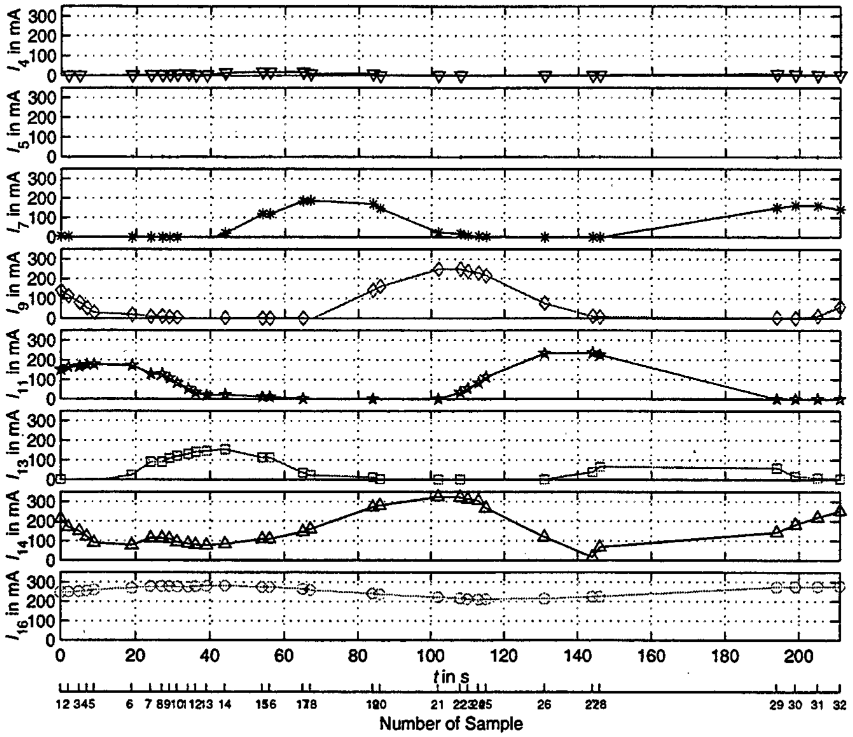


Figure G.55: Solar Panel Currents (i = 10, see Table G.1)

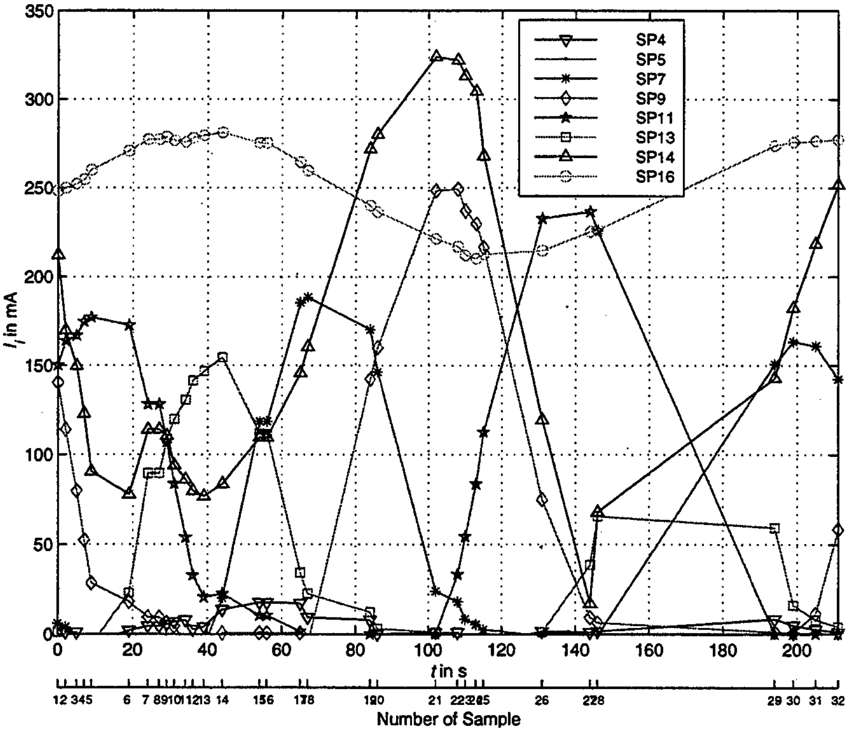


Figure G.56: Solar Panel Currents, in one Plot (i = 10)

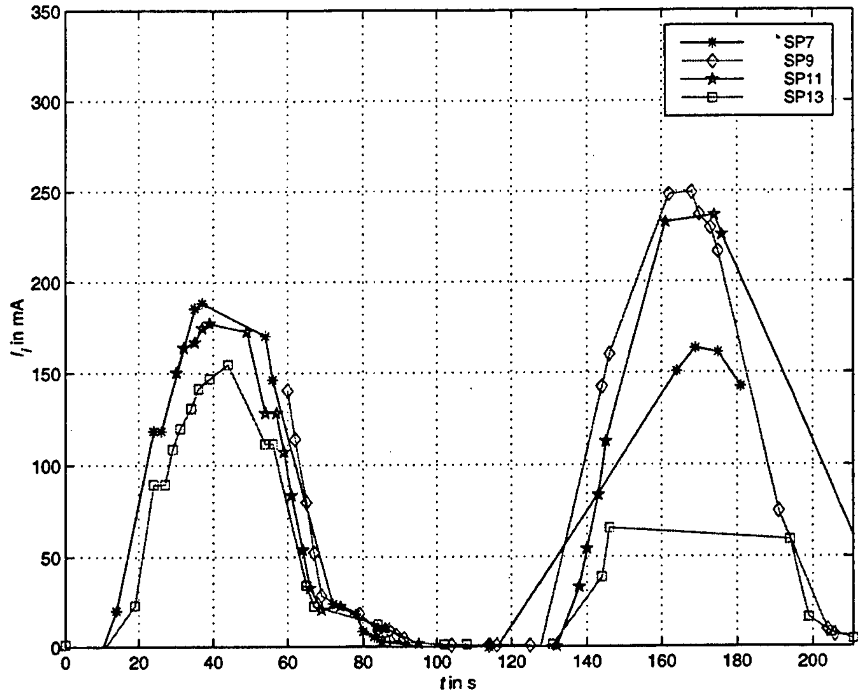


Figure G.57: CV Panel Currents, Phase Shift Suppressed (i = 10)

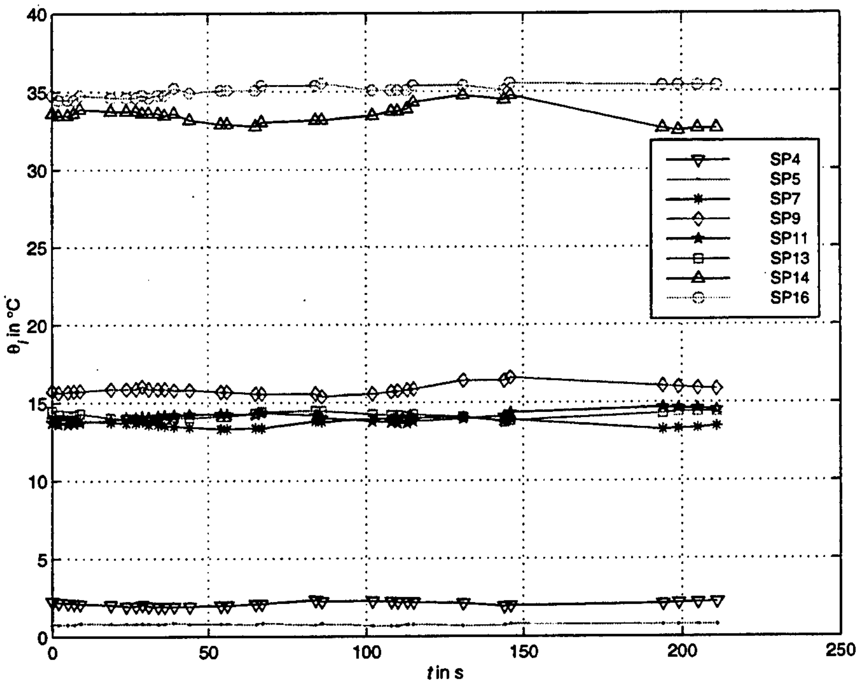


Figure G.58: Temperatures of Panels Equipped with Current Sensors (i = 10)

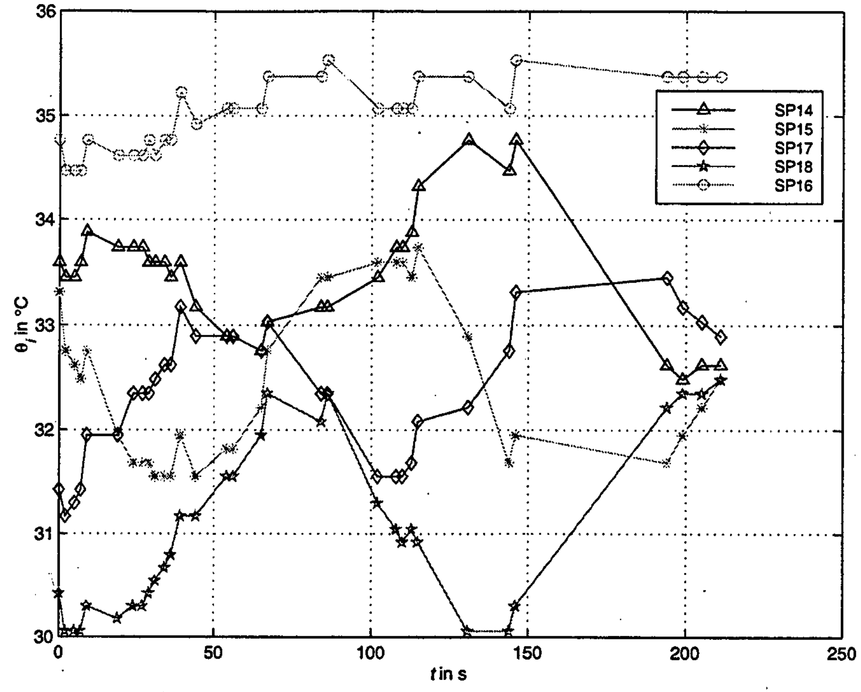


Figure G.59: Temperatures of UDCK and TP panels ($i = 10$)

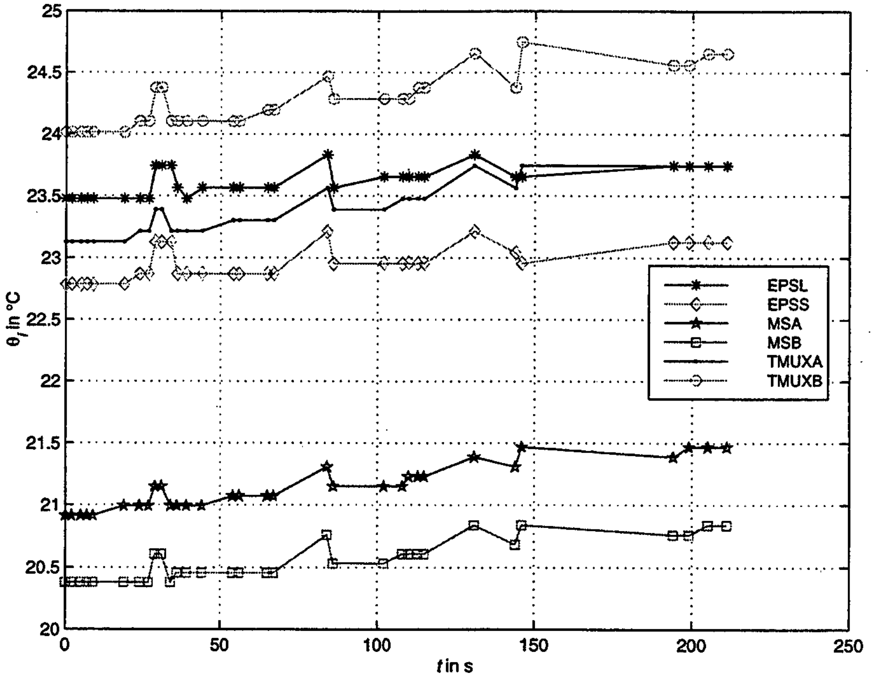


Figure G.60: Temperatures of Components in the Upper Deck ($i = 10$)

[This page is intentionally left blank.]

APPENDIX H. DATA REGARDING CURVE FITTING OF CURRENTS

This appendix contains the data to calculate the estimated solar panel currents with Equation (5.7), see Table H.2. It is determined by a least squares fit described in Chapter V using the weight factors of Table H.1 in Equations (5.5) and (5.8). Furthermore, the plots of the estimated panel currents are provided for each status data of Table G.1.

Table H.1: Weight Factors for Weighted Least Squares Method

Index[32]	*Panel*	*Weight Factors w_i*
1	SP7	0,0,0,0,1, 0,1,1,1,1, 10,10,1,1,1, 0,0,0,0,0, 1,1,1,10,10, 1,0,0
	SP9	1,1,0,0,0, 0,0,0,0,0, 1,1,1,1,1, 10,1,1,0,0, 0,0,0,1,1, 1,1,1
	SP11	1,1,1,1,1, 0,0,0,0,0, 0,0,0,0,0, 1,1,1,1,1, 1,0,0,0,0, 0,1,1
	SP13	0,1,1,1,10, 10,1,1,1,1, 1,1,0,0,0, 0,0,0,1,1, 10,1,1,1,0, 0,0,1
	SP16	1,1,1,1,1, 0,0,0,0,1, 1,1,1,0,1, 1,0,1,1,1, 1,1,1,1,1, 1,1,1
2	SP7	1,1,0,0,0, 1,1,1,10,10, 0,0,0,0,0, 0,0,0,0,0, 1,1,10,10,1, 1,0,0
	SP9	1,1,1,0,0, 0,0,0,0,1, 1,1,1,1,1, 0,0,0,0,0, 0,0,0,1,1, 1,1,1
	SP11	0,0,1,1,0, 0,0,0,0,0, 1,1,1,0,10, 10,0,1,1,0, 0,0,0,0,0, 1,1,1
	SP13	0,0,0,1,1, 1,1,1,1,0, 0,0,0,0,1, 1,1,1,1,0, 1,1,1,0,0, 0,0,0
	SP16	1,1,1,0,1, 1,0,1,1,1, 0,0,0,1,1, 1,0,0,1,0, 0,0,0,1,0, 1,1,1
3	SP7	1,1,3,1,0, 1,1,1,0,0, 0,0,0,0,0, 1,1,1,3,1, 0
	SP9	0,0,0,1,0, 0,1,1,1,1, 1,0,0,0,0, 0,0,0,0,1, 0
	SP11	0,0,0,0,0, 0,0,1,1,1, 1,1,1,1,1, 1,0,0,0,0, 0
	SP13	0,1,0,0,1, 1,0,0,0,0, 0,2,0,1,0, 1,1,1,0,0, 0
	SP16	1,1,1,1,1, 0,0,1,0,0, 0,2,0,0,0, 1,0,0,1,1, 1
4	SP7	0,0,1,1,0, 0,0,0,0,0, 0,0,1,1,0
	SP9	0,1,0,1,1, 1,1,1,1,0, 0,0,0,0,1
	SP11	0,2,0,0,1, 1,1,1,1,1, 1,1,0,0,0
	SP13	1,1,1,0,0, 0,0,0,1,1, 1,1,1,1,0
	SP16	0,1,1,0,0, 1,1,1,1,0, 0,0,1,1,0
5	SP7	0,0,0,1,1, 1,1,1,1,1, 1,1,0,0,0, 0,0,0;0,0, 1,1,1,0,0, 0,0,0,0,0, 1,1,1,1,1, 1,1,1,0,0, 0,0,0,0,1, 1,1,1,0,0
	SP9	0,0,0,0,0, 0,0,0,1,1, 1,1,1,1,1, 1,0,0,0,0, 0,1,1,1,1, 1,1,1,0,0, 0,0,0,0,1, 1,1,1,1,1, 1,1,1,0,0, 1,1,1,1,1
	SP11	1,1,1,1,0, 0,0,0,0,0, 0,1,1,1,1, 1,1,1,1,1, 0,0,0,0,1, 1,1,1,1,1, 0,0,0,0,0, 0,0,0,1,1, 1,1,1,1,0, 0,0,1,1,1
	SP13	1,1,1,1,1, 1,1,1,0,0, 0,0,0,0,0, 0,1,1,1,1, 1,1,0,0,0, 0,0,1,1,1, 1,1,1,1,1, 0,0,0,0,0, 0,0,0,1,1, 1,0,0,0,0
	SP16	0,0,1,1,0, 0,0,1,1,0, 1,1,0,0,0, 10,1,1,1,1, 1,1,1,1,1, 1,1,1,0,10, 0,0,0,1,1, 1,1,1,1,1, 0,0,1,1,1, 10,0,1,1,1
6	SP7	0,1,1,0,0, 1,1,1,1,1, 1,1,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0
	SP9	1,0,1,1,0, 0,0,0,1,1, 1,1,1,1,1, 0,1,0,0,0, 0,0,1,0,0, 0,0
	SP11	1,0,0,1,1, 0,0,0,0,0, 0,0,0,1,1, 1,10,1,1,0, 0,0,1,1,1, 1,1
	SP13	0,1,1,1,1, 1,1,1,1,0, 0,0,0,0,0, 0,1,1,1,10, 1,1,0,1,1, 1,1
	SP16	1,0,0,1,1, 0,1,1,1,0, 0,1,0,0,0, 1,1,1,1,1, 1,0,1,1,1, 0,0

[32] The index is referred to Table G.1 that gives the timestamp for the snapshot data.

7	SP7	0,0,0,1,1, 1,0,0,0,0, 0,0,1,1,0, 0,0,0,0,0, 1,1
	SP9	0,0,0,0,1, 1,1,10,0,0, 0,0,0,1,1, 1,1,0,0,0, 0,0
	SP11	1,1,1,0,0, 0,1,1,1,1, 1,1,0,0,1, 1,1,1,1,1, 0,0
	SP13	1,1,1,1,0, 1,1,1,1,1, 0,1,1,1,1, 1,1,0,0,1, 1,1
	SP16	1,1,1,1,0, 1,1,1,1,1, 0,1,1,1,1, 1,1,0,0,1, 1,0
8	SP7	1,1,1,1,1, 1,0,0,0,0, 1,1,1,1,1, 1,1,0,0,0, 0,0,0,0,0
	SP9	0,1,1,1,1, 1,1,0,0,0, 0,0,0,0,0, 0,1,1,1,1, 1,0,0,0,0
	SP11	0,0,0,0,0, 0,1,1,0,0, 0,0,0,0,0, 0,0,0,0,0, 1,1,1,1,0
	SP13	0,0,0,0,0, 0,0,1,1,1, 1,1,1,1,1, 1,0,0,0,0, 0,1,1,1,1
	SP16	1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1,1,0,1,0
9	SP7	1,1,1,0,0, 0,0,0,0,0, 0,0,0,1,0, 0,1,10,1,1, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0
	SP9	1,1,10,1,1, 1,0,0,0,0, 0,0,0,0,0, 0,0,1,1,1, 1,1,0,1,1, 1,0,0,0,0, 0,0,0,0,0
	SP11	0,0,0,1,1, 1,1,1,1,1, 1,0,0,0,0, 0,0,0,0,0, 1,1,1,1,1, 1,0,1,1,1, 1,10,1,1,1
	SP13	0,0,0,0,0, 0,1,1,1,1, 1,1,1,0,0, 1,1,0,0,0, 0,0,0,0,0, 0,1,1,1,1, 1,1,1,1,1
	SP16	1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 0,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1
10	SP7	0,0,0,0,0, 0,0,0,0,0, 0,0,0,1,1, 0,1,1,1,1, 1,0,0,0,0, 0,0,0,1,10, 1,1,0
	SP9	1,1,1,1,1, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,1,1, 1,1,1,1,1, 1,0,0,0,0, 0,0,0
	SP11	1,1,1,1,1, 10,0,1,1,1, 1,1,1,0,0, 0,0,0,0,0, 0,1,1,1,1, 1,1,1,0,0, 0,0,1
	SP13	0,0,0,0,0, 1,1,1,1,1, 1,1,1,10,1, 1,1,1,0,0, 0,0,0,0,0, 0,1,1,1,1, 0,0,0
	SP16	1,1,1,1,1, 1,0,0,0,0, 0,0,1,1,1, 1,1,1,1,1, 1,0,0,0,1, 1,1,1,1,1, 1,1,1

Table H.2: Parameters for Estimated Sine Function

Index	*Panel*	*T*	a_0	b_0	c_0	d_0	e_0	f_0
1	SP7	88	0.0379	0.2287	-1.9626	0.0020	0.0135	-7.2803
	SP9	87	0.0191	0.2669	-3.5644	-0.0031	0.0080	-7.2803
	SP11	88	-0.0211	-0.2824	-8.2818	0.0033	0.0113	-7.2803
	SP13	87	0.0306	0.2387	-0.4164	-0.0058	0.0167	-7.2803
	SP16	116	0.1949	0.0322	-7.2803	0	0	0
2	SP7	59	-0.0257	0.3140	2.7382	-0.0044	-0.0082	-2.6115
	SP9	59	-0.0328	-0.3277	-1.9363	0.0029	-0.0179	-2.6115
	SP11	59	-0.0031	-0.2947	-3.5570	-0.0021	-0.0030	-2.6115
	SP13	60	-0.0270	0.2700	-1.7698	0.0541	-0.0051	-2.6115
	SP16	79	0.1511	0.0349	-2.6115	0	0	0
3	SP7	59	0.0149	-0.2949	-7.9227	-0.0001	-0.0106	-7.1391
	SP9	60	-0.0471	-0.3734	-3.0005	0.0004	0.0005	-7.1391
	SP11	59	0.0070	0.3174	-7.9994	-0.0008	-0.0021	-7.1391
	SP13	59	0.0263	-0.2918	-6.3863	-0.0098	-0.0107	-7.1391
	SP16	80	0.0983	0.0436	-7.1391	0	0	0
4	SP7	58	-0.0468	-0.3963	2.7894	-0.0045	-0.0159	-8.0386
	SP9	58	-0.0469	0.3528	-8.2841	0.0089	-0.0138	-8.0386
	SP11	59	-0.0352	0.3663	-3.2968	-0.0047	-0.0064	-8.0386
	SP13	59	0.0055	-0.3210	-1.7698	0.0007	-0.0013	-8.0386
	SP16	82	0.0251	0.0470	-8.0386	0	0	0
5	SP7	68	-0.0002	0.3185	-7.5591	-0.0040	0.0137	0.3220
	SP9	68	-0.0136	0.3253	-9.1364	0.0031	0.0121	0.3220
	SP11	68	-0.0177	-0.3184	-7.5301	-0.0004	0.0046	0.3220
	SP13	69	0.0003	-0.3065	-2.6938	0.0047	0.0295	0.3220
	SP16	91	0.0779	-0.0604	0.3220	0	0	0
6	SP7	70	-0.0147	0.3154	-2.4085	0.0026	-0.045	-7.8698
	SP9	70	0.0269	-0.2739	-7.1403	-0.0009	0.0170	-7.8698
	SP11	70	-0.0139	-0.3140	-2.3899	-0.0022	0.0147	-7.8698
	SP13	69	0.0367	0.2704	-0.9101	-0.0156	0.0357	-7.8698
	SP16	93	0.1386	-0.0546	-7.8698	0	0	0
7	SP7	72	-0.0406	0.3124	-1.2401	-0.0062	-0.0066	-8.4109
	SP9	72	-0.0226	0.3218	-2.7804	-0.0217	0.0313	-8.4109
	SP11	72	0.0153	-0.2704	-7.5021	-0.0010	-0.0069	-8.4109
	SP13	72	-0.0599	0.3508	0.3049	-0.0181	0.0331	-8.4109
	SP16	97	0.1857	-0.0496	-8.4109	0	0	0
8	SP7	73	-0.0147	-0.2138	-2.4325	0.0009	-0.0126	-7.4717
	SP9	74	0.0248	0.2423	-0.5700	-0.0183	0.0499	-7.4717
	SP11	74	0.0353	0.2836	4.0005	-0.0794	0.0384	-7.4717
	SP13	73	0.0275	-0.2356	-7.1352	-0.0001	0.0049	-7.4717
	SP16	97	0.2165	-0.0407	-7.4717	0	0	0
9	SP7	79	0.0376	-0.1909	-1.3349	-0.0009	-0.029	-7.5302
	SP9	79	0.0308	-0.2038	-2.9486	-0.0028	-0.0167	-7.5302
	SP11	79	0.0186	0.1891	-7.6524	-0.0004	-0.0334	-7.5302
	SP13	79	-0.0277	0.2565	-2.9015	-0.0103	-0.0364	-7.5302
	SP16	105	0.2467	0.0398	-7.5302	0	0	0
10	SP7	129	0.0024	0.1795	-8.2615	0.0008	-0.0143	0.0902
	SP9	130	0.0417	0.2033	2.7509	0	0	0.0902
	SP11	127	-0.0008	-0.1981	-8.3025	0.0040	-0.0144	0.0902
	SP13	129	-0.0176	0.2024	-6.7735	-0.0100	-0.0231	0.0902
	SP16	167	0.2455	0.0330	0.0902	0	0	0

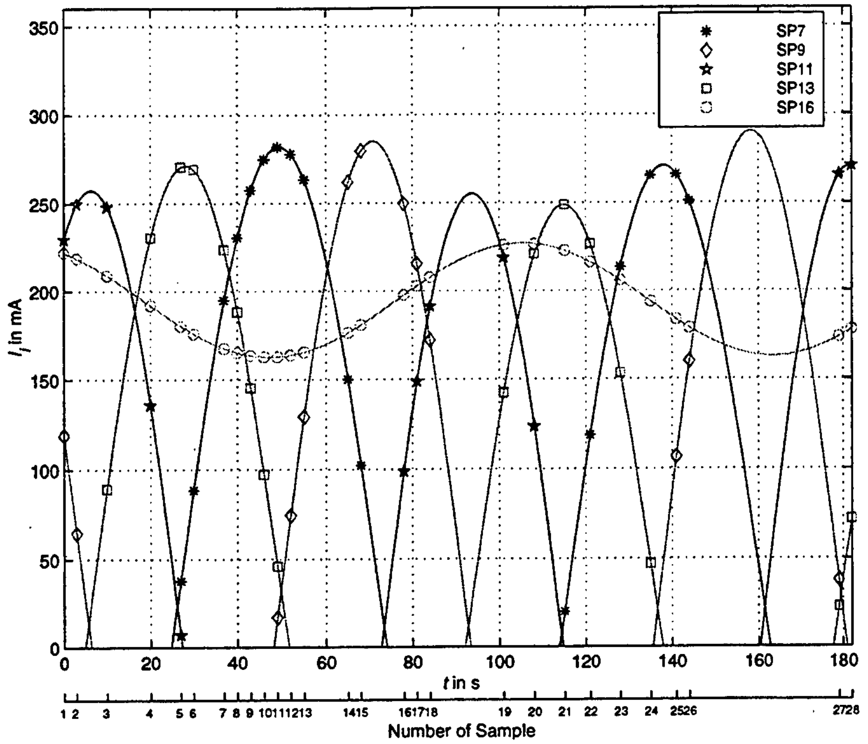


Figure H.1: Solar Panel Currents after Curve Fitting (*i* = 1)

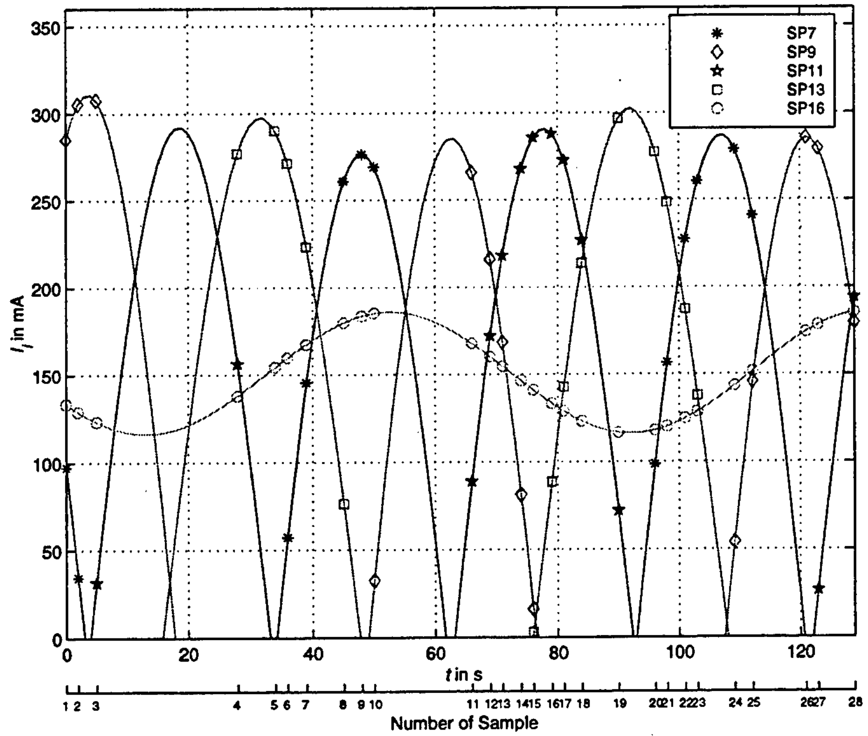


Figure H.2: Solar Panel Currents after Curve Fitting (*i* = 2)

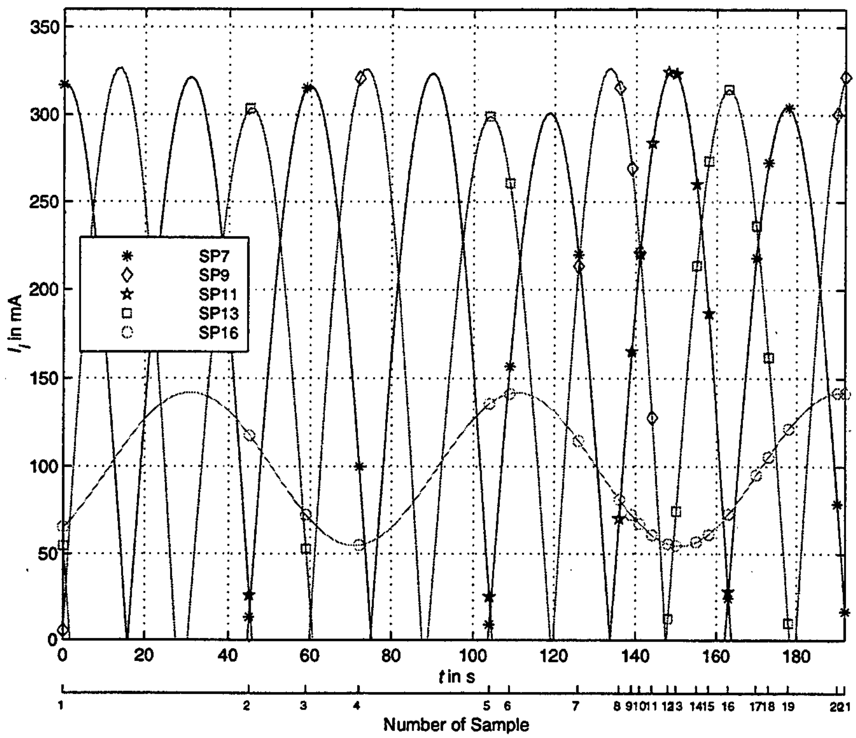


Figure H.3: Solar Panel Currents after Curve Fitting ($i = 3$)

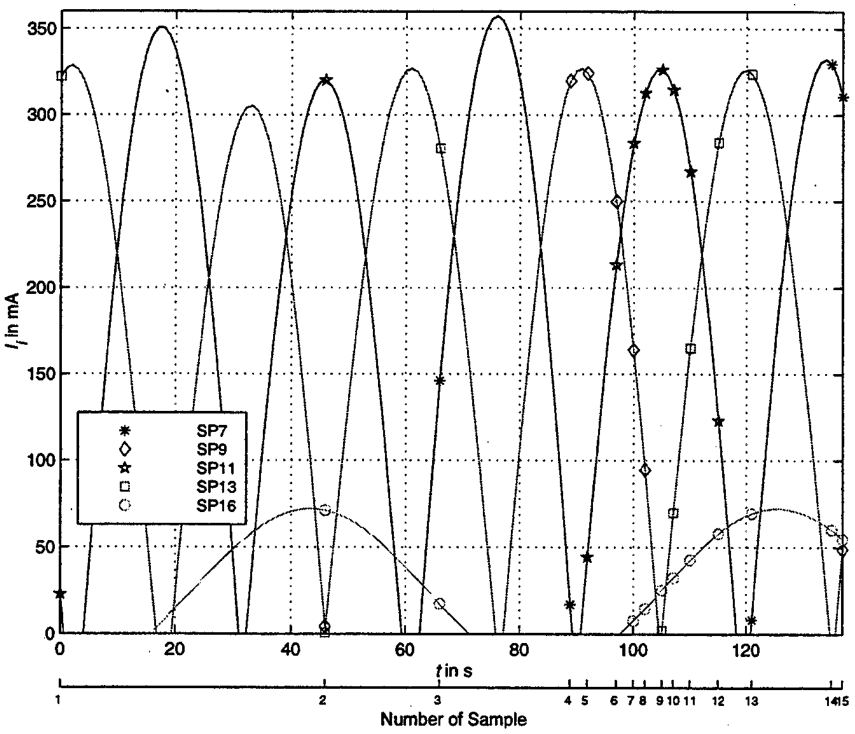


Figure H.4: Solar Panel Currents after Curve Fitting ($i = 4$)

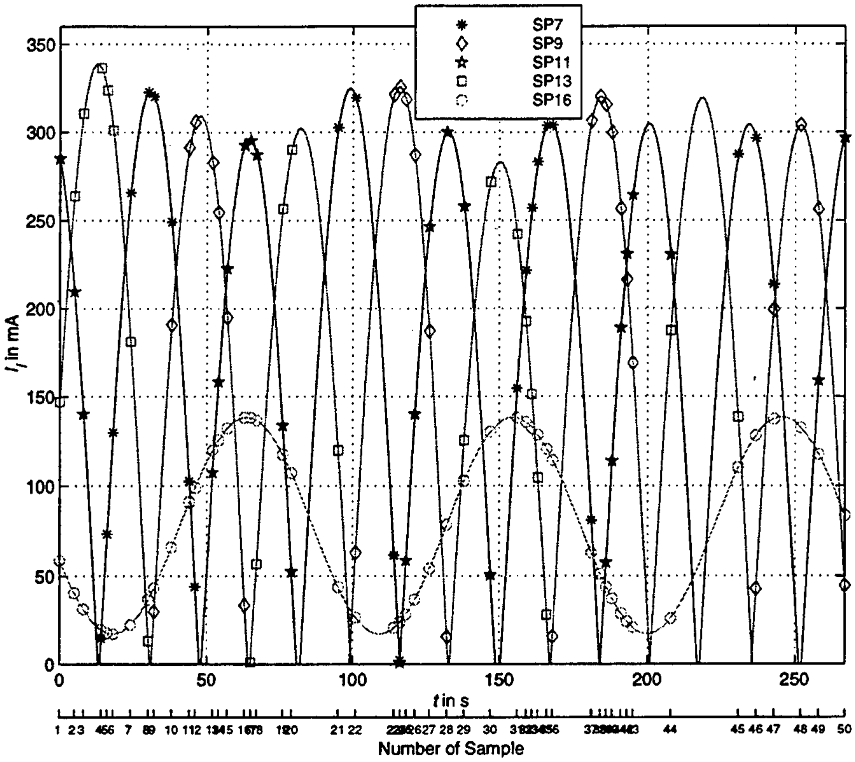


Figure H.5: Solar Panel Currents after Curve Fitting (i = 5)

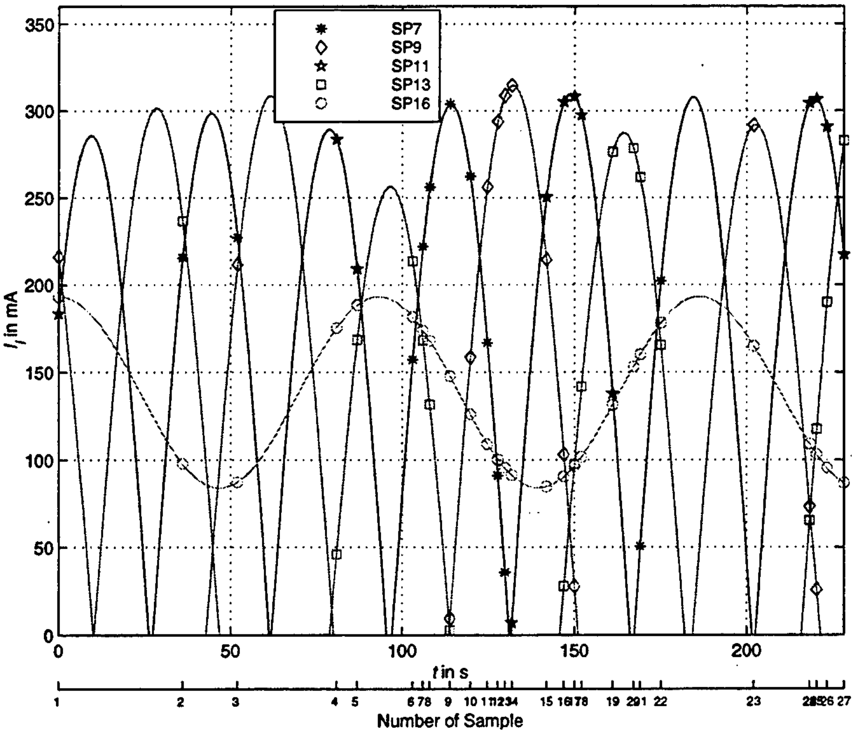


Figure H.6: Solar Panel Currents after Curve Fitting (i = 6)

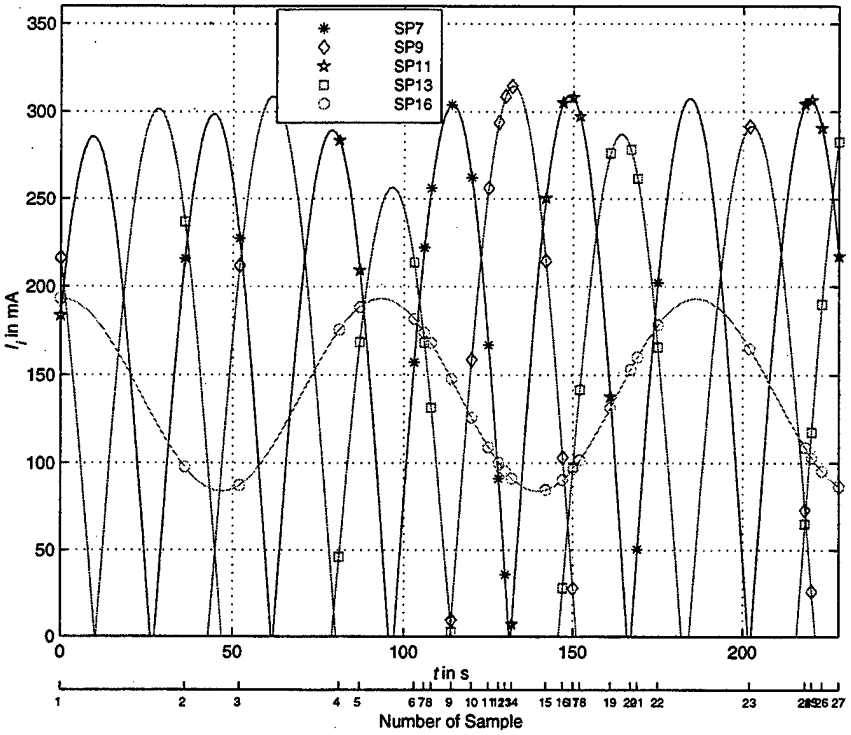


Figure H.7: Solar Panel Currents after Curve Fitting ($i = 7$)

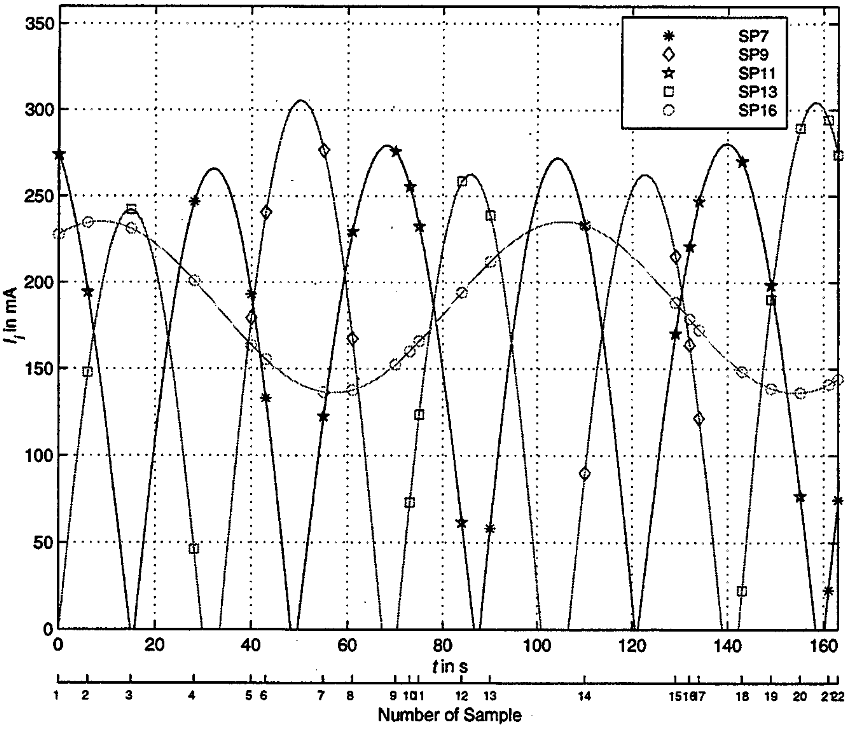


Figure H.8: Solar Panel Currents after Curve Fitting ($i = 8$)

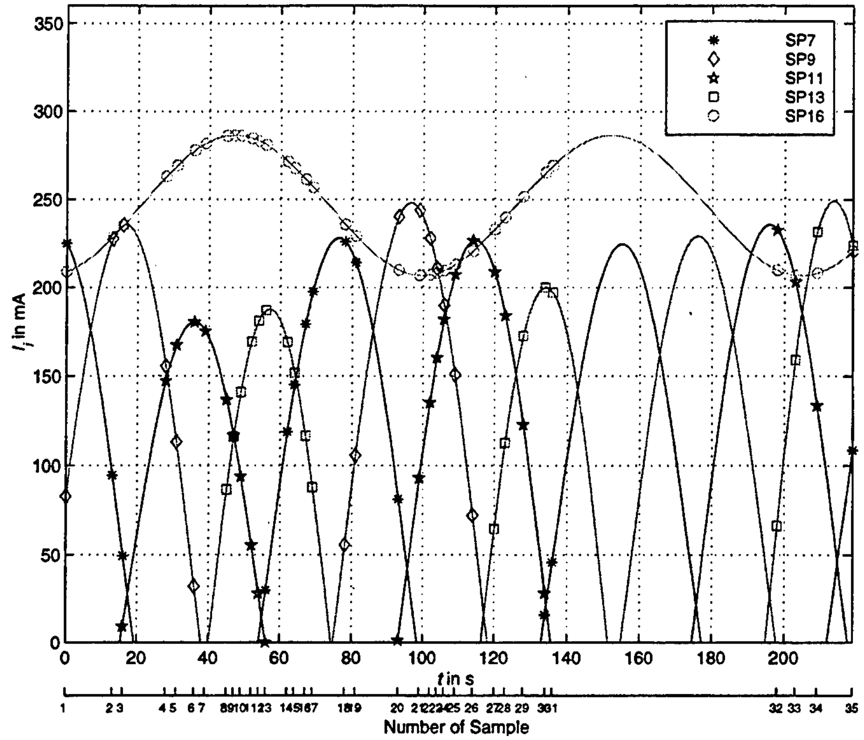


Figure H.9: Solar Panel Currents after Curve Fitting ($i = 9$)

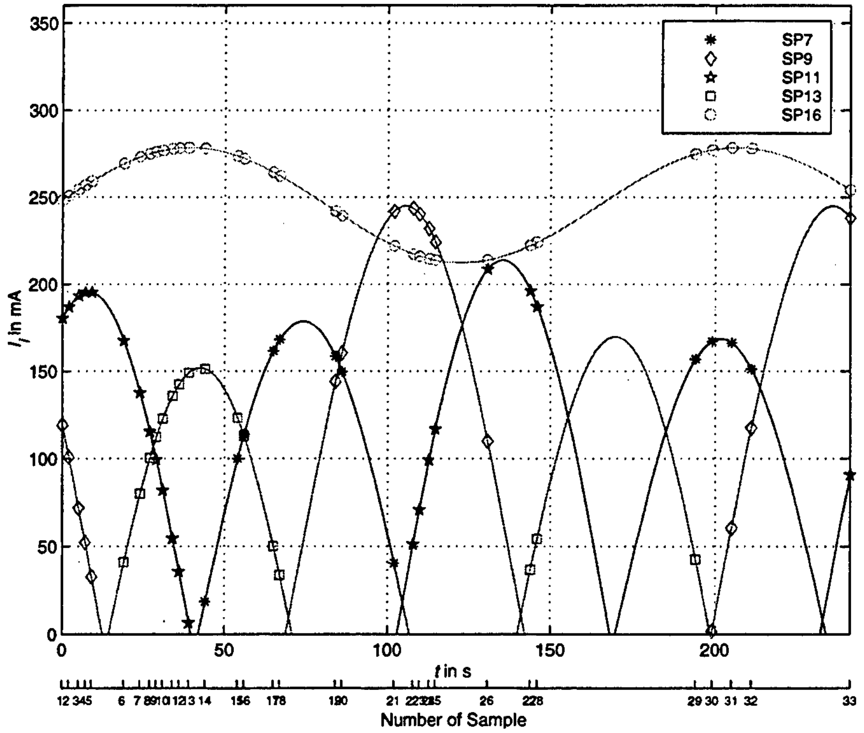


Figure H.10: Solar Panel Currents after Curve Fitting ($i = 10$)

APPENDIX I. VISUALIZATION OF ATTITUDE AND ROLL-RATE

The visualization tools provided in this report are briefly described in the following.

In order to display the attitude and rotational motion of PANSAT, the spacecraft is modeled using the vector coordinates of the solar panels and the LVI. The surface areas are color coded to simplify analysis of the attitude displayed as follows: solar panels equipped with current sensors are designated by magenta, all the other panels are designated by dark blue, and the triangles are marked by a light blue color. The geometrical axes fixed to the spacecraft are shown as well.

The attitude of the spacecraft at deployment being the only reference attitude is visualized with respect to the orbit-plane axes. This is done by using the Euler angles to transform the orbit-plane axes to the satellite's geometrical axes as determined in Section IV.A. and applied on Equation (3.15). The resulting DCM is required to transform the geometrical axes displayed and the vector coordinates of the model into the new orientation. The result is presented in Figure 4.2. The MATLAB source code can be found in Appendix J.

Furthermore, the roll-rate of PANSAT is displayed relative to the Sun. As discussed in Chapter V, the angular velocity component parallel to the Sun line cannot be detected and, therefore, is missing in the visualization as well. The solar vector computed from the estimated solar panel currents utilizing least square methods is used to determine the azimuth and elevation angle of the Sun in the geometrical coordinate system of PANSAT from Equations (5.21) and (5.22). These angles are determined ambiguously by observing the quadrant of the solar vector in order to transform the geometrical axes to a Sun directed system; the direction cosine matrix is given by Equation (5.20). This is done for every sample in order to animate the attitude relative to a fixed Sun's ray. Additionally, lighting is added to the scene. The source code is provided in the next appendix.

The spin and the nutation, which are examined from the solar panel currents in Section V.A.1., can be easily recognized by using this roll-rate display.

[This page is intentionally left blank.]

APPENDIX J. SOFTWARE SOURCE CODE

This appendix contains the software source code for visualization of data and for calculations in this report. The software is written in MATLAB version 5.2 and MAPLE V Release 5. The following source code files are included:

Filename	*Description*	*Pages*
DEPLOY.M	Determination of attitude at deployment	138 – 140
DISPALLP.M	Display of status snapshot data	141 – 146
DISPFITP.M	Display of solar panel currents after least squares fit	147 – 151
DISPRR.M	Display of data to evaluate roll-rate and attitude	152 – 153
DETSOLVX.M	Computation of solar vector from panel currents	154 – 155
DETSANG.M	Determination of location of Sun in geometrical frame	156 – 158
DETATTP.M	Determination of data required for attitude evaluation	159 – 165
DISPATTD.M	Display of attitude of PANSAT at deployment	166 – 170
DISPATTS.M	Visualization of roll-rate relative to the Sun	171 – 176
DEPLOY.MWS	Verification of the attitude of PANSAT at deployment	177 – 179
BODYAXES.MWS	Determination of principal moments of inertia and axes	180 – 181
CURVFIT.MWS	Curve fitting of solar panel currents	182 – 183

DEPLOY.M

```
function DEPLOY();

% DEPLOY
%
% Determination of Initial Attitude of PANSAT at Deployment
% from Orbiter (data provided by NASA JSC)
%
%
% (c) 1998 by Grasi


%%%%%%%%%%%%%%%%%%%%
% Initial attitude %
%%%%%%%%%%%%%%%%%%%%

% data @ MET 0/23:25:59

% Geocentric inertial to orbit-plane (M50 to LVLH)
fprintf('Geocentric inertial to orbit-plane\n');

q(1) = -.3200; % vector x component
q(2) = .5349; % vector y component
q(3) = -.7789; % vector z component
q(4) = .0699; % scalar

flag = q(1)^2 + q(2)^2 + q(3)^2 + q(4)^2;
if round(flag) ~= 1
   break
end

% Transformation of quaternions to DCM
T_LVLH_M50 = [2*(q(1)^2+q(4)^2)-1,2*(q(1)*q(2)+q(3)*q(4)),2*(q(1)*q(3)-q(2)*q(4));...
   2*(q(1)*q(2)-q(3)*q(4)),2*(q(2)^2+q(4)^2)-1,2*(q(2)*q(3)+q(1)*q(4));...
   2*(q(1)*q(3)+q(2)*q(4)),2*(q(2)*q(3)-q(1)*q(4)),2*(q(3)^2+q(4)^2)-1]

T_M50_S = [0,0,1;1,0,0;0,1,0]

T_OS = T_LVLH_M50 * T_M50_S

% Determination of euler angles from DCM
xi1 = atan(-T_OS(2,3)/T_OS(2,1));
xi2 = acos(-T_OS(2,2));
xi3_ = atan(-T_OS(3,2)/T_OS(1,2));

fprintf('OBSERVE QUADRANTS! (angles are not unambigious)\n');
fprintf('xi1: %f\n',xi1 * 180/pi);
fprintf('xi2: %f\n',xi2 * 180/pi);
fprintf('xi3: %f\n',xi3_ * 180/pi);

% Tranformation of orbit-plane axes to geocentric Axes
xi1=150.82*pi/180, xi2=28.60*pi/180, xi3_=332.23*pi/180 % determined by...
   % observing signs of sin/cos

T_OS_1 = [cos(xi1),0,-sin(xi1);0,1,0;sin(xi1),0,cos(xi1)];
T_OS_2 = [cos(xi2),sin(xi2),0;-sin(xi2),cos(xi2),0;0,0,1];
T_OS_3 = [cos(xi3_),0,-sin(xi3_);0,1,0;sin(xi3_),0,cos(xi3_)];
T_OS_4 = [1,0,0;0,-1,0;0,0,-1];

T_OS_ = T_OS_4 * T_OS_3 * T_OS_2 * T_OS_1


% Orbit-plane to body (LVLH to body (ORBITER))
fprintf('Orbit-plane to body (ORBITER)\n');

q(1) = .6322; % vector x component
q(2) = .2436; % vector y component
```

```
q(3) = -.7017; % vector z component
q(4) = .2206; % scalar

flag = q(1)^2 + q(2)^2 + q(3)^2 + q(4)^2;
if round(flag) ~= 1
   break
end

% Transformation of quaternions to DCM
T_BOrbO = [2*(q(1)^2+q(4)^2)-1,2*(q(1)*q(2)+q(3)*q(4)),2*(q(1)*q(3)-q(2)*q(4));...
   2*(q(1)*q(2)-q(3)*q(4)),2*(q(2)^2+q(4)^2)-1,2*(q(2)*q(3)+q(1)*q(4));...
   2*(q(1)*q(3)+q(2)*q(4)),2*(q(2)*q(3)-q(1)*q(4)),2*(q(3)^2+q(4)^2)-1]

% Determination of Euler angles from DCM
zet1 = atan(T_BOrbO(2,3)/T_BOrbO(3,3));
zet2 = asin(-T_BOrbO(1,3));
zet3 = atan(T_BOrbO(1,2)/T_BOrbO(1,1));

fprintf('OBSERVE QUADRANTS! (angles are not unambigious)\n');
fprintf('zet1: %f\n',zet1 * 180/pi);
fprintf('zet2: %f\n',zet2 * 180/pi);
fprintf('zet3: %f\n',zet3 * 180/pi);

% Transformation of orbit-plane axes to body axes (orbiter)
zet1=322.52*pi/180, zet2=84.10*pi/180, zet3=180.88*pi/180 % determined...
   % by observing signs of sin/cos

T_BOrbO_1 = [cos(zet3),sin(zet3),0;-sin(zet3),cos(zet3),0;0,0,1];
T_BOrbO_2 = [cos(zet2),0,-sin(zet2);0,1,0;sin(zet2),0,cos(zet2)];
T_BOrbO_3 = [1,0,0;0,cos(zet1),sin(zet1);0,-sin(zet1),cos(zet1)];

T_BOrbO_ = T_BOrbO_3 * T_BOrbO_2 * T_BOrbO_1


% Orbit-plane to geometrical PANSAT (LVLH to geometrical)
fprintf('Orbit-plane to geometrical (PANSAT)\n');

q(1) = .6322; % vector x component
q(2) = .2436; % vector y component
q(3) = -.7017; % vector z component
q(4) = .2206; % scalar

flag = q(1)^2 + q(2)^2 + q(3)^2 + q(4)^2;
if round(flag) ~= 1
   break
end

% Transformation of quaternions to DCM
T_BOrbO = [2*(q(1)^2+q(4)^2)-1,2*(q(1)*q(2)+q(3)*q(4)),2*(q(1)*q(3)-q(2)*q(4));...
   2*(q(1)*q(2)-q(3)*q(4)),2*(q(2)^2+q(4)^2)-1,2*(q(2)*q(3)+q(1)*q(4));...
   2*(q(1)*q(3)+q(2)*q(4)),2*(q(2)*q(3)-q(1)*q(4)),2*(q(3)^2+q(4)^2)-1];

T_X_BOrb = [0,1,0;0,0,-1;-1,0,0];
T_X_O = T_X_BOrb * T_BOrbO

% Determination of Euler angles from DCM
zet1 = atan(T_X_O(2,3)/T_X_O(3,3));
zet2 = asin(-T_X_O(1,3));
zet3 = atan(T_X_O(1,2)/T_X_O(1,1));

fprintf('OBSERVE QUADRANTS! (angles are not unambigious)\n');
fprintf('zet1: %f\n',zet1 * 180/pi);
fprintf('zet2: %f\n',zet2 * 180/pi);
fprintf('zet3: %f\n',zet3 * 180/pi);

% Transformation of orbit-plane axes to geometrical axes
zet1=355.28*pi/180, zet2=3.61*pi/180, zet3=308.23*pi/180 % determined...
   % by observing signs of sin/cos
```

```
T_X_O_1 = [cos(zet3),sin(zet3),0;-sin(zet3),cos(zet3),0;0,0,1];
T_X_O_2 = [cos(zet2),0,-sin(zet2);0,1,0;sin(zet2),0,cos(zet2)];
T_X_O_3 = [1,0,0;0,cos(zet1),sin(zet1);0,-sin(zet1),cos(zet1)];

T_X_O_ = T_X_O_3 * T_X_O_2 * T_X_O_1


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Transformation between standard epochs %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf('Transformation between epochs J2000 and B1950\n\n');

% Transformation matrix
T_J2000_B1950 = [.999925679,-.011181483,-.004859004;...
   .011181483,.999937485,-.000027163;...
   .004859004,-.000027170,.999988194];


% Geocentric inertial to orbit-plane: J2000

fprintf('Geocentric inertial to orbit-plane: J2000\n');

q(1) = -.3200; % vector x component
q(2) = .5349; % vector y component
q(3) = -.7789; % vector z component
q(4) = .0699; % scalar

flag = q(1)^2 + q(2)^2 + q(3)^2 + q(4)^2;
if round(flag) ~= 1
   break
end

% Transformation of quaternions to DCM
T_LVLH_M50 = [2*(q(1)^2+q(4)^2)-1,2*(q(1)*q(2)+q(3)*q(4)),2*(q(1)*q(3)-q(2)*q(4));...
   2*(q(1)*q(2)-q(3)*q(4)),2*(q(2)^2+q(4)^2)-1,2*(q(2)*q(3)+q(1)*q(4));...
   2*(q(1)*q(3)+q(2)*q(4)),2*(q(2)*q(3)-q(1)*q(4)),2*(q(3)^2+q(4)^2)-1]

T_M50_S = [0,0,1;1,0,0;0,1,0]

T_OS = T_LVLH_M50 * T_J2000_B1950.' * T_M50_S

% Determination of Euler angles from DCM
xi1 = atan(-T_OS(2,3)/T_OS(2,1));
xi2 = acos(-T_OS(2,2));
xi3_ = atan(-T_OS(3,2)/T_OS(1,2));

fprintf('OBSERVE QUADRANTS! (angles are not unambigious)\n');
fprintf('xi1: %f\n',xi1 * 180/pi);
fprintf('xi2: %f\n',xi2 * 180/pi);
fprintf('xi3: %f\n',xi3_ * 180/pi);

% Tranformation of orbit-plane axes to geocentric axes
xi1=151.91*pi/180, xi2=28.46*pi/180, xi3_=331.72*pi/180 % determined by...
   % observing signs of sin/cos

T_OS_1 = [cos(xi1),0,-sin(xi1);0,1,0;sin(xi1),0,cos(xi1)];
T_OS_2 = [cos(xi2),sin(xi2),0;-sin(xi2),cos(xi2),0;0,0,1];
T_OS_3 = [cos(xi3_),0,-sin(xi3_);0,1,0;sin(xi3_),0,cos(xi3_)];
T_OS_4 = [1,0,0;0,-1,0;0,0,-1];

T_OS_J2000 = T_OS_4 * T_OS_3 * T_OS_2 * T_OS_1
```

End of DEPLOY.M

DISPALLP.M

```
function DISPALLP()

% DISPALLP
%
% Display of Status Snapshot Data Concerning the Determination of
% PANSAT's Roll-rate (solar panel currents and temperatures)
%
%
% (c) 1998 by Grasi


% Load status data

if c == 1
   file = 'g:\pansat\data\status\1998-11-09\all_1998-11-09_10-24-38.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-22,-45,22,0,NaN,NaN];

elseif c == 2
   file = 'g:\pansat\data\status\1998-11-17\all_1998-11-17_06-29-17.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-30,-45,0,-15,NaN,NaN];

elseif c == 3
   file = 'g:\pansat\data\status\1998-11-18\all_1998-11-18_06-23-44.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-30,-45,0,-15,NaN,NaN];

elseif c == 4
   file = 'g:\pansat\data\status\1998-11-19\all_1998-11-19_06-19-52.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-30,-45,0,-15,NaN,NaN];

elseif c == 5
   file = 'g:\pansat\data\status\1998-12-05\all_1998-12-05_17-16-15.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,0,-17,34,17,NaN,NaN];


elseif c == 6
   file = 'g:\pansat\data\status\1998-12-06\all_1998-12-06_17-11-03.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,0,-17,34,17,NaN,NaN];

elseif c == 7
   file = 'g:\pansat\data\status\1998-12-07\all_1998-12-07_17-07-17.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-17,-35,18,0,NaN,NaN];

elseif c == 8
   file = 'g:\pansat\data\status\1998-12-08\all_1998-12-08_15-22-02.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-17,-35,18,0,NaN,NaN];

elseif c == 9
   file = 'g:\pansat\data\status\1998-12-09\all_1998-12-09_15-17-07.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,+20,0,-20,-40,NaN,NaN];


elseif c == 10
   file = 'g:\pansat\data\status\1998-12-14\all_1998-12-14_16-35-41.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-30,+60,+30,0,NaN,NaN];


end

load(file);


% Needed variables
N = size(time(:,1),1)-1
t = time(1:N,1)-time(1,1);
tE = t(N)

nB = 1, nE = N;


% Extract filename from file string
sB = findstr(file,'all');
sE = length(file);

filename = file(sB:sE);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot solar panel currents %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);
pos = get(gca,'Position');

% SP4
subplot(8,1,1);
plot(t,isolar(nB:nE,1)*1e3,'kv-');

set(gca,'Position',get(gca,'Position') + [0,.01,0,.01]);
axis([0 tE 0 350])
set(gca,'YTick',[0;100;200;300],'XTickLabel','');
grid;
ylabel('\itI\rm_{4} in mA');
title(['Solar Panel Currents of PANSAT (',filename,')'],'Interpreter','none');

% SP5
subplot(8,1,2);
plot(t,isolar(nB:nE,2)*1e3,'g.-');

set(gca,'Position',get(gca,'Position') + [0,.01,0,.01]);
axis([0 tE 0 350])
set(gca,'YTick',[0;100;200;300],'XTickLabel','');
grid;
ylabel('\itI\rm_{5} in mA');

% SP7
subplot(8,1,3);
plot(t,isolar(nB:nE,3)*1e3,'b*-');

set(gca,'Position',get(gca,'Position') + [0,.01,0,.01]);
axis([0 tE 0 350])
set(gca,'YTick',[0;100;200;300],'XTickLabel','');
grid;
ylabel('\itI\rm_{7} in mA');
```

```
% SP9
subplot(8,1,4);
plot(t,isolar(nB:nE,4)*1e3,'rd-');

set(gca,'Position',get(gca,'Position') + [0,.01,0,.01]);
axis([0 tE 0 350])
set(gca,'YTick',[0;100;200;300],'XTickLabel','');
grid;
ylabel('\itI\rm_{9} in mA');

% SP11
subplot(8,1,5);
plot(t,isolar(nB:nE,5)*1e3,'bp-');

set(gca,'Position',get(gca,'Position') + [0,.01,0,.01]);
axis([0 tE 0 350])
set(gca,'YTick',[0;100;200;300],'XTickLabel','');
grid;
ylabel('\itI\rm_{11} in mA');

% SP13
subplot(8,1,6);
plot(t,isolar(nB:nE,6)*1e3,'rs-');

set(gca,'Position',get(gca,'Position') + [0,.01,0,.01]);
axis([0 tE 0 350])
set(gca,'YTick',[0;100;200;300],'XTickLabel','');
grid;
ylabel('\itI\rm_{13} in mA');

% SP14
subplot(8,1,7);
plot(t,isolar(nB:nE,7)*1e3,'k^-');

set(gca,'Position',get(gca,'Position') + [0,.01,0,.01]);
axis([0 tE 0 350])
set(gca,'YTick',[0;100;200;300],'XTickLabel','');
grid;
ylabel('\itI\rm_{14} in mA');

% SP16
subplot(8,1,8);
plot(t,isolar(nB:nE,8)*1e3,'go-');

set(gca,'Position',get(gca,'Position') + [0,.01,0,.01]);
axis([0 tE 0 350])
set(gca,'YTick',[0;100;200;300]);
grid;
xlabel('\itt\rm in s');
ylabel('\itI\rm_{16} in mA');

% Create second Xaxis
c = get(gcf,'Color');

axes('Position',pos - [0,.07,0,.818],'Color','none','XColor','k','YColor',c);
xlabel('Number of Sample');

x = t / t(N);
for i = 1:N
   if i < 10
      s(i,1:2) = [num2str(i) ' '];
   else
      s(i,1:2) = num2str(i);
   end
end
set(gca,'XTick',x,'XTickLabel',s,'FontSize',7);
```

```
% All panel currents in one plot

scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);
pos = get(gca,'Position');

plot(t,isolar(nB:nE,1)*1e3,'kv-',t,isolar(nB:nE,2)*1e3,'g.-',...
   t,isolar(nB:nE,3)*1e3,'b*-',t,isolar(nB:nE,4)*1e3,'rd-',...
   t,isolar(nB:nE,5)*1e3,'bp-',t,isolar(nB:nE,6)*1e3,'rs-',...
   t,isolar(nB:nE,7)*1e3,'k^-',t,isolar(nB:nE,8)*1e3,'go-');

axis([0 tE 0 350])
grid;
xlabel('\itt\rm in s');
ylabel('\itI_{i}\rm in mA');

legend('SP4','SP5','SP7','SP9','SP11','SP13','SP14','SP16');
title(['Solar Panel Currents of PANSAT (',filename,')'],'Interpreter','none');

% Create second Xaxis
c = get(gcf,'Color');

axes('Position',pos - [0,.07,0,.818],'Color','none','XColor','k','YColor',c);
xlabel('Number of Sample');

x = t / t(N);
for i = 1:N
   if i < 10
      s(i,1:2) = [num2str(i) ' '];
   else
      s(i,1:2) = num2str(i);
   end
end
set(gca,'XTick',x,'XTickLabel',s,'FontSize',7);


% CV panel currents in one plot (phase shift suppressed)

scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);

plot(t+t_ps(3),isolar(nB:nE,3)*1e3,'b*-',t+t_ps(4),isolar(nB:nE,4)*1e3,'rd-',...
   t+t_ps(5),isolar(nB:nE,5)*1e3,'bp-',t+t_ps(6),isolar(nB:nE,6)*1e3,'rs-');

axis([0 tE 0 350])
grid;
xlabel('\itt\rm in s');
ylabel('\itI_{i}\rm in mA');

legend('SP7','SP9','SP11','SP13');
title(['Solar Panel Currents of PANSAT (phase shift adjusted) (',filename,')'],...
   'Interpreter','none');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot solar panel temperatures %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Panel temperatures of panels equipped with current sensor

scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);

plot(t,thermistors(nB:nE,36),'kv-',t,thermistors(nB:nE,50),'g.-',...
   t,thermistors(nB:nE,38),'b*-',t,thermistors(nB:nE,40),'rd-',...
   t,thermistors(nB:nE,42),'bp-',t,thermistors(nB:nE,44),'rs-',...
   t,thermistors(nB:nE,45),'k^-',t,thermistors(nB:nE,47),'go-');
```

```
grid;
xlabel('\itt\rm in s');
ylabel('\it\theta_{i}\rm in \circC');

legend('SP4','SP5','SP7','SP9','SP11','SP13','SP14','SP16');
title(['Solar Panel Temperatures of PANSAT (',filename,')'],'Interpreter','none');


% Panel temperatures of UDCK panels (SP14, SP15, SP17, SP18 and SP16)

scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);

plot(t,thermistors(nB:nE,45),'k^-',t,thermistors(nB:nE,46),'g*-';...
   t,thermistors(nB:nE,48),'bd-',t,thermistors(nB:nE,49),'rp-',...
   t,thermistors(nB:nE,47),'go-');

grid;
xlabel('\itt\rm in s');
ylabel('\it\theta_{i}\rm in \circC');
legend('SP14','SP15','SP17','SP18','SP16');
title(['Solar Panel Temperatures of PANSAT (',filename,')'],'Interpreter','none');


% Temperatures of MSA, MSB, TMUXA, TMUXB, EPS

scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);

plot(t,thermistors(nB:nE,23),'k*-',t,thermistors(nB:nE,24),'gd-',...
   t,thermistors(nB:nE,27),'bp-',t,thermistors(nB:nE,28),'rs-',...
   t,thermistors(nB:nE,31),'k.-',t,thermistors(nB:nE,32),'go-');

grid;
xlabel('\itt\rm in s');
ylabel('\it\theta_{i}\rm in \circC');
legend('EPSL','EPSS','MSA','MSB','TMUXA','TMUXB');
title(['Temperatures of Components of PANSAT (',filename,')'],'Interpreter','none');


break

% save variables

if c == 1
   fid = fopen('g:\pansat\data\status\1998-11-09\all_1998-11-09_10-24-38_cf.txt','w')
elseif c == 2
   fid = fopen('g:\pansat\data\status\1998-11-17\all_1998-11-17_06-29-17_cf.txt','w')
elseif c == 3
   fid = fopen('g:\pansat\data\status\1998-11-18\all_1998-11-18_06-23-44_cf.txt','w')
elseif c == 4
   fid = fopen('g:\pansat\data\status\1998-11-19\all_1998-11-19_06-19-52_cf.txt','w')
elseif c == 5
   fid = fopen('g:\pansat\data\status\1998-12-05\all_1998-12-05_17-16-15_cf.txt','w')
elseif c == 6
   fid = fopen('g:\pansat\data\status\1998-12-06\all_1998-12-06_17-11-03_cf.txt','w')
elseif c == 7
   fid = fopen('g:\pansat\data\status\1998-12-07\all_1998-12-07_17-07-17_cf.txt','w')
elseif c == 8
   fid = fopen('g:\pansat\data\status\1998-12-08\all_1998-12-08_15-22-02_cf.txt','w')
elseif c == 9
   fid = fopen('g:\pansat\data\status\1998-12-09\all_1998-12-09_15-17-07_cf.txt','w')
elseif c == 10
   fid = fopen('g:\pansat\data\status\1998-12-14\all_1998-12-14_16-35-41_cf.txt','w')
end

for i = 1:N
   fprintf(fid,'%d %f %f %f %f %f %f %f %f\n',t(i),isolar(i,:));
```

```
end
fclose(fid);
```

End of DISPALLP.M

DISPFITP.M

```
function DISPFITP()

% DISPFITP
%
% Display of Solar Panel Currents after Curve Fitting
%
%
% (c) 1998 by Grasi


c = 1

if c == 1

   file = 'g:\pansat\data\status\1998-11-09\all_1998-11-09_10-24-38.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-22,-45,22,0,NaN,NaN];

   % Parameters after curve fitting
   T = [NaN,NaN,88,87,88,87,NaN,116];
   a0 = [NaN,NaN,.0379,.0191,-.0211,.0306,NaN,.1949];
   b0 = [NaN,NaN,.2287,.2669,-.2824,.2387,NaN,-.0322];
   c0 = [NaN,NaN,-1.9626,-3.5644,-8.2818,-.4164,NaN,-7.2803];
   d0 = [NaN,NaN,.0020,-.0031,.0033,-.0058,NaN,0];
   e0 = [NaN,NaN,.0135,.0080,.0113,.0167,NaN,0];

elseif c == 2

   file = 'g:\pansat\data\status\1998-11-17\all_1998-11-17_06-29-17.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-30,-45,0,-15,NaN,NaN];

   % Parameters after curve fitting
   T = [NaN,NaN,59,59,59,60,NaN,79];
   a0 = [NaN,NaN,-.0257,-.0328,-.0031,-.0270,NaN,.1511];
   b0 = [NaN,NaN,.3140,-.3277,-.2947,.2700,NaN,.0349];
   c0 = [NaN,NaN,2.7382,-1.9363,-3.5570,-1.7698,NaN,-2.6115];
   d0 = [NaN,NaN,-.0044,.0029,-.0021,.0541,NaN,0];
   e0 = [NaN,NaN,-.0082,-.0179,-.0030,-.0051,NaN,0];

elseif c == 3

   file = 'g:\pansat\data\status\1998-11-18\all_1998-11-18_06-23-44.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-30,-45,0,-15,NaN,NaN];

   % Parameters after curve fitting
   T = [NaN,NaN,59,60,59,59,NaN,80];
   a0 = [NaN,NaN,.0149,-.0471,.0070,.0263,NaN,.0983];
   b0 = [NaN,NaN,-.2949,-.3734,.3174,-.2918,NaN,.0436];
   c0 = [NaN,NaN,-7.9227,-3.0005,-7.9994,-6.3863,NaN,-7.1391];
   d0 = [NaN,NaN,-.0001,.0004,-.0008,-.0098,NaN,0];
   e0 = [NaN,NaN,-.0106,.0005,-.0021,-.0107,NaN,0];

elseif c == 4

   file = 'g:\pansat\data\status\1998-11-19\all_1998-11-19_06-19-52.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-30,-45,0,-15,NaN,NaN];

   % Parameters after curve fitting
   T = [NaN,NaN,58,58,59,59,NaN,82];
   a0 = [NaN,NaN,-.0468,-.0469,-.0352,.0055,NaN,.0251];
   b0 = [NaN,NaN,-.3963,.3528,.3663,-.3210,NaN,.0470];
   c0 = [NaN,NaN,2.7894,-8.2841,-3.2968,-1.7698,NaN,-8.0386];
   d0 = [NaN,NaN,-.0045,.0089,-.0047,.0007,NaN,0];
   e0 = [NaN,NaN,-.0159,-.0138,-.0064,-.0013,NaN,0];

elseif c == 5

   file = ’g:\pansat\data\status\1998-12-05\all_1998-12-05_17-16-15.mat’;

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,0,-17,34,17,NaN,NaN];

   % Parameters after curve fitting
   T = [NaN,NaN,68,68,68,69,NaN,91];
   a0 = [NaN,NaN,-.0002,-.0136,-.0177,.0003,NaN,.0779];
   b0 = [NaN,NaN,.3185,.3253,-.3184,-.3065,NaN,-.0604];
   c0 = [NaN,NaN,-7.5591,-9.1364,-7.5301,-2.6938,NaN,.3220];
   d0 = [NaN,NaN,-.0040,.0031,-.0004,.0047,NaN,0];
   e0 = [NaN,NaN,.0137,.0121,.0046,.0295,NaN,0];

elseif c == 6

   file = ’g:\pansat\data\status\1998-12-06\all_1998-12-06_17-11-03.mat’;

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,0,-17,34,17,NaN,NaN];

   % Parameters after curve fitting
   T = [NaN,NaN,70,70,70,69,NaN,93];
   a0 = [NaN,NaN,-.0147,.0269,-.0139,.0367,NaN,.1386];
   b0 = [NaN,NaN,.3154,-.2739,-.3140,.2704,NaN,-.0546];
   c0 = [NaN,NaN,-2.4085,-7.1403,-2.3899,-.9101,NaN,-7.8698];
   d0 = [NaN,NaN,.0026,-.0009,-.0022,-.0156,NaN,0];
   e0 = [NaN,NaN,-.0045,.0170,.0147,.0357,NaN,0];

elseif c == 7

   file = ’g:\pansat\data\status\1998-12-07\all_1998-12-07_17-07-17.mat’;

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-17,-35,18,0,NaN,NaN];

   % Parameters after curve fitting
   T = [NaN,NaN,72,72,72,72,NaN,97];
   a0 = [NaN,NaN,-.0406,-.0226,.0153,-.0599,NaN,.1857];
   b0 = [NaN,NaN,.3124,.3218,-.2704,.3508,NaN,-.0496];
   c0 = [NaN,NaN,-1.2401,-2.7804,-7.5021,.3049,NaN,-8.4109];
   d0 = [NaN,NaN,-.0062,-.0217,-.0010,-.0181,NaN,0];
   e0 = [NaN,NaN,-.0066,.0313,-.0069,.0331,NaN,0];

elseif c == 8

   file = ’g:\pansat\data\status\1998-12-08\all_1998-12-08_15-22-02.mat’;

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-17,-35,18,0,NaN,NaN];

   % Parameters after curve fitting
   T = [NaN,NaN,73,74,74,73,NaN,97];
   a0 = [NaN,NaN,-.0147,.0248,.0353,.0275,NaN,.2165];
   b0 = [NaN,NaN,-.2138,.2423,.2836,-.2356,NaN,-.0407];
   c0 = [NaN,NaN,-2.4325,-.5700,4.0005,-7.1352,NaN,-7.4717];
   d0 = [NaN,NaN,.0009,-.0183,-.0794,-.0001,NaN,0];
   e0 = [NaN,NaN,-.0126,.0499,.0384,.0049,NaN,0];

elseif c == 9

   file = ’g:\pansat\data\status\1998-12-09\all_1998-12-09_15-17-07.mat’;
   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,+20,0,-20,-40,NaN,NaN];

   % Parameters after curve fitting
   T = [NaN,NaN,79,79,79,79,NaN,105];
   a0 = [NaN,NaN,.0376,.0308,.0186,-.0277,NaN,.2467];
   b0 = [NaN,NaN,-.1909,-.2038,.1891,.2565,NaN,.0398];
   c0 = [NaN,NaN,-1.3349,-2.9486,-7.6524,-2.9015,NaN,-7.5302];
   d0 = [NaN,NaN,-.0009,-.0028,-.0004,-.0103,NaN,0];
   e0 = [NaN,NaN,-.0029,-.0167,-.0334,-.0364,NaN,0];

elseif c == 10

   file = 'g:\pansat\data\status\1998-12-14\all_1998-12-14_16-35-41.mat';

   % Parameters for phase shift suppression
   t_ps = [NaN,NaN,-30,+60,+30,0,NaN,NaN];

   % Parameters after curve fitting (CURVFIT10.MWS)
   T = [NaN,NaN,129,130,127,129,NaN,167];
   a0 = [NaN,NaN,.0024,.0417,-.0008,-.0176,NaN,.2455];
   b0 = [NaN,NaN,.1795,.2033,-.1981,.2024,NaN,.0330];
   c0 = [NaN,NaN,-8.2615,2.7509,-8.3025,-6.7735,NaN,.0902];
   d0 = [NaN,NaN,.0008,0,.0040,-.0100,NaN,0];
   e0 = [NaN,NaN,-.0143,0,-.0144,-.0231,NaN,0];

end


% Load status snapshot data
load(file);

% Needed variables
N = size(time(:,1),1);
t = time(1:N,1)-time(1,1);
tE = t(N);

% Fit two sine curves in CV current curves (influence of TOP panel current)
for j = 1:8
   for l = 1:N
      cur(l,j) = a0(j) + b0(j).*sin(2*pi./T(j).*t(l) + c0(j)) + d0(j)...
         + e0(j).*sin(2*pi./T(8).*t(l) + c0(8));
   end
end

t_ = 0:tE;
N_ = length(t_);
cur_ = zeros(N_,8);

for j = 1:8
   for l = 1:N_
      cur_(l,j) = a0(j) + b0(j).*sin(2*pi./T(j).*t_(l) + c0(j)) + d0(j)...
         + e0(j).*sin(2*pi./T(8).*t_(l) + c0(8));
   end
end

% Extract filename from file string
sB = findstr(file,'all');
sE = length(file);

filename = file(sB:sE);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot solar panel currents %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% All panel currents in one plot
```

```
scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);
pos = get(gca,'Position');

plot(t,isolar(:,1)*1e3,'kv-',t,isolar(:,2)*1e3,'g.-',t,isolar(:,3)*1e3,'b*-',...
   t,isolar(:,4)*1e3,'rd-',t,isolar(:,5)*1e3,'bp-',t,isolar(:,6)*1e3,'rs-',...
   t,isolar(:,7)*1e3,'k^-',t,isolar(:,8)*1e3,'go-');

axis([0 tE 0 360])
grid;
xlabel('\itt\rm in s');
ylabel('\itI_{i}\rm in mA');

legend('SP4','SP5','SP7','SP9','SP11','SP13','SP14','SP16');
title(['Solar Panel Currents of PANSAT (',filename,')'],'Interpreter','none');

% Create second X axis
c = get(gcf,'Color');

axes('Position',pos - [0,.07,0,.818],'Color','none','XColor','k','YColor',c);
xlabel('Number of Sample');

x = t / t(N);
for i = 1:N
   if i < 10
      s(i,1:2) = [num2str(i) ' '];
   else
      s(i,1:2) = num2str(i);
   end
end
set(gca,'XTick',x,'XTickLabel',s,'FontSize',7);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solar panel currents after curve fitting %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% CV + TOP panel currents in one plot (after curve fitting)
scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);
pos = get(gca,'Position');

plot(t,cur(:,3)*1e3,'b*',t,cur(:,4)*1e3,'rd',t,cur(:,5)*1e3,'bp',...
   t,cur(:,6)*1e3,'rs',t,cur(:,8)*1e3,'go',...
   t_,cur_(:,3)*1e3,'b-',t_,cur_(:,4)*1e3,'r-',t_,cur_(:,5)*1e3,'b-',...
   t_,cur_(:,6)*1e3,'r-',t_,cur_(:,8)*1e3,'g-');

axis([0 tE 0 360])
grid;
xlabel('\itt\rm in s');
ylabel('\itI_{i}\rm in mA');

legend('SP7','SP9','SP11','SP13','SP16');
title(['Currents after Curve Fitting (',filename,')'],'Interpreter','none');

% Create second X axis
c = get(gcf,'Color');

axes('Position',pos - [0,.07,0,.818],'Color','none','XColor','k','YColor',c);
xlabel('Number of Sample');

x = t / t(N);
for i = 1:N
   if i < 10
      s(i,1:2) = [num2str(i) ' '];
   else
      s(i,1:2) = num2str(i);
```

```
    end
end
set(gca,'XTick',x,'XTickLabel',s,'FontSize',7);
```

End of DISPFITP.M

DISPRR.M

```
function disprr

% DISPRR
%
% Display of Data to Determine Roll-rate and Attitude
% Information of PANSAT
%
%
% (c) 1998 by Grasi


%%%%%%%%%%%%%%
% Data given %
%%%%%%%%%%%%%%

% UT required
tel_date = [1998 10 30 18 45 34;1998 11 09 18 24 38;1998 11 17 14 29 17;
   1998 11 18 14 23 44;1998 11 19 14 19 52;1998 12 06 01 16 15;
   1998 12 07 01 11 03;1998 12 08 01 07 17;1998 12 08 23 22 02;
   1998 12 09 23 17 07;1998 12 15 00 35 41;1998 12 16 00 31 17]

T_side = [72,88,59,59,58,68,70,72,73,79,129,127];

T_top = [NaN,116,79,80,82,91,93,97,97,105,167,181];

N = length(T_top);

% Moments of inertia (based on CAD model)
I1 = 1.160;
I2 = 1.066;
I3 = .792;


%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of time %
%%%%%%%%%%%%%%%%%%%%%%%%

for j = 1:N

% Universal time
UT = (tel_date(j,4) + tel_date(j,5)/60 + tel_date(j,6)/3600) % in hours

% Determination of Julian Date
dam = [31 28 31 30 31 30 31 31 30 31 30 31]; % days a month

% Check for leap years...
   % (leap years must be divisible by 4, 2000 by 400;...
   % since 2000 is a leap year, no different algorithm necessary)
if rem(tel_date(j,1)/4,1) == 0
   dam(2) = 29
end

% Calculate day of year
i = 1:(tel_date(j,2)-1);
doy = tel_date(j,3) + sum(dam(i));

tab_JD_Jan00_UT = [1997,50448.5;1998,50813.5;1999,51178.5;2000,51543.5];

for i = 1:size(tab_JD_Jan00_UT(:,1),1)
   if tel_date(j,1) == tab_JD_Jan00_UT(i,1)
      JD_Jan00_UT = tab_JD_Jan00_UT(i,2) + 2400000;
   end
end

JD_UT1(j) = JD_Jan00_UT + doy + UT/24 % Julian Date for tel_date
```

```
end

t = JD_UT1(:) - JD_UT1(1)


%%%%%%%%%%%%%
% Roll-rate %
%%%%%%%%%%%%%

w_side = 360./T_side;

w_top = 360./T_top;
w_top(1) = 0;

scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);

plot(t,w_side,'bd-',t,w_top,'r.-');

grid;
xlabel('\itt\rm in days');
ylabel('\it\omega\rm in \circ/s');

legend('\it\omega_{side}\rm','\it\omega_{top}\rm');
title('Rotational Motion of PANSAT');


%%%%%%%%%%%%%%%%%%%%%
% Rotational energy %
%%%%%%%%%%%%%%%%%%%%%

T_rot = 1/2*((I1+I2)*(2*pi./T_top).^2 + I3*(2*pi./T_side).^2);

scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);

plot(t,T_rot*1e03,'bd-');

grid;
xlabel('t in days');
ylabel('T_{rot} in mJ');

title('Rotational Energy');
```

End of DISPRR.M

DETSOLVX.M

```
function rSX_ = DETSOLVX(outp,isolar)

% DETSOLVX
%
% Determination of Solar Vector from Solar Panel Data with
% respect to Geometrical Axes of PANSAT
%
% usage: DETSOLVX(outp,isolar)
%
%          outp = 0: no display of currents
%          outp = 1: display of currents
%          isolar  : solar panel currents matrix
%
% returns matrix of solar vector rSX_
%
%
% (c) 1998 by Grasi


% Relationship between logical numbers and MATLAB's arrays numbers
SP = [4;5;7;9;11;13;14;16];

% Number of samples
N = size(isolar(:,1),1);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Display solar panel current measurements %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if outp == 1
   fprintf('\nSolar Panel Currents (in mA)');
   for l = 1:N
      fprintf('\n l=%d: ',l);
      for j = 1:8
         fprintf('I%d=%.1f, ',SP(j),isolar(l,j)*1e3);
      end
   end
   fprintf('\n\n');
end


%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine solar vector %
%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initialize arrays
cur = zeros(1,8); % array for currents
val = zeros(1,3); % vector for solar vector elements at a sample point

for l = 1:N
   % Initialize arrays
   cur = zeros(1,8); % array for currents
   val = zeros(1,3);

   cur = isolar(l,:);
   cur(1) = 0; % SP4 and SP14 are not taken into account!
   cur(7) = 0;

   % Determine two highest current values
   for j = 1:2
      [Y I] = max(cur);
      cur(I) = 0; % set zero due to second loop

      switch I
         case 2, val(2) = -Y; % SP5 (-Y_), 2nd element in vector
         case 3, val(3) = Y; % SP7 (+Z_), 3rd element in vector
         case 4, val(1) = -Y; % SP9 (-X_), 1st element in vector
         case 5, val(3) = -Y; % SP11 (-Z_), 3rd element in vector
         case 6, val(1) = Y; % SP13 (+X_), 1st element in vector
         case 8, val(2) = Y; % SP16 (+Y_), 2nd element in vector
      end
   end

   % Which element is left?
   for j = 1:3
      if val(j) == 0
         h = j;
      end
   end

   % Match third current to corresponding quadrant
   switch h
      case 1, poss = [4,6]; % 1st element: SP9 (-) or SP13 (+)
      case 2, poss = [2,8]; % 2nd element: SP5 (-) or SP16 (+)
      case 3, poss = [5,3]; % 3rd element: SP11 (-) or SP7 (+)
   end

   if cur(poss(1)) > cur(poss(2)) % take current with higher value
      val(h) = -cur(poss(1)); % always negative
   else
      val(h) = cur(poss(2)); % always positive
   end

   % Normalize solar vector
   rSX_(1,:) = val / norm(val,2);

end
```

End of DETSOLVX.M

DETSANG.M

```
function DETSANG();

% DETSANG
%
% Determination of Azimuth and Elevation of Solar Vector
% in Geometrical System
%
%
% (c) 1998 by Grasi


%%%%%%%%%%%%%%
% Data given %
%%%%%%%%%%%%%%

c = 2

% Files containing status data

if c == 1

   file = 'g:\pansat\data\status\1998-11-09\all_1998-11-09_10-24-38.mat';

   % Parameters after curve fitting
   T = [NaN,NaN,88,87,88,87,NaN,116]; % period
   a0 = [NaN,NaN,.0379,.0191,-.0211,.0306,NaN,.1949]; % shift along y axis (1st sine)
   b0 = [NaN,NaN,.2287,.2669,-.2824,.2387,NaN,-.0322]; % amplitude
   c0 = [NaN,NaN,-1.9626,-3.5644,-8.2818,-.4164,NaN,-7.2803]; % phase shift
   d0 = [NaN,NaN,.0020,-.0031,.0033,-.0058,NaN,0]; % shift along y axis (2nd sine)
   e0 = [NaN,NaN,.0135,.0080,.0113,.0167,NaN,0]; % amplitude

elseif c == 2

   file = 'g:\pansat\data\status\1998-11-17\all_1998-11-17_06-29-17.mat';

   % Parameters after curve fitting
   T = [NaN,NaN,59,59,59,60,NaN,79];
   a0 = [NaN,NaN,-.0257,-.0328,-.0031,-.0270,NaN,.1511];
   b0 = [NaN,NaN,.3140,-.3277,-.2947,.2700,NaN,.0349];
   c0 = [NaN,NaN,2.7382,-1.9363,-3.5570,-1.7698,NaN,-2.6115];
   d0 = [NaN,NaN,-.0044,.0029,-.0021,.0541,NaN,0];
   e0 = [NaN,NaN,-.0082,-.0179,-.0030,-.0051,NaN,0];

elseif c == 3

   file = 'g:\pansat\data\status\1998-11-18\all_1998-11-18_06-23-44.mat';

   % Parameters after curve fitting
   T = [NaN,NaN,59,60,59,59,NaN,80];
   a0 = [NaN,NaN,.0149,-.0471,.0070,.0263,NaN,.0983];
   b0 = [NaN,NaN,-.2949,-.3734,.3174,-.2918,NaN,.0436];
   c0 = [NaN,NaN,-7.9227,-3.0005,-7.9994,-6.3863,NaN,-7.1391];
   d0 = [NaN,NaN,-.0001,.0004,-.0008,-.0098,NaN,0];
   e0 = [NaN,NaN,-.0106,.0005,-.0021,-.0107,NaN,0];

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate required variables %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Load status data
load(file);

% Extract filename from file string
```

```
sB = findstr(file,'all');
sE = length(file);

filename = file(sB:sE)


% Needed variables
N = 0; t = 0; tE = 0;

N = size(time(:,1),1);
t = time(1:N,1)-time(1,1);
tE = t(N);

% Or more samples!
Ts = 3;
t = 0:Ts:tE;
N = length(t)

% Calculate currents at sample points
cur = zeros(N,8);
for j = 1:8
   for l = 1:N
      cur(l,j) = a0(j) + b0(j).*sin(2*pi./T(j).*t(l) + c0(j)) + d0(j)...
         + e0(j).*sin(2*pi./T(8).*t(l) + c0(8));
   end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine solar vector %
%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determine solar vector from solar panel currents

rSX_ = detsolvX(0,cur); % referred to geometrical axes of PANSAT


% Calculate azimuth and elevation

for l = 1:N
   aS(l,1) = atan(rSX_(l,1)/rSX_(l,3));
   dS(l,1) = asin(rSX_(l,2));
end

aS*180/pi
dS*180/pi

% Plot
scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);

plot(t,aS*180/pi,'kd-',t,dS*180/pi,'bo-');

grid;
xlabel('\itt\rm in s');
ylabel('Angles in \circ');
legend('\it\alpha_{S}','\it\delta_{S}');
title(['Azimuth and Elevation of Solar Vector in Geometric Frame ' '(',filename,')'],...
   'Interpreter','none');

% Angular velocities from azimuth and elevation of PANSAT

for l = 1:N-1
   if sign(aS(l+1)) == sign(aS(l))
      wY_(l,1) = -(aS(l+1) - aS(l))/Ts;
   else
      wY_(l,1) = NaN;
   end
   wX_S(1,1) = -(dS(l+1) - dS(l))/Ts;
end

wY_*180/pi
wX_S*180/pi

% Plot
scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);

plot(t(1:N-1),wX_S*180/pi,'bo-',t(1:N-1),wY_*180/pi,'rd-');

grid;
xlabel('\itt\rm in s');
ylabel('Angular Velocities in \circ/s');
legend('\it\omega\rm_{X'',\itS}','\it\omega\rm_{Y''}');
title(['Angular Velocities about two Axes referred to the Sun ' '(',filename,')'],...
   'Interpreter','none');
```

End of DETSANG.M

DETATTP.M

```
function DETATTP();

% DETATTP
%
% Determination of Attitude of PANSAT
% Relate Angle between Symmetrical Axis and Solar Vector
% to Beta Angle
%
%
% (c) 1998 by Grasi


nB = 1;
nE = 10;


%%%%%%%%%%%%%%
% Data given %
%%%%%%%%%%%%%%

for c = nB:nE

   c

% Files containing status data

if c == 1

file = 'g:\pansat\data\status\1998-11-09\all_1998-11-09_10-24-38.mat';

% Given orbital elements (98-11-09_18-25-00)
tel_date = [1998 11 09 18 25 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 86.5*pi/180; % right ascension of the ascending node in rad
xi2 = 28.44*pi/180; % orbit inclination in rad
xi3 = 113*pi/180; % argument of perigee
tau = 355*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,88,87,88,87,NaN,116]; % period
a0 = [NaN,NaN,.0379,.0191,-.0211,.0306,NaN,.1949]; % shift along y axis (1st sine)
b0 = [NaN,NaN,.2287,.2669,-.2824,.2387,NaN,-.0322]; % amplitude
c0 = [NaN,NaN,-1.9626,-3.5644,-8.2818,-.4164,NaN,-7.2803]; % phase shift
d0 = [NaN,NaN,.0020,-.0031,.0033,-.0058,NaN,0]; % shift along y axis (2nd sine)
e0 = [NaN,NaN,.0135,.0080,.0113,.0167,NaN,0]; % amplitude

elseif c == 2

file = 'g:\pansat\data\status\1998-11-17\all_1998-11-17_06-29-17.mat';

% Given orbital elements (98-11-17_06-29-17)
tel_date = [1998 11 17 14 29 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 35.2*pi/180; % right ascension of the ascending node in rad
xi2 = 28.45*pi/180; % orbit inclination in rad
xi3 = 133*pi/180; % argument of perigee
tau = 338*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,59,59,59,60,NaN,79];
a0 = [NaN,NaN,-.0257,-.0328,-.0031,-.0270,NaN,.1511];
b0 = [NaN,NaN,.3140,-.3277,-.2947,.2700,NaN,.0349];
c0 = [NaN,NaN,2.7382,-1.9363,-3.5570,-1.7698,NaN,-2.6115];
d0 = [NaN,NaN,-.0044,.0029,-.0021,.0541,NaN,0];
e0 = [NaN,NaN,-.0082,-.0179,-.0030,-.0051,NaN,0];
```

```
elseif c == 3

file = 'g:\pansat\data\status\1998-11-18\all_1998-11-18_06-23-44.mat';

% Given orbital elements (98-11-18_06-23-44)
tel_date = [1998 11 18 14 24 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 28.7*pi/180; % right ascension of the ascending node in rad
xi2 = 28.45*pi/180; % orbit inclination in rad
xi3 = 143*pi/180; % argument of perigee
tau = 327*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,59,60,59,59,NaN,80];
a0 = [NaN,NaN,.0149,-.0471,.0070,.0263,NaN,.0983];
b0 = [NaN,NaN,-.2949,-.3734,.3174,-.2918,NaN,.0436];
c0 = [NaN,NaN,-7.9227,-3.0005,-7.9994,-6.3863,NaN,-7.1391];
d0 = [NaN,NaN,-.0001,.0004,-.0008,-.0098,NaN,0];
e0 = [NaN,NaN,-.0106,.0005,-.0021,-.0107,NaN,0];

elseif c == 4

file = 'g:\pansat\data\status\1998-11-19\all_1998-11-19_06-19-52.mat';

% Given orbital elements (98-11-19_06-19-52)
tel_date = [1998 11 19 14 20 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 22.1*pi/180; % right ascension of the ascending node in rad
xi2 = 28.46*pi/180; % orbit inclination in rad
xi3 = 134*pi/180; % argument of perigee
tau = 342*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,58,58,59,59,NaN,82];
a0 = [NaN,NaN,-.0468,-.0469,-.0352,.0055,NaN,.0251];
b0 = [NaN,NaN,-.3963,.3528,.3663,-.3210,NaN,.0470];
c0 = [NaN,NaN,2.7894,-8.2841,-3.2968,-1.7698,NaN,-8.0386];
d0 = [NaN,NaN,-.0045,.0089,-.0047,.0007,NaN,0];
e0 = [NaN,NaN,-.0159,-.0138,-.0064,-.0013,NaN,0];

elseif c == 5

file = 'g:\pansat\data\status\1998-12-05\all_1998-12-05_17-16-15.mat';

% Given orbital elements (98-12-06_01-16-00)
tel_date = [1998 12 06 01 16 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 274.4*pi/180; % right ascension of the ascending node in rad
xi2 = 28.46*pi/180; % orbit inclination in rad
xi3 = 67*pi/180; % argument of perigee
tau = 349*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,68,68,68,69,NaN,91];
a0 = [NaN,NaN,-.0002,-.0136,-.0177,.0003,NaN,.0779];
b0 = [NaN,NaN,.3185,.3253,-.3184,-.3065,NaN,-.0604];
c0 = [NaN,NaN,-7.5591,-9.1364,-7.5301,-2.6938,NaN,.3220];
d0 = [NaN,NaN,-.0040,.0031,-.0004,.0047,NaN,0];
e0 = [NaN,NaN,.0137,.0121,.0046,.0295,NaN,0];

elseif c == 6

file = 'g:\pansat\data\status\1998-12-06\all_1998-12-06_17-11-03.mat';

% Given orbital elements (98-12-07_01-11-00)
tel_date = [1998 12 07 01 11 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 267.9*pi/180; % right ascension of the ascending node in rad
xi2 = 28.46*pi/180; % orbit inclination in rad
```

```
xi3 = 72*pi/180; % argument of perigee
tau = 347*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,70,70,70,69,NaN,93];
a0 = [NaN,NaN,-.0147,.0269,-.0139,.0367,NaN,.1386];
b0 = [NaN,NaN,.3154,-.2739,-.3140,.2704,NaN,-.0546];
c0 = [NaN,NaN,-2.4085,-7.1403,-2.3899,-.9101,NaN,-7.8698];
d0 = [NaN,NaN,.0026,-.0009,-.0022,-.0156,NaN,0];
e0 = [NaN,NaN,-.0045,.0170,.0147,.0357,NaN,0];

elseif c == 7

file = 'g:\pansat\data\status\1998-12-07\all_1998-12-07_17-07-17.mat';

% Given orbital elements (98-12-08_01-07-00)
tel_date = [1998 12 08 01 07 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 261.4*pi/180; % right ascension of the ascending node in rad
xi2 = 28.46*pi/180; % orbit inclination in rad
xi3 = 78*pi/180; % argument of perigee
tau = 347*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,72,72,72,72,NaN,97];
a0 = [NaN,NaN,-.0406,-.0226,.0153,-.0599,NaN,.1857];
b0 = [NaN,NaN,.3124,.3218,-.2704,.3508,NaN,-.0496];
c0 = [NaN,NaN,-1.2401,-2.7804,-7.5021,.3049,NaN,-8.4109];
d0 = [NaN,NaN,-.0062,-.0217,-.0010,-.0181,NaN,0];
e0 = [NaN,NaN,-.0066,.0313,-.0069,.0331,NaN,0];

elseif c == 8

file = 'g:\pansat\data\status\1998-12-08\all_1998-12-08_15-22-02.mat';

% Given orbital elements (98-12-08_23-22-00)
tel_date = [1998 12 08 23 22 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 255.3*pi/180; % right ascension of the ascending node in rad
xi2 = 28.46*pi/180; % orbit inclination in rad
xi3 = 76*pi/180; % argument of perigee
tau = 335*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,73,74,74,73,NaN,97];
a0 = [NaN,NaN,-.0147,.0248,.0353,.0275,NaN,.2165];
b0 = [NaN,NaN,-.2138,.2423,.2836,-.2356,NaN,-.0407];
c0 = [NaN,NaN,-2.4325,-.5700,4.005,-7.1352,NaN,-7.4717];
d0 = [NaN,NaN,.0009,-.0183,-.0794,-.0001,NaN,0];
e0 = [NaN,NaN,-.0126,.0499,.0384,.0049,NaN,0];

elseif c == 9

file = 'g:\pansat\data\status\1998-12-09\all_1998-12-09_15-17-07.mat';

% Given orbital elements (98-12-09_23-17-00)
tel_date = [1998 12 09 23 17 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 248.8*pi/180; % right ascension of the ascending node in rad
xi2 = 28.46*pi/180; % orbit inclination in rad
xi3 = 83*pi/180; % argument of perigee
tau = 332*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,79,79,79,79,NaN,105];
a0 = [NaN,NaN,.0376,.0308,.0186,-.0277,NaN,.2467];
b0 = [NaN,NaN,-.1909,-.2038,.1891,.2565,NaN,.0398];
c0 = [NaN,NaN,-1.3349,-2.9486,-7.6524,-2.9015,NaN,-7.5302];
d0 = [NaN,NaN,-.0009,-.0028,-.0004,-.0103,NaN,0];
```

```
e0 = [NaN,NaN,-.0029,-.0167,-.0334,-.0364,NaN,0];

elseif c == 10

file = ’g:\pansat\data\status\1998-12-14\all_1998-12-14_16-35-41.mat’;

% Given orbital elements (98-12-09_23-17-00)
tel_date = [1998 12 15 00 35 00]; % start date and time for telemetry records (UT)
   % (format: year month day hour minute second)
xi1 = 215.6*pi/180; % right ascension of the ascending node in rad
xi2 = 28.45*pi/180; % orbit inclination in rad
xi3 = 115*pi/180; % argument of perigee
tau = 343*pi/180; % true anomaly

% Parameters after curve fitting
T = [NaN,NaN,129,130,127,129,NaN,167];
a0 = [NaN,NaN,.0024,.0417,-.0008,-.0176,NaN,.2455];
b0 = [NaN,NaN,.1795,.2033,-.1981,.2024,NaN,.0330];
c0 = [NaN,NaN,-8.2615,2.7509,-8.3025,-6.7735,NaN,.0902];
d0 = [NaN,NaN,.0008,0,.0040,-.0100,NaN,0];
e0 = [NaN,NaN,-.0143,0,-.0144,-.0231,NaN,0];


end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate required variables %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Load status data
load(file);

% Extract filename from file string
sB = findstr(file,’all’);
sE = length(file);

filename = file(sB:sE)

if c == nB
   filenameB = filename;
elseif c == nE
   filenameE = filename;
end

date_stamp(c,:) = filename(5:sE-sB-3);
date_stamp(c,11) = ’;’;

% Needed variables
N = 0; t = 0; tE = 0;

N = size(time(:,1),1);
t = time(1:N,1)-time(1,1);
tE = t(N);

% Or more samples!
t = 0:tE;
N = length(t);

% Calculate currents at sample points
cur = zeros(N,8);
for j = 1:8
   for l = 1:N
      cur(l,j) = a0(j) + b0(j).*sin(2*pi./T(j).*t(l) + c0(j)) + d0(j)...
         + e0(j).*sin(2*pi./T(8).*t(l) + c0(8));
   end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine solar vector and angle of sun cone %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determine solar vector from solar panel currents

rSX_ = detsolvX(0,cur); % referred to geometrix axes of PANSAT


% Determine angle of sun cone...
   % (i.e., angle between solar vector and symmetric (Y_) axis of PANSAT)

a_SY = zeros(N,1);

Y_ = [0,1,0];

for l = 1:N
   a_SY(l,1) = acos(dot(rSX_(l,:),Y_)/norm(rSX_(l,:),2));
end

a_SY_max = max(a_SY)*180/pi
a_SY_min = min(a_SY)*180/pi
a_SY_mean = mean(a_SY)*180/pi


% Plot
scr_sz = get(0,’ScreenSize’);
figure(’Position’,[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,’Position’,[.130 .122 .775 .822]);

plot(t,a_SY*180/pi,’k-’);

grid;
xlabel(’t in s’);
ylabel(’Angle in deg’);
legend(’Geom - Y’’’);
title([’Angle Geometrical and Solar Vector’ ’(’,filename,’)’],...
   ’Interpreter’,’none’);

a_SY_(1,c) = max(a_SY);
a_SY_(2,c) = min(a_SY);
a_SY_(3,c) = mean(a_SY);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of beta angle %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Universal time
UT = (tel_date(4) + tel_date(5)/60 + tel_date(6)/3600) % in hours

% Determination of Julian Date
dam = [31 28 31 30 31 30 31 31 30 31 30 31]; % days a month

% Check for leap years...
   % (leap years must be divisible by 4, 2000 by 400;...
   % since 2000 is a leap year, no different algorithm necessary)
if rem(tel_date(1)/4,1) == 0
   dam(2) = 29
end

% Calculate day of year
i = 1:(tel_date(2)-1);
doy = tel_date(3) + sum(dam(i));

tab_JD_Jan00_UT = [1997,50448.5;1998,50813.5;1999,51178.5;2000,51543.5];

for i = 1:size(tab_JD_Jan00_UT(:,1),1)
   if tel_date(1) == tab_JD_Jan00_UT(i,1)
```

```
      JD_Jan00_UT = tab_JD_Jan00_UT(i,2) + 2400000;
   end
end

JD_UT1 = JD_Jan00_UT + doy + UT/24 % Julian Date for tel_date

T_UT1 = (JD_UT1 - 2451545.0)/36525; % Julian centuries (J2000)

% Determine beta angle from orbital elements
lam_M_sun = 280.4606184 + 36000.77005361*T_UT1; % mean longitude of the sun in deg
M_sun = 357.5277233 + 35999.05034*T_UT1; % mean anomaly for the sun in deg

lam_M_sun_ = mod(lam_M_sun,360);
M_sun_ = mod(M_sun,360)*pi/180;

lam_eclip = (lam_M_sun_ + 1.914666471*sin(M_sun_) + .019994643*sin(2*M_sun_))*pi/180;
   % ecliptic longitude in rad (= angular seperation vernal equinox/sun)

eps = 23.44*pi/180; % obliquity of the ecliptic

bet = asin(sin(eps)*cos(xi2)*sin(lam_eclip) - cos(eps)*sin(xi2)*cos(xi1)...
   *sin(lam_eclip) + sin(xi2)*sin(xi1)*cos(lam_eclip));

bet*180/pi

for l = 1:3
   bet_a(l,c) = bet;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of solar vector %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Normalized solar vector referred to geocentric inertial axes
r2_S = [sin(lam_eclip)*cos(eps);sin(lam_eclip)*sin(eps);cos(lam_eclip)]

% Sum of argument of perigee and true anomaly
xi3_ = mod(xi3 + tau,2*pi);

% Tranformation of orbit-plane axes to geocentric axes
T_OS_1 = [cos(xi1),0,-sin(xi1);0,1,0;sin(xi1),0,cos(xi1)];
T_OS_2 = [cos(xi2),sin(xi2),0;-sin(xi2),cos(xi2),0;0,0,1];
T_OS_3 = [cos(xi3_),0,-sin(xi3_);0,1,0;sin(xi3_),0,cos(xi3_)];
T_OS_4 = [1,0,0;0,-1,0;0,0,-1];

T_OS = T_OS_4 * T_OS_3 * T_OS_2 * T_OS_1;

% Normalized solar vector referred to orbit-plane axes
r2_O = T_OS * r2_S

% Determine angle to each orbit-plane axis
O1 = [1;0;0];
O2 = [0;1;0];
O3 = [0;0;1];

vec1 = [r2_O(1);r2_O(2);0];
vec2 = [r2_O(1);r2_O(2);0];
vec3 = [r2_O(1);0;r2_O(3)];

ang_r2_O1 = acos(dot(vec1,O1)/(norm(vec1,2)))*180/pi
ang_r2_O2 = acos(dot(vec2,O2)/(norm(vec2,2)))*180/pi
ang_r2_O3 = acos(dot(vec3,O3)/(norm(vec3,2)))*180/pi


end % for-loop (c)

% Plot
scr_sz = get(0,'ScreenSize');
```

```
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);
set(gca,'Position',[.130 .122 .775 .822]);

plot(a_SY_(:,1)*180/pi,bet_a(:,1)*180/pi,'b.-',...
   a_SY_(:,2)*180/pi,bet_a(:,2)*180/pi,'ro-',...
   a_SY_(:,3)*180/pi,bet_a(:,3)*180/pi,'gx-',...
   a_SY_(:,4)*180/pi,bet_a(:,4)*180/pi,'k+-',...
   a_SY_(:,5)*180/pi,bet_a(:,5)*180/pi,'b*-',...
   a_SY_(:,6)*180/pi,bet_a(:,6)*180/pi,'rs-',...
   a_SY_(:,7)*180/pi,bet_a(:,7)*180/pi,'gd-',...
   a_SY_(:,8)*180/pi,bet_a(:,8)*180/pi,'kv-',...
   a_SY_(:,9)*180/pi,bet_a(:,9)*180/pi,'bp-',...
   a_SY_(:,10)*180/pi,bet_a(:,10)*180/pi,'r^-');

grid;

legend(date_stamp(1,:),date_stamp(2,:),date_stamp(3,:),date_stamp(4,:),...
   date_stamp(5,:),date_stamp(6,:),date_stamp(7,:),date_stamp(8,:),...
   date_stamp(9,:),date_stamp(10,:),4);

xlabel('Angles between Axis of Symmetry and Solar Vector in \circ');
ylabel('Beta Angles in \circ');
title(['Angles Geom. Axis/Solar Vector vs. Beta Angle (',filenameB,' - ',...
   filenameE,')'],'Interpreter','none');
```

End of DETATTP.M

DISPATTD.M

```
function DISPATTD();

% DISPATTD
%
% Display of Attitude of PANSAT at Deployment with respect to
% Orbit-plane Reference Frame
%
%
% (c) 1998 by Grasi


%%%%%%%%%%%%%%
% Data given %
%%%%%%%%%%%%%%

% PANSAT deployment at GMT 303/18:45:33, MET 0/23:25:59;

% Zeta angles for geometrical axes of PANSAT

zet1 = 355.3*pi/180;
zet2 = 3.6*pi/180;
zet3 = 308.2*pi/180;


%%%%%%%%%%%%%%%%%%%
% Graphic display %
%%%%%%%%%%%%%%%%%%%

% Axes visibility
vis_ax_B = ’on’;
vis_txt_B = ’on’;

vis_ax_O = ’on’;
vis_txt_O = ’on’;

% Display orbit-plane axes
BtoO = ’on’;

% Visibility of PANSAT’s model
vis_PANSAT = ’on’;

% Visibility of graph
vis_graph = ’off’;

% Location of camera
v_az = 3*37.5; % both angles in deg
v_el = .5*15;


% Figure
scr_sz = get(0,’ScreenSize’);
figure(’Position’,[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);


% Define model of PANSAT (data given in X_, Y_, Z_ - coordinates)

% c.m. is assumed to be located in center of geometry of 26-sided polyhedron,
% i.e., dimension and mass of LVI is not considered for the visualization

% Definition of color for panels
pan_sens = [.96 .16 .53]; % solar panel’s equipped with current sensor (deep pink)
pan_std = [.08 .02 .4]; % standard solar panel, without current sensor (navy blue)
pan_base = [.96 .16 .53]; % solar panel at base plate, SP5 (deep pink)
pan_lvi = [.83 .63 .09]; % LVI (gold)
tri_ldck = [.23 .73 1]; % triangles at lower deck (deep sky blue)
tri_udck = [.23 .73 1]; % triangles at upper deck (deep sky blue)
```

```
% Vertices of model (polyhedron and lvi)
vertx = [-9.3 -21.8 9.3;9.3 -21.8 9.3;9.3 -21.8 -9.3;-9.3 -21.8 -9.3;...
   -9.3 -9.3 21.8;   9.3 -9.3 21.8;21.8 -9.3 9.3;21.8 -9.3 -9.3;...
   9.3 -9.3 -21.8;-9.3 -9.3 -21.8;   -21.8 -9.3 -9.3;-21.8 -9.3 9.3;...
   -9.3 9.3 21.8;9.3 9.3 21.8;21.8 9.3 9.3;21.8 9.3 -9.3;9.3 9.3 -21.8;...
   -9.3 9.3 -21.8;-21.8 9.3 -9.3;-21.8 9.3 9.3;-9.3 21.8 9.3;9.3 21.8 9.3;...
   9.3 21.8 -9.3;-9.3 21.8 -9.3]*1e-2; % in m
vertx_lvi = [0 -26.9 11.7;8.3 -26.9 8.3;11.7 -26.9 0;8.3 -26.9 -8.3;...
   0 -26.9 -11.7;-8.3 -26.9 -8.3;-11.7 -26.9 0;-8.3 -26.9 8.3;0 -21.8 9.1;...
   6.4 -21.8 6.4;9.1 -21.8 0;6.4 -21.8 -6.4;0 -21.8 -9.1;-6.4 -21.8 -6.4;...
   -9.1 -21.8 0;-6.4 -21.8 6.4]*1e-2; % in m

% Faces of model
facs_rec = [1 2 6 5;4 1 12 11;3 4 10 9;2 3 8 7;1 2 3 4;6 7 15 14;5 6 14 13;...
   12 5 13 20;11 12 20 19;10 11 19 18;9 10 18 17;8 9 17 16;7 8 16 15;...
   19 20 21 24;13 14 22 21;21 22 23 24;15 16 23 22;17 18 24 23];
facs_tri = [1 5 12;2 7 6;3 9 8;4 11 10;20 13 21;14 15 22;16 17 23;18 19 24];
facs_lvi = [1 2 10 9;2 3 11 10;3 4 12 11;4 5 13 12;5 6 14 13;6 7 15 14;...
   7 8 16 15;8 1 9 16];
facs_lvi_base = [1 2 3 4 5 6 7 8;9 10 11 12 13 14 15 16];

% Color of faces
col_rec = [pan_std;pan_std;pan_std;pan_sens;tri_ldck;pan_std;pan_sens;...
   pan_std;pan_sens;pan_std;pan_sens;pan_std;pan_sens;pan_sens;pan_std;...
   pan_sens;pan_std;pan_std];
col_tri = [tri_ldck;tri_ldck;tri_ldck;tri_ldck;tri_udck;tri_udck;...
   tri_udck;tri_udck];
col_lvi = [pan_lvi;pan_lvi;pan_lvi;pan_lvi;pan_lvi;pan_lvi;pan_lvi;pan_lvi];

% Handles
yel =[1 .95 0]; % darker yellow
h_rec = patch('Vertices',vertx,'Faces',facs_rec,'FaceVertexCData',col_rec,...
   'FaceColor','flat','EdgeColor',yel,'LineWidth',2,'Visible',vis_PANSAT);
h_tri = patch('Vertices',vertx,'Faces',facs_tri,'FaceVertexCData',col_tri,...
   'FaceColor','flat','EdgeColor',yel,'LineWidth',2,'Visible',vis_PANSAT);
h_lvi = patch('Vertices',vertx_lvi,'Faces',facs_lvi,...
   'FaceVertexCData',col_lvi,'FaceColor','flat','EdgeColor',yel,...
   'LineWidth',2,'Visible',vis_PANSAT);
h_l_base = patch('Vertices',vertx_lvi,'Faces',facs_lvi_base,...
   'FaceVertexCData',pan_base,'FaceColor','flat','EdgeColor',yel,...
   'LineWidth',2,'Visible',vis_PANSAT);


% Body fixed axes of PANSAT (geometrical)
le_B = 1.0;  % length in m
h_B1 = line('XData',[0 le_B],'YData',[0 0],'ZData',[0 0],'Color','b',...
   'LineWidth',1,'Visible',vis_ax_B);
h_B2 = line('XData',[0 0],'YData',[0 le_B],'ZData',[0 0],'Color','r',...
   'LineWidth',1,'Visible',vis_ax_B);
h_B3 = line('XData',[0 0],'YData',[0 0],'ZData',[0 le_B],'Color','g',...
   'LineWidth',1,'Visible',vis_ax_B);

h_B1_t = text(le_B+.1,0,0,'X\prime','Color','k','Visible',vis_txt_B);
h_B2_t = text(0,le_B+.1,0,'Y\prime','Color','k','Visible',vis_txt_B);
h_B3_t = text(0,0,le_B+.1,'Z\prime','Color','k','Visible',vis_txt_B);


% Orbit-plane axes
le_O = 1.2; % length in m
h_O1 = line('XData',[0 le_O],'YData',[0 0],'ZData',[0 0],'Color','b',...
   'LineWidth',2,'Visible',vis_ax_O);
h_O2 = line('XData',[0 0],'YData',[0 le_O],'ZData',[0 0],'Color','r',...
   'LineWidth',2,'Visible',vis_ax_O);
h_O3 = line('XData',[0 0],'YData',[0 0],'ZData',[0 le_O],'Color','g',...
   'LineWidth',2,'Visible',vis_ax_O);

h_O1_t = text(le_O+.1,0,0,'O_{1}','Color','k','Visible',vis_txt_O);
h_O2_t = text(0,le_O+.1,0,'O_{2}','Color','k','Visible',vis_txt_O);
```

```
h_O3_t = text(0,0,le_O+.1,'O_{3}','Color','k','Visible',vis_txt_O);


% Change appearance of axes
axis('square');
axis([-.5 1.1 -.5 1.1 -.5 1.1]);

view(v_az,v_el);

set(gcf,'Color','w');

set(gca,'Color','w','XColor','k','YColor','k','ZColor','k',...
   'XGrid','off','YGrid','off','ZGrid','off','Visible',vis_graph,...
   'XLabel',text('String','x'),'YLabel',text('String','y'),...
   'ZLabel',text('String','z'));

set(gca,'CameraUpVectorMode','manual','CameraUpVector',[0 0 -1]);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine attitude of PANSAT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Tranformation of orbit-plane axes to body axes
T_BO_1 = [cos(zet3),sin(zet3),0;-sin(zet3),cos(zet3),0;0,0,1];
T_BO_2 = [cos(zet2),0,-sin(zet2);0,1,0;sin(zet2),0,cos(zet2)];
T_BO_3 = [1,0,0;0,cos(zet1),sin(zet1);0,-sin(zet1),cos(zet1)];

T_BO = T_BO_3 * T_BO_2 * T_BO_1;


%%%%%%%%%%%%%%%%%
% Show attitude %
%%%%%%%%%%%%%%%%%

% Obtain coordinates of axes and labels

% Get current coordinates of patches of PANSAT model
X_rec = get(h_rec,'XData');
Y_rec = get(h_rec,'YData');
Z_rec = get(h_rec,'ZData');

X_tri = get(h_tri,'XData');
Y_tri = get(h_tri,'YData');
Z_tri = get(h_tri,'ZData');

X_lvi = get(h_lvi,'XData');
Y_lvi = get(h_lvi,'YData');
Z_lvi = get(h_lvi,'ZData');

X_l_base = get(h_l_base,'XData');
Y_l_base = get(h_l_base,'YData');
Z_l_base = get(h_l_base,'ZData');

% Get current coordinates of body axes
X_B1 = get(h_B1,'XData');
Y_B1 = get(h_B1,'YData');
Z_B1 = get(h_B1,'ZData');

X_B2 = get(h_B2,'XData');
Y_B2 = get(h_B2,'YData');
Z_B2 = get(h_B2,'ZData');

X_B3 = get(h_B3,'XData');
Y_B3 = get(h_B3,'YData');
Z_B3 = get(h_B3,'ZData');

% Get current position of body axes labels
POS_B1_t = get(h_B1_t,'Position');
```

```
POS_B2_t = get(h_B2_t,'Position');
POS_B3_t = get(h_B3_t,'Position');

% Display of transformation of body axes to reference axes

% DCM
if strcmp(BtoO,'on') == 1
   A = T_BO.';
end

% Transformation of axes
B1 = A * [X_B1(2);Y_B1(2);Z_B1(2)];
B2 = A * [X_B2(2);Y_B2(2);Z_B2(2)];
B3 = A * [X_B3(2);Y_B3(2);Z_B3(2)];

set(h_B1,'XData',[0 B1(1)],'YData',[0 B1(2)],'ZData',[0 B1(3)]);
set(h_B2,'XData',[0 B2(1)],'YData',[0 B2(2)],'ZData',[0 B2(3)]);
set(h_B3,'XData',[0 B3(1)],'YData',[0 B3(2)],'ZData',[0 B3(3)]);

% Transformation of labels
B1_t = A * POS_B1_t.';
B2_t = A * POS_B2_t.';
B3_t = A * POS_B3_t.';

set(h_B1_t,'Position',B1_t);
set(h_B2_t,'Position',B2_t);
set(h_B3_t,'Position',B3_t);

% Transformation of vertices of PANSAT model
for i = 1:size(X_rec(1,:),2) % columns
   for j = 1:size(X_rec(:,1),1) % rows
      rec = A * [X_rec(j,i);Y_rec(j,i);Z_rec(j,i)];
      X_rec_(j,i) = rec(1);
      Y_rec_(j,i) = rec(2);
      Z_rec_(j,i) = rec(3);
   end
end

for i = 1:size(X_tri(1,:),2) % columns
   for j = 1:size(X_tri(:,1),1) % rows
      tri = A * [X_tri(j,i);Y_tri(j,i);Z_tri(j,i)];
      X_tri_(j,i) = tri(1);
      Y_tri_(j,i) = tri(2);
      Z_tri_(j,i) = tri(3);
   end
end

for i = 1:size(X_lvi(1,:),2) % columns
   for j = 1:size(X_lvi(:,1),1) % rows
      lvi = A * [X_lvi(j,i);Y_lvi(j,i);Z_lvi(j,i)];
      X_lvi_(j,i) = lvi(1);
      Y_lvi_(j,i) = lvi(2);
      Z_lvi_(j,i) = lvi(3);
   end
end

for i = 1:size(X_l_base(1,:),2) % columns
   for j = 1:size(X_l_base(:,1),1) % rows
      l_base = A * [X_l_base(j,i);Y_l_base(j,i);Z_l_base(j,i)];
      X_l_base_(j,i) = l_base(1);
      Y_l_base_(j,i) = l_base(2);
      Z_l_base_(j,i) = l_base(3);
   end
end

set(h_rec,'XData',X_rec_,'YData',Y_rec_,'ZData',Z_rec_);
set(h_tri,'XData',X_tri_,'YData',Y_tri_,'ZData',Z_tri_);
set(h_lvi,'XData',X_lvi_,'YData',Y_lvi_,'ZData',Z_lvi_);
set(h_l_base,'XData',X_l_base_,'YData',Y_l_base_,'ZData',Z_l_base_);
```

```
drawnow
```

End of DISPATTD.M

DISPATTS.M

```
function DISPATTS();

% DISPATTS
%
% Display of Roll-rate of PANSAT with respect to the Sun
% (Note that the Angular Velocity Component Aligned with the
% Solar Vector Cannot Be Detected!)
%
%
% (c) 1998 by Grasi


%%%%%%%%%%%%%%
% Data given %
%%%%%%%%%%%%%%

c = 1

% Files containing status data

if c == 1

   file = 'g:\pansat\data\status\1998-11-09\all_1998-11-09_10-24-38.mat';

   % Parameters after curve fitting
   T = [NaN,NaN,88,87,88,87,NaN,116]; % period
   a0 = [NaN,NaN,.0379,.0191,-.0211,.0306,NaN,.1949]; % shift along y axis (1st sine)
   b0 = [NaN,NaN,.2287,.2669,-.2824,.2387,NaN,-.0322]; % amplitude
   c0 = [NaN,NaN,-1.9626,-3.5644,-8.2818,-.4164,NaN,-7.2803]; % phase shift
   d0 = [NaN,NaN,.0020,-.0031,.0033,-.0058,NaN,0]; % shift along y axis (2nd sine)
   e0 = [NaN,NaN,.0135,.0080,.0113,.0167,NaN,0]; % amplitude

elseif c == 2

   file = 'g:\pansat\data\status\1998-11-17\all_1998-11-17_06-29-17.mat';

   % Parameters after curve fitting
   T = [NaN,NaN,59,59,59,60,NaN,79];
   a0 = [NaN,NaN,-.0257,-.0328,-.0031,-.0270,NaN,.1511];
   b0 = [NaN,NaN,.3140,-.3277,-.2947,.2700,NaN,.0349];
   c0 = [NaN,NaN,2.7382,-1.9363,-3.5570,-1.7698,NaN,-2.6115];
   d0 = [NaN,NaN,-.0044,.0029,-.0021,.0541,NaN,0];
   e0 = [NaN,NaN,-.0082,-.0179,-.0030,-.0051,NaN,0];

elseif c == 3

   file = 'g:\pansat\data\status\1998-11-18\all_1998-11-18_06-23-44.mat';

   % Parameters after curve fitting
   T = [NaN,NaN,59,60,59,59,NaN,80];
   a0 = [NaN,NaN,.0149,-.0471,.0070,.0263,NaN,.0983];
   b0 = [NaN,NaN,-.2949,-.3734,.3174,-.2918,NaN,.0436];
   c0 = [NaN,NaN,-7.9227,-3.0005,-7.9994,-6.3863,NaN,-7.1391];
   d0 = [NaN,NaN,-.0001,.0004,-.0008,-.0098,NaN,0];
   e0 = [NaN,NaN,-.0106,.0005,-.0021,-.0107,NaN,0];

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculate required variables %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Load status data
load(file);
```

```
% Needed variables
N = 0; t = 0; tE = 0;

N = size(time(:,1),1);
t = time(1:N,1)-time(1,1);
tE = t(N);

% Or more samples!
Ts = 3;
t = 0:Ts:tE;
N = length(t)

% Calculate currents at sample points
cur = zeros(N,8);
for j = 1:8
   for l = 1:N
      cur(l,j) = a0(j) + b0(j).*sin(2*pi./T(j).*t(l) + c0(j)) + d0(j)...
         + e0(j).*sin(2*pi./T(8).*t(l) + c0(8));
   end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine solar vector %
%%%%%%%%%%%%%%%%%%%%%%%%%%

% Determine solar vector from solar panel currents

rSX_ = detsolvX(0,cur); % referred to geometrical axes of PANSAT


% Calculate azimuth and elevation of Sun in geometrical frame

for l = 1:N

   aS(l,1) = atan(abs(rSX_(l,1)/rSX_(l,3))); % azimuth

   % Determine azimuth angle ambiguously by observing quadrants
   if sign(rSX_(l,1)) == 1 & sign(rSX_(l,3)) == -1
      aS(l,1) = pi - aS(l,1);
   elseif sign(rSX_(l,1)) == -1 & sign(rSX_(l,3)) == -1
      aS(l,1) = pi + aS(l,1);
   elseif sign(rSX_(l,1)) == -1 & sign(rSX_(l,3)) == 1
      aS(l,1) = 2*pi - aS(l,1);
   end

   dS(l,1) = asin(rSX_(l,2)); % elevation
end


%%%%%%%%%%%%%%%%%%%
% Graphic display %
%%%%%%%%%%%%%%%%%%%

% Visibility of PANSAT's model
vis_PANSAT = 'on';

% Visibility of graph
vis_graph = 'off';

% Visibility of Sun line
vis_sun_l = 'on';

% Visibility of lighting
vis_li = 'on';

% Axes visibility
vis_ax_B = 'on';
vis_txt_B = 'on';
```

```
% Location of camera
v_az = 0.5*37.5; % both angles in deg
v_el = 4*15;


% Figure
scr_sz = get(0,'ScreenSize');
figure('Position',[scr_sz(4)/10 scr_sz(4)/10  scr_sz(3)/1.2 scr_sz(4)/1.2]);


% Define model of PANSAT (data given in X_, Y_, Z_ - coordinates)

% c.m. is assumed to be located in center of geometry of 26-sided polyhedron,
% i.e., dimension and mass of LVI is not considered for the visualization

% Definition of color for panels
pan_sens = [.96 .16 .53]; % solar panel's equipped with current sensor (deep pink)
pan_std = [.08 .02 .4]; % standard solar panel, without current sensor (navy blue)
pan_base = [.96 .16 .53]; % solar panel at base plate, SP5 (deep pink)
pan_lvi = [.83 .63 .09]; % LVI (gold)
tri_ldck = [.23 .73 1]; % triangles at lower deck (deep sky blue)
tri_udck = [.23 .73 1]; % triangles at upper deck (deep sky blue)

% Vertices of model (polyhedron and lvi)
vertx = [-9.3 -21.8 9.3;9.3 -21.8 9.3;9.3 -21.8 -9.3;-9.3 -21.8 -9.3;...
   -9.3 -9.3 21.8;  9.3 -9.3 21.8;21.8 -9.3 9.3;21.8 -9.3 -9.3;...
   9.3 -9.3 -21.8;-9.3 -9.3 -21.8;   -21.8 -9.3 -9.3;-21.8 -9.3 9.3;...
   -9.3 9.3 21.8;9.3 9.3 21.8;21.8 9.3 9.3;21.8 9.3 -9.3;9.3 9.3 -21.8;...
   -9.3 9.3 -21.8;-21.8 9.3 -9.3;-21.8 9.3 9.3;-9.3 21.8 9.3;9.3 21.8 9.3;...
   9.3 21.8 -9.3;-9.3 21.8 -9.3]*1e-2; % in m
vertx_lvi = [0 -26.9 11.7;8.3 -26.9 8.3;11.7 -26.9 0;8.3 -26.9 -8.3;...
   0 -26.9 -11.7;-8.3 -26.9 -8.3;-11.7 -26.9 0;-8.3 -26.9 8.3;0 -21.8 9.1;...
   6.4 -21.8 6.4;9.1 -21.8 0;6.4 -21.8 -6.4;0 -21.8 -9.1;-6.4 -21.8 -6.4;...
   -9.1 -21.8 0;-6.4 -21.8 6.4]*1e-2; % in m

% Faces of model
facs_rec = [1 2 6 5;4 1 12 11;3 4 10 9;2 3 8 7;1 2 3 4;6 7 15 14;5 6 14 13;...
   12 5 13 20;11 12 20 19;10 11 19 18;9 10 18 17;8 9 17 16;7 8 16 15;...
   19 20 21 24;13 14 22 21;21 22 23 24;15 16 23 22;17 18 24 23];
facs_tri = [1 5 12;2 7 6;3 9 8;4 11 10;20 13 21;14 15 22;16 17 23;18 19 24];
facs_lvi = [1 2 10 9;2 3 11 10;3 4 12 11;4 5 13 12;5 6 14 13;6 7 15 14;...
   7 8 16 15;8 1 9 16];
facs_lvi_base = [1 2 3 4 5 6 7 8;9 10 11 12 13 14 15 16];

% Color of faces
col_rec = [pan_std;pan_std;pan_std;pan_sens;tri_ldck;pan_std;pan_sens;...
   pan_std;pan_sens;pan_std;pan_sens;pan_std;pan_sens;pan_sens;pan_std;...
   pan_sens;pan_std;pan_std];
col_tri = [tri_ldck;tri_ldck;tri_ldck;tri_ldck;tri_udck;tri_udck;...
   tri_udck;tri_udck];
col_lvi = [pan_lvi;pan_lvi;pan_lvi;pan_lvi;pan_lvi;pan_lvi;pan_lvi;pan_lvi];

% Handles
yel =[1 .95 0]; % darker yellow
h_rec = patch('Vertices',vertx,'Faces',facs_rec,'FaceVertexCData',col_rec,...
   'FaceColor','flat','EdgeColor',yel,'LineWidth',2,'Visible',vis_PANSAT);
h_tri = patch('Vertices',vertx,'Faces',facs_tri,'FaceVertexCData',col_tri,...
   'FaceColor','flat','EdgeColor',yel,'LineWidth',2,'Visible',vis_PANSAT);
h_lvi = patch('Vertices',vertx_lvi,'Faces',facs_lvi,...
   'FaceVertexCData',col_lvi,'FaceColor','flat','EdgeColor',yel,...
   'LineWidth',2,'Visible',vis_PANSAT);
h_l_base = patch('Vertices',vertx_lvi,'Faces',facs_lvi_base,...
   'FaceVertexCData',pan_base,'FaceColor','flat','EdgeColor',yel,...
   'LineWidth',2,'Visible',vis_PANSAT);


% Lighting properties of model
set(h_rec,'FaceLighting','flat','EdgeLighting','flat','BackFaceLighting','lit');
set(h_tri,'FaceLighting','flat','EdgeLighting','flat','BackFaceLighting','lit');
set(h_lvi,'FaceLighting','flat','EdgeLighting','flat','BackFaceLighting','lit');
set(h_l_base,'FaceLighting','flat','EdgeLighting','flat',...
   'BackFaceLighting','reverselit');

material dull; % dull or shiny for best results


% Body fixed axes of PANSAT (geometrical)
le_B = 0.8;  % length in m
h_B1 = line('XData',[0 le_B],'YData',[0 0],'ZData',[0 0],'Color','b',...
   'LineWidth',1,'Visible',vis_ax_B);
h_B2 = line('XData',[0 0],'YData',[0 le_B],'ZData',[0 0],'Color','r',...
   'LineWidth',1,'Visible',vis_ax_B);
h_B3 = line('XData',[0 0],'YData',[0 0],'ZData',[0 le_B],'Color','g',...
   'LineWidth',1,'Visible',vis_ax_B);

h_B1_t = text(le_B+.1,0,0,'X\prime','Color','y','Visible',vis_txt_B);
h_B2_t = text(0,le_B+.1,0,'Y\prime','Color','y','Visible',vis_txt_B);
h_B3_t = text(0,0,le_B+.1,'Z\prime','Color','y','Visible',vis_txt_B);


% Sun's ray
le_S = 1.2; % length in m
h_l_S = line('XData',[0 0],'YData',[0 0],'ZData',[0 le_S],'Color','y',...
   'LineWidth',2,'Visible',vis_sun_l);


% Add lighting
h_li = light('Color','w','Position',-[0 0 1],...
   'Style','infinite','Visible',vis_li);


% Change appearance of axes
axis('square');
axis([-.5 1.1 -.5 1.1 -.5 1.1]);

view(v_az,v_el);

set(gcf,'Color','k');

set(gca,'Color','k','XColor','w','YColor','w','ZColor','w',...
   'XGrid','off','YGrid','off','ZGrid','off','Visible',vis_graph,...
   'XLabel',text('String','x'),'YLabel',text('String','y'),...
   'ZLabel',text('String','z'),...
   'Title',text('String','Petite Amateur Navy Satellite','Color','y',...
   'FontName','Arial','FontSize',17,'FontWeight','bold'));

set(gca,'CameraUpVectorMode','manual','CameraUpVector',[-1 0 0]);


for l = 1:N

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine attitude of PANSAT %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% DCM for transformation of geometrical axes to Sun directed system
T_XS_X_1 = [cos(aS(l)),0,-sin(aS(l));0,1,0;sin(aS(l)),0,cos(aS(l))];
T_XS_X_2 = [1,0,0;0,cos(dS(l)),-sin(dS(l));0,sin(dS(l)),cos(dS(l))];

T_XS_X_ = T_XS_X_2 * T_XS_X_1;


%%%%%%%%%%%%%%%%%
% Show attitude %
%%%%%%%%%%%%%%%%%
```

```
% Obtain coorďinates of axes and labels for animations

if 1 == 1


% Get current coordinates of patches of PANSAT model
X_rec = get(h_rec,'XData');
Y_rec = get(h_rec,'YData');
Z_rec = get(h_rec,'ZData');

X_tri = get(h_tri,'XData');
Y_tri = get(h_tri,'YData');
Z_tri = get(h_tri,'ZData');

X_lvi = get(h_lvi,'XData');
Y_lvi = get(h_lvi,'YData');
Z_lvi = get(h_lvi,'ZData');

X_l_base = get(h_l_base,'XData');
Y_l_base = get(h_l_base,'YData');
Z_l_base = get(h_l_base,'ZData');

% Get current coordinates of body axes
X_B1 = get(h_B1,'XData');
Y_B1 = get(h_B1,'YData');
Z_B1 = get(h_B1,'ZData');

X_B2 = get(h_B2,'XData');
Y_B2 = get(h_B2,'YData');
Z_B2 = get(h_B2,'ZData');

X_B3 = get(h_B3,'XData');
Y_B3 = get(h_B3,'YData');
Z_B3 = get(h_B3,'ZData');

% Get current position of body axes labels
POS_B1_t = get(h_B1_t,'Position');
POS_B2_t = get(h_B2_t,'Position');
POS_B3_t = get(h_B3_t,'Position');

end % if

% Display of transformation of body axes to reference axes

% DCM
A = T_XS_X_;

% Transformation of axes
B1 = A * [X_B1(2);Y_B1(2);Z_B1(2)];
B2 = A * [X_B2(2);Y_B2(2);Z_B2(2)];
B3 = A * [X_B3(2);Y_B3(2);Z_B3(2)];

set(h_B1,'XData',[0 B1(1)],'YData',[0 B1(2)],'ZData',[0 B1(3)]);
set(h_B2,'XData',[0 B2(1)],'YData',[0 B2(2)],'ZData',[0 B2(3)]);
set(h_B3,'XData',[0 B3(1)],'YData',[0 B3(2)],'ZData',[0 B3(3)]);

% Transformation of labels
B1_t = A * POS_B1_t.';
B2_t = A * POS_B2_t.';
B3_t = A * POS_B3_t.';

set(h_B1_t,'Position',B1_t);
set(h_B2_t,'Position',B2_t);
set(h_B3_t,'Position',B3_t);

% Transformation of vertices of PANSAT model
for i = 1:size(X_rec(1,:),2) % columns
   for j = 1:size(X_rec(:,1),1) % rows
      rec = A * [X_rec(j,i);Y_rec(j,i);Z_rec(j,i)];
```

```
        X_rec_(j,i) = rec(1);
        Y_rec_(j,i) = rec(2);
        Z_rec_(j,i) = rec(3);
    end
end

for i = 1:size(X_tri(1,:),2) % columns
    for j = 1:size(X_tri(:,1),1) % rows
        tri = A * [X_tri(j,i);Y_tri(j,i);Z_tri(j,i)];
        X_tri_(j,i) = tri(1);
        Y_tri_(j,i) = tri(2);
        Z_tri_(j,i) = tri(3);
    end
end

for i = 1:size(X_lvi(1,:),2) % columns
    for j = 1:size(X_lvi(:,1),1) % rows
        lvi = A * [X_lvi(j,i);Y_lvi(j,i);Z_lvi(j,i)];
        X_lvi_(j,i) = lvi(1);
        Y_lvi_(j,i) = lvi(2);
        Z_lvi_(j,i) = lvi(3);
    end
end

for i = 1:size(X_l_base(1,:),2) % columns
    for j = 1:size(X_l_base(:,1),1) % rows
        l_base = A * [X_l_base(j,i);Y_l_base(j,i);Z_l_base(j,i)];
        X_l_base_(j,i) = l_base(1);
        Y_l_base_(j,i) = l_base(2);
        Z_l_base_(j,i) = l_base(3);
    end
end

set(h_rec,'XData',X_rec_,'YData',Y_rec_,'ZData',Z_rec_);
set(h_tri,'XData',X_tri_,'YData',Y_tri_,'ZData',Z_tri_);
set(h_lvi,'XData',X_lvi_,'YData',Y_lvi_,'ZData',Z_lvi_);
set(h_l_base,'XData',X_l_base_,'YData',Y_l_base_,'ZData',Z_l_base_);


drawnow

end % for-loop (1)
```

End of DISPATTS.M

DEPLOY.MWS

```
DEPLOY.MWS, MAPLE V RELEASE 5
Attitude of PANSAT at deployment
(c) 1998 by Grasi
> restart;
> with(linalg):
>
> # data based on PANSAT deployment at MET 0/23:25:59, UT 303:18:45:33
>
> # ECI (M50) to Orbiter (body)
> roll:=evalf(280.45*Pi/180): # roll (+X sense)
> pitch:=evalf(151.33*Pi/180): # pitch
> yaw:=evalf(56.39*Pi/180): # yaw
>
> T1:=matrix(3,3,[cos(pitch),0,-sin(pitch),0,1,0,sin(pitch),0,cos(pitch)]):
> T2:=matrix(3,3,[cos(yaw),sin(yaw),0,-sin(yaw),cos(yaw),0,0,0,1]):
> T3:=matrix(3,3,[1,0,0,0,cos(roll),sin(roll),0,-sin(roll),cos(roll)]):
> T_Orbiter_ECI:=evalm(T3 &* T2 &* T1);
>
> v_Orbiter:=matrix(3,1,[0,0,-1]):
> v_ECI:=evalm(transpose(T_Orbiter_ECI)
>    &* v_Orbiter); # solar vector M50
>
>
> # ECI (M50) to LVLH
> roll:=evalf(117.93*Pi/180): # roll (+X sense)
> pitch:=evalf(208.34*Pi/180): # pitch
> yaw:=evalf(333.18*Pi/180): # yaw
>
> T1:=matrix(3,3,[cos(pitch),0,-sin(pitch),0,1,0,sin(pitch),0,cos(pitch)]):
> T2:=matrix(3,3,[cos(yaw),sin(yaw),0,-sin(yaw),cos(yaw),0,0,0,1]):
> T3:=matrix(3,3,[1,0,0,0,cos(roll),sin(roll),0,-sin(roll),cos(roll)]):
> T_LVLH_ECI:=evalm(T3 &* T2 &* T1);
>
>
> # ECI (M50) to LVLH
> Omega:=evalf(150.83*Pi/180):
> i:=evalf(28.60*Pi/180):
> omega:=evalf(11.67*Pi/180):
> tau:=evalf(320.60*Pi/180):
> om_tau:=omega+tau:
>
> T1:=matrix(3,3,[cos(Omega),sin(Omega),0,-sin(Omega),cos(Omega),0,0,0,1]):
> T2:=matrix(3,3,[1,0,0,0,cos(i),sin(i),0,-sin(i),cos(i)]):
> T3:=matrix(3,3,[cos(om_tau),sin(om_tau),0,-sin(om_tau),cos(om_tau),0,0,0,1]):
> T_:=matrix(3,3,[0,1,0,0,0,-1,-1,0,0]):
> T_LVLH_ECI:=evalm(T_ &* T3 &* T2 &* T1);
>
>
> # LVLH to Orbiter (body)
> roll:=evalf(141.63*Pi/180): # roll (+X sense)
> pitch:=evalf(95.93*Pi/180): # pitch
> yaw:=evalf(359.91*Pi/180): # yaw
>
> T1:=matrix(3,3,[cos(pitch),0,-sin(pitch),0,1,0,sin(pitch),0,cos(pitch)]):
> T2:=matrix(3,3,[cos(yaw),sin(yaw),0,-sin(yaw),cos(yaw),0,0,0,1]):
> T3:=matrix(3,3,[1,0,0,0,cos(roll),sin(roll),0,-sin(roll),cos(roll)]):
> T_Orbiter_LVLH:=evalm(T3 &* T2 &* T1);
>
> v_Orbiter:=matrix(3,1,[0,0,-1]):
> v_ECI:=evalm(transpose(T_LVLH_ECI) &* transpose(T_Orbiter_LVLH) &* v_Orbiter); # solar
vector M50
>
> # Orbiter to PANSAT (geometrical)
> T_X_Orbiter:=matrix(3,3,[0,1,0,0,0,-1,-1,0,0]):
> v_X_:=matrix(3,1,[0,1,0]):
```

```
> v_ECI:=evalm(transpose(T_LVLH_ECI) &* transpose(T_Orbiter_LVLH) &*
transpose(T_X_Orbiter) &* v_X_); # solar vector M50
>
>
> # Solar vector at time of deployment
> epsilon:=evalf(23.44*Pi/180):
> JD_UT1:=2451117.2816; # GMT 303/18:45:33 @ deployment
> T_UT1:=(JD_UT1-2451545.0)/36525;
> lambda_M_sun:=280.4606+36000.7701*T_UT1:
> M_sun:=357.5277+35999.0503*T_UT1:
> lambda_M_sun_:=lambda_M_sun+360:
> M_sun_:=(M_sun+360)*Pi/180:
> lambda_ecl:=evalf((lambda_M_sun_+1.9147*sin(M_sun_)+.0200*sin(2*M_sun_))*Pi/180); #
J2000
>
>
rS_E:=sigma2*matrix(3,1,[sin(lambda_ecl)*cos(epsilon),sin(lambda_ecl)*sin(epsilon),cos(la
mbda_ecl)]); # J2000
> T_ECI_S:=matrix(3,3,[0,0,1,1,0,0,0,1,0]):
> rS_ECI:=evalm(T_ECI_S &* rS_E); # J2000
>
> # Transformation of epochs
> T_J2000_B1950:=matrix(3,3,[.999925679,-.011181483,-.004859004,
>                            .011181483,.999937485,-.000027163,
>                            .004859004,-.000027170,.999988194]):
>
> evalm(inverse(T_J2000_B1950) &* rS_ECI); # solar vector B1950
>
>
> # Beta angle (J2000)
> epsilon:=23.44*Pi/180:
> i:=28.46*Pi/180:
>
> beta:=arcsin(sin(epsilon)*cos(i)*sin(lambda_ecl)-
cos(epsilon)*sin(i)*cos(Omega)*sin(lambda_ecl)+sin(i)*sin(Omega)*cos(lambda_ecl)):
> evalf(%*180/Pi); # beta angle J2000
>
>
> # Geocentric inertial to geometrical PANSAT (J2000)
> xi1:=evalf(151.91*Pi/180):
> xi2:=evalf(28.46*Pi/180):
> xi3_:=evalf(331.72*Pi/180):
>
> T1:=matrix(3,3,[cos(xi1),0,-sin(xi1),0,1,0,sin(xi1),0,cos(xi1)]):
> T2:=matrix(3,3,[cos(xi2),sin(xi2),0,-sin(xi2),cos(xi2),0,0,0,1]):
> T3:=matrix(3,3,[cos(xi3_),0,-sin(xi3_),0,1,0,sin(xi3_),0,cos(xi3_)]):
> T_OQ:=diag(1,-1,-1):
>
> T_OS:=evalm(T_OQ &* T3 &* T2 &* T1);
>
> zeta1:=evalf(355.28*Pi/180):
> zeta2:=evalf(3.61*Pi/180):
> zeta3:=evalf(308.23*Pi/180):

> T_3:=matrix(3,3,[cos(zeta3),sin(zeta3),0,-sin(zeta3),cos(zeta3),0,0,0,1]):
> T_2:=matrix(3,3,[cos(zeta2),0,-sin(zeta2),0,1,0,sin(zeta2),0,cos(zeta2)]):
> T_1:=matrix(3,3,[1,0,0,0,cos(zeta1),sin(zeta1),0,-sin(zeta1),cos(zeta1)]):
>
> T_X_O:=evalm(T_1 &* T_2 &* T_3);
>
> T_X_S:=evalm(T_X_O &* T_OS);
>
> v_S:=evalm(transpose(T_X_S) &* v_X_); # solar vector J2000
> v_S:=evalm(transpose(T_J2000_B1950) &* transpose(T_X_S) &* v_X_); # solar vector B1950
>
>
> # Solar vector referred to orbit-plane axes (J2000)
> r2_O:=evalm(T_OS &* r2_S);
>
```

```
> r2_O_:=matrix(3,1,[.7777618040,.6229351894,0]):
> O1:=vector([1,0,0]):
> evalf(angle(convert(r2_O_,vector),O1)*180/Pi); # beta angle (J2000)
>
> # Axis Y_ referred to orbit-plane (should be pointing to sun)
> Y_:=matrix(3,1,[0,1,0]):
> O2:=evalm(transpose(T_X_O) &* Y_);
> evalf(angle(convert(O2,vector),O1)*180/Pi); # beta angle (J2000)
>
```

End of DEPLOY.MWS

BODYAXES.MWS

```
BODYAXES.M,  MAPLE V RELEASE 5
Determination of Principal Axes of PANSAT by Solving Eigenvalue Problem
(c) 1998 by Grasi
>
> restart;
> with(linalg): with(plots):
>
DETERMINATION OF BODY AXES
>
Data given
Coordinates of center of mass (c.m.) referred to PANSAT's X, Y, Z-System as determined at
NASA GSFC
> xs:=.1e-2;
> ys:=23.2e-2;
> zs:=-.2e-2;
>
Moments of inertia (referred to X_,Y_,Z_-system; based on CAD model)
> Ix_:=1.299;
> Iy_:=.976;
> Iz_:=1.421;
> Ixy_:=-.001;
> Iyz_:=.003;
> Izx_:=.003;
>
Inertia tensor (referred to X_,Y_,Z_-system)
> Iyx_:=Ixy_: Izy_:=Iyz_: Ixz_:=Izx_:
> I_:=matrix([[Ix_,-Ixy_,-Ixz_],[-Iyx_,Iy_,-Iyz_],[-Izx_,-Izy_,Iz_]]);
>
Transformation to principal moments of inertia (eigenvalue problem)
Eigenvalues (correspond to principal moments of inertia)
> EW:=eigenvals(I_);
>
Eigenvectors (correspond to principal axes)
> EV:=eigenvects(I_);
>
Arranging eigenvalues in order of quantity and relate eigenvectors to eigenvalues
> lam:=[seq(EV[i][1],i=1..3)]:
> ew:=-sort(-lam);
> for i from 1 to 3 do
>    for j from 1 to 3 do
>       if ew[i] = EV[j][1] then
>          ev[i]:=normalize(EV[j][3][1]);
>          print(ev[i]);
>       fi;
>    od;
> od;
>
Plot of principal axes system (correspond to body axes B1, B2, B3)
> data_ev:=seq(eval(ev[i]),i=1..3);
> p_ev:=pointplot3d([data_ev],color=cyan,shading=none):
> data1:=seq([ev[1][1]*j,ev[1][2]*j,ev[1][3]*j],j=0..1):
> p1:=pointplot3d([data1],style=line,color=blue):
> data2:=seq([ev[2][1]*j,ev[2][2]*j,ev[2][3]*j],j=0..1):
> p2:=pointplot3d([data2],style=line,color=red):
> data3:=seq([ev[3][1]*j,ev[3][2]*j,ev[3][3]*j],j=0..1):
> p3:=pointplot3d([data3],style=line,color=green):
> pl_0:=pointplot3d([-xs,-ys,-zs],color=red):
>
display(p_ev,p1,p2,p3,pl_0,axes=normal,scaling=constrained,labels=[X_,Y_,Z_],orientation=
[30,80]);
>
display(p_ev,p1,p2,p3,pl_0,axes=normal,scaling=constrained,labels=[X_,Y_,Z_],orientation=
[0,90]);
>
display(p_ev,p1,p2,p3,pl_0,axes=normal,scaling=constrained,labels=[X_,Y_,Z_],orientation=
[90,90]);
```

```
>
display(p1,p2,p3,pl_0,axes=normal,scaling=constrained,labels=[X_,Y_,Z_],orientation=[90,0
]);
>
Calculation of rotation angles (between X_,Y_,Z_-system and body axes)
> theta_x1:=angle(vector([ev[1][2],ev[1][3]]),vector([0,1]));
> theta_x2:=angle(vector([ev[3][2],ev[3][3]]),vector([-1,0]));
> theta_x:=(theta_x1+theta_x2)/2; evalf(%*180/Pi);
>
> theta_y1:=angle(vector([ev[1][1],ev[1][3]]),vector([0,1]));
> theta_y2:=angle(vector([ev[2][1],ev[2][3]]),vector([-1,0]));
> theta_y:=(theta_y1+theta_y2)/2; evalf(%*180/Pi);
>
> theta_z1:=angle(vector([ev[3][1],ev[3][2]]),vector([0,-1]));
> theta_z2:=angle(vector([ev[2][1],ev[2][2]]),vector([-1,0]));
> theta_z:=(theta_z1+theta_z2)/2; evalf(%*180/Pi);
>
```

End of BODYAXES.MWS

CURVFIT.MWS

```
CURVFIT.MWS, MAPLE V RELEASE 5
Curve Fitting of Solar Panel Currents Using Least Squares Methods
(c) 1998 by Grasi
>
> restart;
> with(linalg): with(plots):
>
Load file
> fn:=fopen(`_cf.txt`,READ):
> N:=; # number of samples
> for i from 1 to N do
>    lst_[i]:=fscanf(fn,`%d %f %f %f %f %f %f %f`);
> od:
> fclose(fn);
> lst:=convert(lst_,array):
>
Input parameters (period, weights)
> n_cur:=; # current of interest
> # SP7 (n_cur = 3)
> T[4]:=:
> W4:=matrix(1,N,[]):
>
> # SP9 (n_cur = 4)
> T[5]:=:
> W5:=matrix(1,N,[]):
>
> # SP11 (n_cur = 5)
> T[6]:=:
> W6:=matrix(1,N,[]):
>
> # SP13 (n_cur = 6)
> T[7]:=:
> W7:=matrix(1,N,[]):
>
> # SP16 (n_cur = 8)
> T[9]:=181:
> W9:=matrix(1,N,[]):
>
> W:=matrix(9,N,0):
> copyinto(W4,W,4,1):
> copyinto(W5,W,5,1):
> copyinto(W6,W,6,1):
> copyinto(W7,W,7,1):
> copyinto(W9,W,9,1):
>
> for k from 1 to N do
>    y_[k]:=lst[k][n]:
>    t_[k]:=lst[k][1]:
> od:
>
>
> n:=n_cur+1:
> y:=convert(y_,array);
> t:=convert(t_,array);
>
> f:=; # one (f = 0) or two (f = 1) sines to fit
>
Least squares algorithm
> if f = 0 then
>    J:=sum(W[n,j]*(y[j]-a0-b0*sin((2*Pi/T[n])*t[j]+c0))^2,j=1..N):
>    eq1:=diff(J,a0):
>    eq2:=diff(J,b0):
>    eq3:=diff(J,c0):
>
> elif f = 1 then
>    a0:=; c0:=; f0:=; # determined by fitting one sine (f = 0)
```

```
>
> J:=sum(W[n,j]*(y[j]-a0-b0*sin((2*Pi/T[n])*t[j]+c0)-d0-
e0*sin((2*Pi/T[9])*t[j]+f0))^2,j=1..N):
>    eq2:=diff(J,b0):
>    eq4:=diff(J,d0):
>    eq5:=diff(J,e0):
> fi:
>
> if f = 0 then
>    sol:=fsolve({eq1,eq2,eq3},{a0,b0,c0});
> elif f = 1 then
>    sol:=fsolve({eq2,eq4,eq5},{b0,d0,e0});
> fi;
>
Plot results (fitted current)
> data_t:=seq(t[i],i=1..N):
> data_y:=seq(y[i],i=1..N):
>
> dat_pl1:=seq([data_t[k],data_y[k]],k=1..N):
> pl1:=plot([dat_pl1],style=line,color=red):
> dat_pl2:=seq([W[n,k]*data_t[k],W[n,k]*data_y[k]],k=1..N):
> pl2:=plot([dat_pl2],style=point,color=black):
> dat_pl3:=seq([data_t[k],data_yf[k]],k=1..N):
> pl3:=plot([dat_pl2],style=point,color=blue):
>
> if f = 0 then
>
> data_yf:=seq(subs(sol,a0)+subs(sol,b0)*sin(2*Pi/T[n]*t[j]+subs(sol,c0)),j=1..N):
>
pl4:=plot(subs(sol,a0)+subs(sol,b0)*sin(2*Pi/T[n]*q+subs(sol,c0)),q=1..t[N],color=blue):
> elif f = 1 then
>
data_yf:=seq(subs(sol,a0)+subs(sol,b0)*sin(2*Pi/T[n]*t[j]+subs(sol,c0))+subs(sol,d0)+subs
(sol,e0)*sin(2*Pi/T[9]*t[j]+subs(sol,f0)),j=1..N):
>
pl4:=plot(subs(sol,a0)+subs(sol,b0)*sin(2*Pi/T[n]*q+subs(sol,c0))+subs(sol,d0)+subs(sol,e
0)*sin(2*Pi/T[9]*q+subs(sol,f0)),q=1..t[N],color=blue):
>
> fi:
>
> d:=0;
> if d = 0 then
>    display(pl1,pl2,pl3,pl4);
> elif d = 1 then
>    display(pl1,pl4);
> fi;
>
```

End of CURVFIT.MWS

[This page is intentionally left blank.]

GLOSSARY OF SYMBOLS

Latin Symbols

a	semi-major axis	m
a_0	vertical shift (of a sine)	-
A	area	m^2
b	semi-minor axis	m
b_0	amplitude (of a sine)	-
B	body axis	-
B^*	ballistic coefficient and perigee density	m^{-1}
$\mathbf{B}$	body axis vector	-
	linear momentum vector	N s
c_0	phase shift (of a sine)	-
C	proportional factor	1
C_D	drag coefficient	1
d	differential	-
	distance	m
d_0	vertical shift (of a sine)	-
∂	partial differential	-
e	eccentricity	1
e_0	amplitude (of a sine)	-
$\mathbf{e}$	eigenvector	-
	axis of rotation	-
E	eccentric anomaly	rad
	total energy	J
$\mathbf{E}$	identity matrix	-
f	frequency	Hz (s^{-1})
f_S	sampling frequency	Hz (s^{-1})
	scale factor	1

f_0	phase shift (of a sine)	-
F	external force	N
F	Earth-Centered, Earth-Fixed (ECEF) axis	-
$\mathbf{F}$	external force vector	N
	ECEF axis vector	-
g	acceleration at the Earth's surface	m/s^2
H	topocentric horizon axis	-
$\mathbf{H}$	vector	-
i	index	1
I	moment of inertia	$kg\ m^2$
	current	A
I_d	inherent degradation	1
$\mathbf{I}$	inertia tensor	$kg\ m^2$
J	Bessel function	-
	loss function	-
JD	Julian Date	days
K	Earth's gravitational constant	m^3/s^2
l	length	m
L	angular momentum	Nm s
$\mathbf{L}$	angular momentum vector	Nm s
m	mass	kg
M	external torque	Nm
	mean anomaly	rad
$\mathbf{M}$	vector of external torque	Nm
n	mean motion	s^{-1}
$\dot{n}$	mean motion rate	s^{-2}
$\ddot{n}$	mean motion acceleration	s^{-3}
N	upper boundary (in a sum)	1
	number of samples	1
O	orbit-plane axis	-

$\mathbf{O}$	vector of orbit-plane axes	-
q_1	vector component of a quaternion	-
q_2	vector component of a quaternion	-
q_3	vector component of a quaternion	-
q_4	scalar part of a quaternion	-
$\mathbf{q}$	vector part of a quaternion	-
$\mathbf{Q}$	quaternion	-
$\overline{\mathbf{Q}}$	skew-symmetric matrix	-
r	radius	m
$\mathbf{r}$	displacement vector from c.m. to c.p.	m
	position vector	m
$\mathbf{r}_S$	solar vector	m
S	solar constant	W/m^2
S	geocentric inertial axis	-
$\mathbf{S}$	vector	-
t	time	s
T	period	s
	kinetic energy	J
T_{UT1}	number of Julian centuries	-
$\mathbf{T}$	direction cosine matrix (DCM)	-
T_S	sampling period	s
T_O	orbit period	s
u	unit vector component	-
$\mathbf{u}$	unit vector (first axis direction)	-
U	voltage	V
	potential energy	J
UT	universal time (mean solar time at Greenwich)	hh:mm:ss
UTC	coordinated universal time	hh:mm:ss
$UT0$	universal time (uncorrected)	hh:mm:ss
$UT1$	universal time (corrected for $UT0$)	hh:mm:ss

v	velocity	m/s
	unit vector component	-
$\mathbf{v}$	velocity vector	m/s
	unit vector (second axis direction)	-
w	unit vector component	-
	weight factor	1
$\mathbf{w}$	unit vector (third axis direction)	
x	coordinate variable	m
	coordinate axis (M50, LVLH or orbiter)	-
$\mathbf{x}$	vector	-
X	x axis of a coordinate system fixed to the satellite	-
$\mathbf{X}$	vector	-
X'	x axis of a coordinate system fixed to satellite with origin at c.m.	-
$\mathbf{X'}$	vector	-
y	coordinate variable	m
	coordinate axis (M50, LVLH or orbiter)	-
y_t	true value	-
Y	y axis of a coordinate system fixed to the satellite	-
$\mathbf{Y}$	vector	-
Y'	y axis of a coordinate system fixed to satellite with origin at c.m.	-
$\mathbf{Y'}$	vector	-
z	coordinate variable	m
	coordinate axis (M50, LVLH or orbiter)	-
Z	z axis of a coordinate system fixed to the satellite	-
$\mathbf{Z}$	vector	-
Z'	z axis of a coordinate system fixed to satellite with origin at c.m.	-
$\mathbf{Z'}$	vector	-

Greek Symbols

α	Euler angle	°, rad

	azimuth	°, rad
α_0	angle between observer's meridian and line of equinoxes	°, rad
β	beta angle (angle between orbit-plane and Sun line)	°, rad
γ	angle	°, rad
Γ	sun incidence angle	°, rad
δ	angle	°, rad
	elevation	°, rad
$\bar{\delta}$	mean angle	°, rad
Δ	delta (difference in sth.)	-
ε	obliquity of the ecliptic	°, rad
ζ	deviation angle of orbit-plane axis from body axis	°, rad
$\boldsymbol{\zeta}$	vector	°, rad
$\dot{\zeta}$	uncoupled angular velocity of body axis	°/s, rad/s
$\dot{\boldsymbol{\zeta}}$	vector	°/s, rad/s
$\ddot{\zeta}$	uncoupled angular acceleration of body axis	$°/s^2$, rad/s^2
$\ddot{\boldsymbol{\zeta}}$	vector	$°/s^2$, rad/s^2
η	rotation angle (for coordinate transformation)	°, rad
	efficiency	1
ϑ	temperature	°C
θ	angle in equatorial plane	rad
λ	eigenvalue	-
	longitude	°, rad
ξ	Euler angle to transform geocentric inertial to orbit-plane axis	°, rad
ξ_1	right ascension of the ascending node (RAAS)	°, rad
ξ_2	orbit inclination	°, rad
ξ_3	argument of perigee	°, rad
$\bar{\xi}_3$	$= \xi_3 + \tau$	°, rad

$\dot{\xi}_1$	change in right ascension of the ascending node (RAAS)	°/s^2, rad/s^2
$\dot{\xi}_2$	change in orbit inclination	°/s^2, rad/s^2
$\dot{\xi}_3$	change in argument of perigee	°/s^2, rad/s^2
$\dot{\bar{\xi}}_3$	$= \dot{\xi}_3$	°/s^2, rad/s^2
π	3.1415...	-
ρ	atmospheric density	kg/m^3
σ	distance from center of Earth to satellite (c.m.)	m
τ	true anomaly	rad
φ_0	geodetic (geographic) latitude	°/s, rad/s
Φ	angle of rotation (eigenvalue)	rad
ω	angular velocity	°/s, rad/s
	radian frequency	rad/s
ω_E	Earth's mean angular rotation	rad/solar s
$\dot{\omega}$	angular acceleration	°/s^2, rad/s^2
$\boldsymbol{\omega}$	angular velocity vector	°/s, rad/s

Superscripts

c.m.	center of mass
c.p.	center of pressure
F	referred to inertial system
T	transpose (of a vector, matrix)
'	referred to c.m.
	rotated position of an axis/vector
"	rotated position of an axis/vector after two successive rotations

Subscripts

a	aerodynamic (force, torque)
B_i	along body axes

BO	transformation from orbit-plane axes to body axes
BR	body axes referred to orbit-plane axes
BX'	transformation from geometrical axes (ref. to c.m.) to body axes
B1950	B1950 epoch
CV	cover (panel)
ecl	ecliptic
E	(referred to) Earth
ECI	Earth-centered inertial
EPS	electrical power system
f	fitted (estimated) value by a least squares method
g	gravity-gradient (force, torque)
GST	Greenwich sidereal time
h	horizontal
HS	transformation of geocentric inertial to topocentric horizon system
i	index
J2000	J2000 epoch
LDCK	lower deck (panel)
LST	local sidereal time
LVI	launch vehicle interface
LVLH	local-vertical, local horizontal
m	measured
MS	mass storage
M50	ECI system referred to mean equinox of 1950
n	normalized
N	Nyquist
O	orbit
	orbit-plane (reference system)
OA	overall
OS	transformation of geocentric inertial to orbit-plane axes
Out	output

P	pitch
PH	polyhedron
PYR	pitch-yaw-roll (sequence)
rot	rotational
R	reference system
	roll
S	solar, Sun
	sampling
	(referred to) geocentric inertial system
SCB	spacecraft bus
SP	solar panel
t	true
trans	translational
TMUX	temperature multiplexer
TP	top (panel)
UDCK	upper deck (panel)
*UT*1	universal time (corrected of *UT*0)
X'	(referred to) geometrical axes
Y	yaw
Y'	(referred to) geometrical axes
Z'	(referred to) geometrical axes
0	at 0 h 0 min 0 s
ω	(referred to) angular velocity
I	Figure E.1
II	Figure E.2

LIST OF REFERENCES

1. Wertz, J. R.; Larson, W. J. (eds.), *Space Mission Analysis and Design*, Kluwer Academic Publishers, Dordrecht/Boston/London 1991.

2. Wertz, J. R. (eds.), *Spacecraft Attitude Determination and Control*, Kluwer Academic Publishers, Dordrecht/Boston/London 1990.

3. Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, The McGraw-Hill Companies, Inc., New York 1997.

4. Hughes, P. C., *Spacecraft Attitude Dynamics*, John Wiley & Sons, New York 1986.

5. Kaplan, M. H., *Modern Spacecraft Dynamics & Control*, John Wiley & Sons, New York 1976.

6. Chobotov, V. A. (ed.), *Orbital Mechanics*, American Institute of Aeronautics and Astronautics, Inc., Washington 1991.

7. Leondes, C. T. (ed.), *Guidance and Control of Aerospace Vehicles*, McGraw-Hill Book Company, Inc., New York 1963.

8. Thomson, W. T., *Introduction to Space Dynamics*, John Wiley & Sons, New York/London 1961.

9. Geyling, F. T.; Westerman, H. R., *Introduction to Orbital Mechanics*, Addison-Wesley Publishing Company, Reading, MA, 1971.

10. Baker, R. M. L., Jr; Makemson, M. W., *An Introduction to Astrodynamics, Second Edition*, Academic Press, New York/London 1967.

11. Magnus, K., *Kreisel, Theorie und Anwendungen*, Springer Verlag, Berlin/Heidelberg/New York 1971.

12. Lunderstaedt, R., *Beitraege zur optimalen Steuerung von drallstabilisierten Flugkoerpern*, Dissertation am Institut fuer Mesz- und Regelungstechnik, Universitaet Karlsruhe 1970.

13. Ross, I. M., *Introduction to Orbital Mechanics*, Course Script, Naval Postgraduate School, Monterey, CA, 1998.

14. Kolve, D. I., *Describing an Attitude*, 16th Annual AAS Guidance and Control Conference, American Astronautical Society Publication Office, San Diego, CA, 1993.

15. Cuff, D. J., *Lifetime and Reentry Prediction for the Petite Amateur Navy Satellite*, Department of Aeronautics and Astronautics, Naval Postgraduate School, Monterey, CA, 1994.

16. *Shuttle Orbiter/Cargo Standard Interfaces*, ICD-2-19001, Revision K 1993.

17. *The Astronomical Almanac for the Year 1998*, Nautical Almanac Office United States Naval Observatory, U.S. Government Printing Office, Washington 1997.

18. Seidelmann, P. K., *Explanatory Supplement to the Astronomical Almanac*, University Science Books, Mill Valley, CA, 1992.

19. Beer, F. P.; Johnston, E.. R., Jr., *Vector Mechanics for Engineers: Statics and Dynamics, Second Edition*, McGraw-Hill Book Company, New York 1972.

20. Goldstein, H., *Classical Mechanics*, Addison-Wesley Publishing Company, Reading, MA, 1950.

21. Chebotarev, G. A., *Analytical and Numerical Methods of Celestial Mechanics*, American Elsevier Publishing Company, Inc., New York 1976.

22. Gross, D.; Hauger, W.; Schnell, W., *Technische Mechanik, Bd. 1: Statik*, Springer Verlag, Berlin/Heidelberg/New York 1995.

23. Hauger, W.; Schnell, W.; Gross, D., *Technische Mechanik, Bd. 3: Kinetik*, Springer Verlag, Berlin/Heidelberg/New York 1995.

24. Lunderstaedt, R., *Diskrete Systeme*, Vorlesungsmanuskript, Universitaet der Bundeswehr, Hamburg 1990.

25. Lunderstaedt, R., *Stochastik*, Vorlesungsmanuskript, Universitaet der Bundeswehr, Hamburg 1990.

26. Rothe, H., *Mesztechnik*, Vorlesungsmanuskript, Universitaet der Bundeswehr, Hamburg 1997.

27. Nahi, N. E., *Estimation Theory and Applications*, John Wiley & Sons, Inc., New York 1969.

28. Bronstein, I. N.; Semendjajew, K. A., *Taschenbuch der Mathematik*, B. G. Teubner Verlagsgesellschaft, Stuttgart/Leipzig und Verlag Nauka, Moskau 1991.

29. Hanselmann, D.; Littlefield, B., *Mastering MATLAB: A Comprehensive Tutorial and Reference*, Prentice-Hall, Upper Saddle River, NJ, 1996.

30. Marchand, P., *Graphics and GUIs with MATLAB*, CRC Press, Boca Raton, FL, 1996.

31. Kofler, M., *Maple V Release 4: Einfuehrung und Leitfaden fuer den Praktiker*, Addison-Wesley Publishing Company, Bonn 1996.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
8725 John J. Kingman Rd., STE 0944
Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library, Code 52 2
Naval Postgraduate School
Monterey, CA 93943-5100

3. Research Office, Code 09 1
Naval Postgraduate School
Monterey, CA 93943-5000

4. Markus Graswald 2
Zscherbener Straße 4
06124 Halle
Germany

5. Chairman, Space Systems Academic Group 2
Naval Postgraduate School
Code (SP)
777 Dyer Rd., Rm. 200
Monterey, CA 93943-5110

6. Prof. Don Danielson 1
Naval Postgraduate School
Code (MA)
Monterey, CA 93943