

## Article

# Integrated Navigation System Design for Micro Planetary Rovers: Comparison of Absolute Heading Estimation Algorithms and Nonlinear Filtering

Muhammad Ilyas <sup>1</sup>, Beomjin Hong <sup>1</sup>, Kuk Cho <sup>2</sup>, Seung-Ho Baeg <sup>2,\*</sup> and Sangdeok Park <sup>2</sup>

<sup>1</sup> Department of Robotics and Virtual Engineering, Korea University of Science and Technology (UST), Daejeon 305-333, Korea; milyasmeo@kitech.re.kr (M.I.); hbj723@kitech.re.kr (B.H.)

<sup>2</sup> Robotics R & BD Group, Korea Institute of Industrial Technology (KITECH), Ansan 426-791, Korea; googi33@kitech.re.kr (K.C.); sdpark@kitech.re.kr (S.P.)

\* Correspondence: shbaeg@kitech.re.kr; Tel.: +82-31-8040-6272

Academic Editor: Stefano Mariani

Received: 23 March 2016; Accepted: 17 May 2016; Published: 23 May 2016

**Abstract:** This paper provides algorithms to fuse relative and absolute microelectromechanical systems (MEMS) navigation sensors, suitable for micro planetary rovers, to provide a more accurate estimation of navigation information, specifically, attitude and position. Planetary rovers have extremely slow speed (~1 cm/s) and lack conventional navigation sensors/systems, hence the general methods of terrestrial navigation may not be applicable to these applications. While relative attitude and position can be tracked in a way similar to those for ground robots, absolute navigation information is hard to achieve on a remote celestial body, like Moon or Mars, in contrast to terrestrial applications. In this study, two absolute attitude estimation algorithms were developed and compared for accuracy and robustness. The estimated absolute attitude was fused with the relative attitude sensors in a framework of nonlinear filters. The nonlinear Extended Kalman filter (EKF) and Unscented Kalman filter (UKF) were compared in pursuit of better accuracy and reliability in this nonlinear estimation problem, using only on-board low cost MEMS sensors. Experimental results confirmed the viability of the proposed algorithms and the sensor suite, for low cost and low weight micro planetary rovers. It is demonstrated that integrating the relative and absolute navigation MEMS sensors reduces the navigation errors to the desired level.

**Keywords:** planetary rovers navigation; absolute heading angle estimation; multi-sensor fusion; Sun sensor; Extended Kalman filter; Unscented Kalman filter

## 1. Introduction

Planetary rovers are wheeled robots characterized by extremely slow speed and rotation. They typically move in a stop-and-go fashion so that they can scan their surroundings to find a drivable path, and discover scientific targets in the exploration area. For example, NASA's Mars Exploration Rovers have an average speed of 1 cm/s when moving on hard and flat surfaces and even less in normal motion [1]. Typical motion scenarios of planetary rovers include straight drives, arc turns and turn-in-place maneuvers, *etc.* [2].

To explore an unknown environment, the rover must know two things: “where am I?” and at the same time, “where am I heading to?” The answers to these questions are essential to allow the manned/unmanned robot to perform reliably and not become lost in a given environment. Navigation would appear to be easy for planetary rovers due to their slow dynamics, but unfortunately the sensors that are available for terrestrial robot navigation are not always available onboard the planetary rovers. In terrestrial applications the traditional methods of estimating the attitude and position of mobile

robots include dead reckoning, use of a global positioning system (GPS), magnetic compass and map matching, *etc.* However, none of these methods alone can solve the problem of self-localization on another planet. For example, a GPS-like system is not available on the Moon or Mars yet. Magnetic compasses commonly employed for absolute heading estimations on Earth cannot be used on other planets. For instance, the magnetic field on Mars and Moon is too weak and irregular to be useful as a navigational aid [3,4]. Similarly, gyro-compassing, which is a common method of seeking the north by high-grade IMUs using Earth's rotation [5], is not suitable on many other planets, due, for example, to the extremely slow rotation of the Moon around its own axis. Digital maps are not available or have too low resolution to provide accurate navigation information for planetary rovers. All these limitations ultimately make planetary rovers navigation in a completely unknown environment difficult. Therefore, there is an urgent need to solve the dual problems of relative and absolute navigation on remote planets.

Relative (local) navigation information can be obtained in a way similar to that used for terrestrial ground vehicles, e.g., through integration of angular rates to get attitude and displacement from wheel encoders [6]. However, absolute (global) navigation information, which provides the navigational information of a vehicle directly in global coordinates and does not diverge with the passage of time, is difficult to obtain on a planetary surface. Another reason that absolute navigational information is needed in the navigating framework of a planetary rover is to limit the unbounded error growth of a dead reckoning positioning system. Absolute heading is especially essential to reduce cross track errors in mobile robots [7].

The field of planetary rover navigation has been widely researched since the launch of the Path-finder mission by NASA in 1997. A Sun sensor was developed in [8] to provide planetary rovers with heading estimations, but the algorithm was computationally too burdensome to be used on actual rovers. Further improvements in design and usage were provided in [9]. A charge-coupled device (CCD) type sensor was used in [10] which has a wide FOV and was modeled as a fish-eye lens. It involves a complex calibration procedure with 21 parameters. An algorithm to estimate absolute heading and relative position using a Sun sensor, Earth sensor and an inclinometer was provided in [11]. The attitude estimation algorithm was described from multiple vector observations using optimization based on Davenport's *q-Method* [12]. However, the authors validate their propositions only through simulations, and provide no real data from field experimentation. They also do not consider all possible motion scenarios, as was done in the present study.

A position sensitive detector (PSD) type Sun sensor, similar to ours, has been used in [13]. The PSD type Sun sensor uses less power and simple algorithm to detect the Sun's position in the sensor frame, and suffers less atmospheric effects and is quite robust. The authors also provided an algorithm for estimating absolute heading and absolute position using the Sun sensor and an inclinometer, employing the *q-Method*. The accuracy of position estimation was quite coarse. They achieved absolute heading angle within a few degrees but used a nonlinear minimization approach, which requires much more time as compared to our approach. Furthermore, the optimization based methods can only be employed in a static condition [14], and requires the vehicle to stop for few minutes [15] during the collection of the sensor's multiple observations. In contrast, our approach is a *one-shot* heading estimation method, called Estimation of absolute Attitude using Sun sensor and Inclinometer (*EASI*), and can be performed in real time at 10 Hz (determined by the Sun sensor's update rate), using the Sun sensor and inclinometer data only.

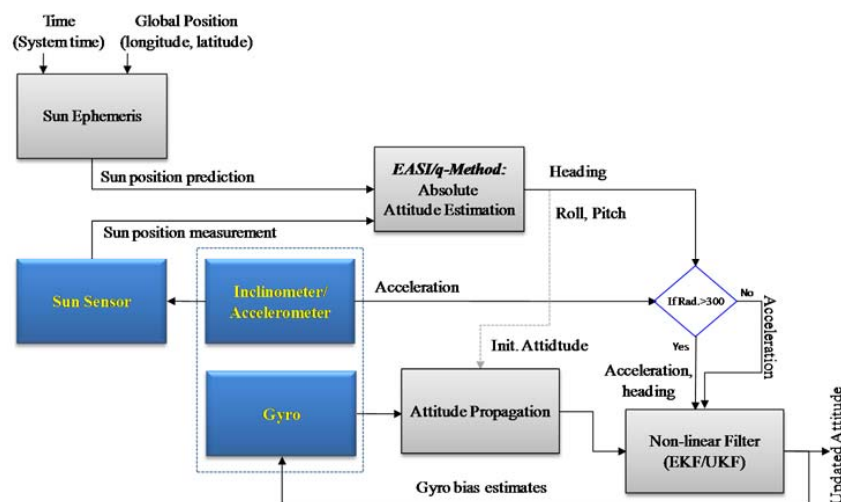
The Sun is the most prominent navigation beacon for space applications. We have designed an absolute heading estimation algorithm which employs low cost, low power and low volume MEMS sensors, suitable for micro rovers (<20 kg). Efforts to employ such miniaturized navigation sensors are already occurring [16,17]. Further, to achieve reliable and more accurate navigation information at a high data rate, we have integrated the relative and absolute navigation sensors in a framework of nonlinear filters. We only chose and compared non-linear filters which can be executed with the limited computation resources of micro planetary rovers, e.g., the Extended Kalman filter (EKF),

Unscented Kalman filters (UKF), etc. Other non-linear filters which have high computational expense (e.g., Particle filters, Divided difference filters, etc.) are not considered in this work, for the sake of simplicity and applicability.

The aim of this work is to achieve reliable and accurate navigation estimation within the desired goals of a planetary rover, e.g., with less than 10% error in position and  $2^\circ$  in attitude [18], taking into account all possible motion scenarios, including long time stationary, move in straight and curved paths, and take turn-in-place (*TiP*) motion [2], etc.

The following are the main elements of this study. First we provide an algorithm (*EASI*) to estimate the absolute heading angle using MEMS sensors, specifically, a Sun sensor and inclinometer. We integrate the relative and absolute navigation sensors to achieve an improvement in navigation accuracy and reliability. Our integration filter integrates the gyro, Sun sensor and inclinometer for long term attitude accuracy, and hence stabilizes the estimation of position. When the rover stops, we use the Sun sensor and inclinometer data to estimate absolute heading and roll, pitch angles. During in-motion state, the integration filter fuses data from the gyro, Sun sensor, and inclinometer to continuously track the attitude. Drifts in gyro-only attitude are bounded by the absolute information of the *EASI* algorithm. Next, the optimization based heading estimation algorithm (*q-Method*) and *EASI* algorithm are compared in different motion scenarios, and finally the nonlinear fusion filters (EKF/UKF) are compared in terms of accuracy and reliability. The aim is to show the viability of using minimal/low-cost MEMS sensors in a micro rover navigation system having very limited space and power resources.

The conceptual diagram of the relative and absolute sensors integration algorithm is shown in Figure 1. At the end of this work, we will show the benefits gained by integrating the relative and absolute navigation sensor data, and compare two nonlinear filters, using real experimental data.



**Figure 1.** Algorithm block diagram for integrating relative and absolute navigation sensors for attitude estimation.

The organization of this paper is as follows: in Section 2, we present the attitude initialization and propagation algorithm, and in Section 3, two absolute attitude estimation algorithms are provided in detail. Section 4 deals with multi-sensor fusion and description of the hardware used in this work. In Section 5, we present the experimental setup, comparisons of the results and discussion. And finally in Section 6, we conclude this paper with some comments on future research directions.

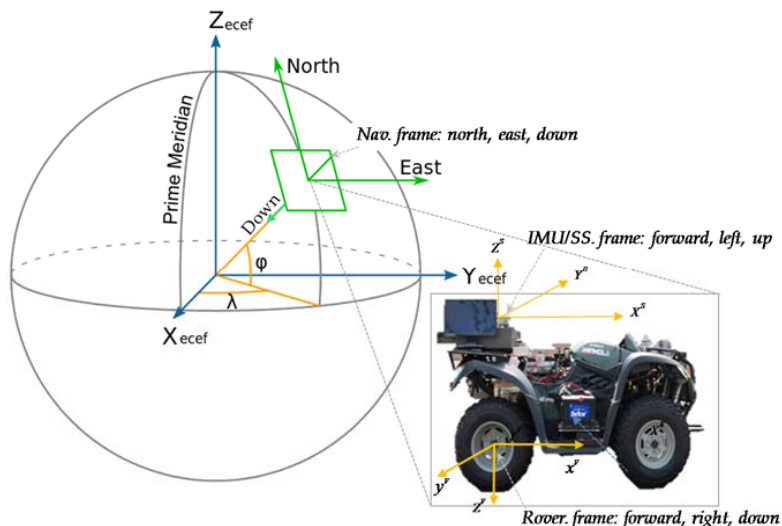
## 2. Attitude Estimation in Planetary Environment

Keeping track of attitude is important to ensure accurate localization; heading accuracy is especially effective at reducing cross track errors and hence enhances the accuracy of overall

localization [19]. Although gyroscopes can be used for such purpose, attitude propagation using only gyros requires mathematical integration of the angular rate signals obtained from the sensor. This causes the noise in the gyro signal to be integrated as well, along with the actual angular rate signal. Moreover, the gyro bias drift will contribute to calculated attitude divergence with time propagation. This is the inherent limitation of gyro-only attitude estimation, and occurs no matter how accurate and expensive a gyro sensor is used. To address this task, first we have to define the different coordinate frames which are used to represent navigation information in this work.

### 2.1. Coordinate Frames

Different sensors are located with different orientations and provide observations in their respective coordinates. In order to use vector observations in a common frame to produce meaningful results, one has to perform transformations among the different coordinate frames. In this work, we have defined the coordinate frames in perspective of planetary rover's navigation, as shown in Figure 2. The navigation frame (n-frame) is most conventionally defined as North-East-Down (NED) frame, IMU and Sun sensor frames are defined in forward, left and up directions. However, the rover frame corresponds to the definition of aerospace vehicles, e.g., by forward, right and down axes. Transformation matrices used in this work are given in our previous work [20].



**Figure 2.** Illustration of the reference frames used for Planetary Rovers.

### 2.2. Attitude Initialization

For a planetary rover, the tilt angles (*i.e.*, roll, pitch) may be initialized by an inclinometer-based angle estimation in a static or quasi-static condition. Since it is not possible to use magnetic or gyro-compasses on most of the other planets, estimating absolute heading is more challenging in a planetary environment. We present an algorithm that uses the Sun sensor and inclinometer data to estimate absolute heading, and then initialize our navigation algorithm with the heading estimated by this algorithm.

#### 2.2.1. Initialization of Roll and Pitch Angles using Inclinometer

In a static condition an inclinometer senses gravitational acceleration and hence can be used to calculate tilt angles. As gravity in a local tangent n-frame can be transformed, using transformation matrix  $C_n^b$ , to a b-frame by compensating with roll ( $\phi$ ), pitch ( $\theta$ ) and yaw ( $\psi$ ) angles, Equation (1) gives the gravity vector in the b-frame. Since the same quantity is measured by the inclinometer in the b-frame, we can calculate the roll and pitch angles, Equations (2) and (3), between these two

frames [21] by comparing the L.H.S and R.H.S of Equation (1), where M samples of the inclinometer data are averaged over the stationary time period.

$$\begin{pmatrix} f_{ib,x}^b \\ f_{ib,y}^b \\ f_{ib,z}^b \end{pmatrix} = \mathbf{g}^b = \mathbf{C}_n^b \mathbf{g}^n; \quad \therefore \mathbf{C}_n^b = \begin{bmatrix} c \theta c \psi & c \theta s \psi & -s \theta \\ -c \phi s \psi + s \phi s \theta c \psi & c \phi c \psi + s \phi s \theta s \psi & s \theta c \phi \\ s \phi s \psi + s \phi s \theta c \psi & -s \phi c \psi + c \phi s \theta s \psi & c \phi c \theta \end{bmatrix} \quad (1)$$

$$\therefore \mathbf{C}_n^b \in \mathbb{R}^{3 \times 3}, \mathbf{g}^n \in \mathbb{R}^{3 \times 1}; c(\cdot) \equiv \cos(\cdot), s(\cdot) \equiv \sin(\cdot)$$

$$Roll \equiv \phi = \tan^{-1} \left( \frac{\bar{f}_{ib,y}^b}{\bar{f}_{ib,z}^b} \right) \quad (2)$$

$$Pitch \equiv \theta = \tan^{-1} \left( \frac{\bar{f}_{ib,x}^b}{\sqrt{(\bar{f}_{ib,y}^b)^2 + (\bar{f}_{ib,z}^b)^2}} \right); \quad \therefore \bar{f}_{ib,xyz}^b = \frac{1}{M} \sum_{j=1}^M \bar{f}_{ib,j}^b \quad (3)$$

### 2.2.2. Initialization of Heading Angle Using Sun Sensor and Inclinometer

As mentioned previously, obtaining absolute heading angle with respect to true north on a remote planet surface is a hard problem, due to the lack of conventional heading sensors that are available on Earth or near the Earth's surface. Sun observations have long been used for navigational purposes because the Sun is a very reliable navigational beacon, especially in a planetary environment where there is no atmospheric interference. We initialize our heading estimate using the Sun sensor and inclinometer data (*i.e.*, with the *EASI* algorithm or *q-Method*). Details of these algorithms are provided in Section 4.

### 2.3. Relative Attitude Estimation Using IMU

There are several methods for expressing the orientation of objects relative to some reference. However, the three most common approaches used are: Euler angles, Direction cosines matrix (DCM) and Quaternion based attitude representation. Euler angles have 12 different formulations, so their sequence of rotation must be defined first. Moreover Euler angles have mathematical singularity (Gimbal lock) that exists when the pitch angle approaches  $90^\circ$ . But for most of the practical platforms in use, especially for planetary rovers, as in this work, this poses no real threat. Positive aspect of Euler angles is the common familiarization of attitude representation with roll, pitch and yaw angles. The DCM is complex to use but is stable. It involves nine elements to be estimated. Quaternions are simplest to use and have no singularities. The only downside is that quaternions are regarded as mysterious and conceptually difficult to understand.

The Euler angle can represent the attitude of a platform from three successive rotations of the navigation frame to the rover frame. In this work, the navigation frame is rotated and fitted into the rover frame in the sequence of yaw, pitch and roll. The transformation of the gyro sensed angular velocity to the Euler angular velocity can be achieved by the following relation [22]:

$$\dot{\Psi}_k = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)/\cos(\theta) \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (4)$$

$$\dot{\Psi}_k = [\mathbf{E}_b^n] \boldsymbol{\omega}_{ib}^b$$

where  $\Psi$  represents attitude vector and  $\boldsymbol{\omega}_{ib}^b = [\omega_x \ \omega_y \ \omega_z]^T$  is the gyro output and matrix  $\mathbf{E}_b^n$  relates the rover body angular velocity to the rate of change of Euler angles. Note that the solution of Equation (4) requires the integration of the first-order nonlinear equation. In discrete time, attitude propagation can be done by simple numerical integration, *i.e.*

$$\begin{bmatrix} \phi_k \\ \theta_k \\ \psi_k \end{bmatrix} = \begin{bmatrix} \phi_{k-1} \\ \theta_{k-1} \\ \psi_{k-1} \end{bmatrix} + \int_{k-1}^k \begin{bmatrix} \dot{\phi}_k \\ \dot{\theta}_k \\ \dot{\psi}_k \end{bmatrix} dt \quad (5)$$

$$\Psi_k = \Psi_{k-1} + \dot{\Psi}_k \Delta t; \Psi_k \in \mathbb{R}^{3 \times 1} \quad (6)$$

### 3. Absolute Attitude Update Using Sun Sensor and Inclinometer in a Planetary Environment

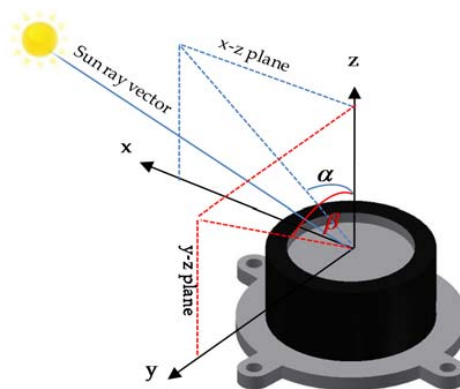
Gyroscopic-based attitude propagation suffers from divergence with time due to the presence of noise in the sensor signals, and consequently this method cannot be used for estimating attitude for operations over a long period of time. The unbounded growth in error in attitude that occurs when it is solely determined by gyroscope must be limited by some absolute reference; otherwise the estimation will become useless after some time, depending on the accuracy of the gyros used.

In near-Earth applications, we have many options for obtaining the absolute heading of the vehicle, but there is a limited number of heading estimation sensors in a planetary rover application. The Sun is the most prominent navigation beacon in the sky and as the technology develops, use of the Sun for navigating unmanned vehicles in space is becoming more practical.

In this section we briefly present two absolute heading estimation algorithms: (i) the Estimation of Absolute heading using a Sun sensor and Inclinometer (*EASI*), developed in our previous work [20]; and (ii) an optimization-based algorithm (*q-Method*) [11,13]. Both algorithms incorporate information from a Sun sensor and an inclinometer, and produce the sensor's absolute roll, pitch angles and heading angle with respect to true north at a particular instant in time, date and location on the surface of the planet during operation.

#### 3.1. The Sun Position Tracking and Sun Ephemeris

Absolute heading estimation algorithms rely on accurate tracking of the Sun's position by some optical sensor, such as a CCD image sensor or a PSD type sensor, which measures the position of a light spot in one or two dimensions on the sensor's surface. Using any kind of optical sensor, the goal is to track the Sun position, *i.e.*, the Sun's azimuth and elevation angles, as it sweeps across the horizon. The Sun ephemeris is a tabulated dataset based on mathematical models of the motion of the Sun and Earth. Given a date and time, these tables also provide a very precise Sun position in the inertial frame. After some transformations, the Sun position can be predicted in the navigation frame [23]. The PSD type Sun sensors, as used in this work and shown in Figure 3, measure the incidence angle of a Sun ray in both azimuth and elevation based on a quadrant photodetector device.



**Figure 3.** CAD diagram of a PSD type Sun sensor's (ISS-D60) reference frame and output angles.

The sunlight is guided to the detector through a window above the sensor. Depending on the angle of incidence, the sunlight induces photocurrents in the four quadrants of the detector. The angle



$\alpha$  is an angle between the z-axis and the projection of the Sun ray on the x-z plane. Similarly, the angle  $\beta$  is an angle between the z-axis and the projection of the Sun ray on the y-z plane. These two angles are also referred to as Angle-x and Angle-y, respectively, in the Sensor's data sheet [24].

### 3.2. Estimation of Absolute Heading Using Sun Sensor and Inclinometer (EASI Algorithm)

The Sun sensor used in this work does not directly give the Sun's position. However, the Sun's azimuth and elevation angles can be calculated from the sensor's output angles (*i.e.*, angle  $\alpha$ , angle  $\beta$ ) using simple trigonometry. The mathematical details of the EASI algorithm are given in our previous work [20]. The EASI algorithm is described here briefly in Algorithm 1, below. The relation between the Sun's position and the rover's absolute heading angle is depicted in Figure 4.

---

**Algorithm 1.** Estimation of Absolute heading angle using Sun sensor and Inclinometer (EASI algorithm)

---

1. Get Sun ray vector from sensor output angles.

$$\mathbf{S}^s = \begin{bmatrix} S_x & S_y & S_z \end{bmatrix}^T$$

2. Transform  $\mathbf{S}^s$  to rover frame.

$$\mathbf{S}^b = \begin{bmatrix} S_x^b \\ S_y^b \\ S_z^b \end{bmatrix} = C_s^b \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} \therefore C_s^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

3. Transform  $\mathbf{S}^b$  to local-flat frame.

$$\mathbf{S}^n = \begin{bmatrix} S_x^n \\ S_y^n \\ S_z^n \end{bmatrix} = C_b^n \begin{bmatrix} S_x^b \\ S_y^b \\ S_z^b \end{bmatrix} \therefore C_b^n = \begin{bmatrix} \cos\theta & \sin\phi\sin\theta & \cos\phi\sin\theta \\ 0 & \cos\phi & -\sin\theta \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix}$$

4. Get Sun's azimuth ( $\alpha^{SS}$ ) and elevation angles ( $\zeta^{SS}$ ) from  $\mathbf{S}^n$  vector as:

$$\alpha^{SS} = \text{atan2}(S_y^n, S_x^n)$$

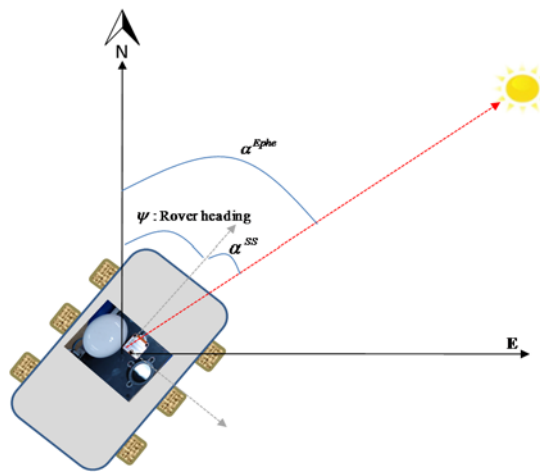
$$\zeta^{SS} = -\sin^{-1}(S_z^n)$$

5. Using local date, time and location (latitude, longitude), get Sun's azimuth ( $\alpha^{Ephe.}$ ) and elevation ( $\zeta^{Ephe.}$ ) angles from Ephemeris models.

6. Determine rover's absolute heading by comparing the Sun's position in steps 4 and 5.

$$\psi_{ss} = \begin{cases} \alpha^{Ephe} - \alpha^{SS} & \text{if } (\alpha^{Ephe} > \alpha^{SS}) \\ \alpha^{SS} - \alpha^{Ephe} & \text{otherwise} \end{cases}$$


---



**Figure 4.** Illustration of the relation between rover heading angle obtained from the Sun position using the Sun sensor, and from Sun ephemeris data.  $\alpha^{Ephe}$  is the Sun's azimuth angle from the ephemeris data,  $\alpha^{SS}$  is the Sun azimuth angle from the Sun sensor and  $\psi_{ss}$  is the rover's heading angle [20].

### 3.3. Optimization-Based Absolute Attitude Estimation (*q-Method*)

Optimization based heading estimation is another method for estimating absolute roll, pitch and heading angles based on multiple Sun observations and gravity vector observations. It uses at least a pair of Sun and gravity vectors and applies optimization techniques to minimize the cost function. However, in practice many observations must be collected from both sensors before applying optimization technique. The result is a transformation matrix between two observation frames, in this case, the Planet-centered Planet-fixed frame (F-frame) and the Sun sensor frame (S-frame). Using this approach, as explained in [11,15] we obtain prediction vectors, namely the Sun position and gravity vector, in the F-frame from the Sun ephemeris data, and from the gravity model, respectively, as in Equation (7):

$$\begin{aligned} \mathbf{S}^F &= \mathbf{C}_I^F \mathbf{S}^I \\ \mathbf{g}^F &= \mathbf{C}_n^F \mathbf{g}^n \end{aligned} \quad (7)$$

where  $\mathbf{S}^I$  and  $\mathbf{g}^n$  are vectors in the inertial frame (I-frame) and n-frame. The corresponding measurements of the Sun position ( $\mathbf{S}^s$ ) and gravity vector ( $\mathbf{g}^s$ ) are obtained from the Sun sensor and inclinometer in the sensor frame. Note that the transformation between the inclinometer and Sun sensor frame is assumed to be a unity matrix, *i.e.*, both sensors are assumed to be perfectly aligned. After we capture the predicted and measured vectors in two different frames, it simply becomes Wahba's attitude estimation problem based on vector observations [25]. The aim is to minimize the cost function, which relates two sets of observation vectors ( $\mathbf{V}, \mathbf{W}$ ), *i.e.*:

$$J(\mathbf{C}_F^S) = \frac{1}{2} \sum_{i=1}^m a_i ||\mathbf{V} - \mathbf{C}_F^S \mathbf{W}||^2 \quad (8)$$

Many methods are available to estimate attitude from a multiple vector observation, *e.g.*, QUEST, FOAM, Davenport's *q-Method* [12], *etc.* By applying any of these methods, we can get the rotations matrix between two observation frames. In addition, this rotation matrix is used to find the absolute tilt angles of the rover with respect to a local level frame, and heading angle from true north.

To start with, we concatenate the predicted and observed vectors and apply Davenport's *q-Method* for rotation estimation. This method is explained with mathematical details in [15]. The output of this rotation estimation algorithm is a matrix ( $\mathbf{C}_F^S$ ) which describes the rotation between two observation frames. Figure 5 and Algorithm 2 show the block diagram and mathematical flow of *q-Method*.

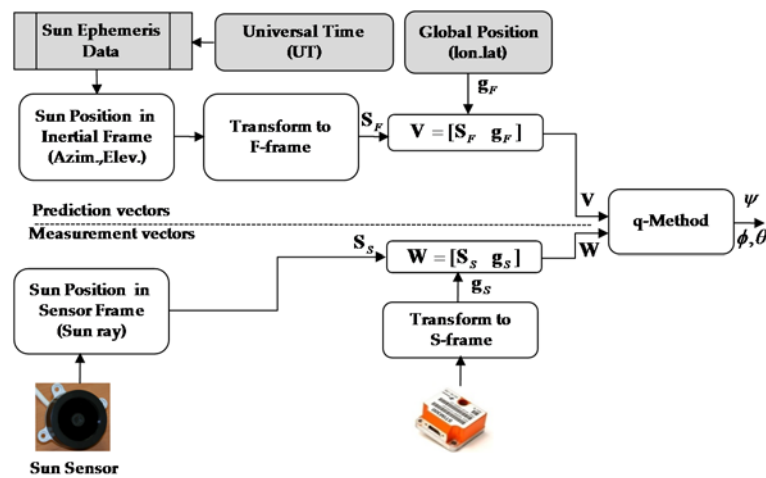


Figure 5. Optimization based attitude estimation (*q-Method*) algorithm block diagram.



**Algorithm 2.** Optimization based absolute attitude estimation algorithm (*q-Method*)

1. Concatenate observation vectors:  $\mathbf{W} = [\mathbf{S}_S \quad \mathbf{g}_S]$   $\therefore$  measurement vectors in S-frame  
 $\mathbf{V} = [\mathbf{S}_F \quad \mathbf{g}_F]$   $\therefore$  prediction vectors in F-frame
2. Calculate:  $\mathbf{B} = \mathbf{W}\mathbf{V}^T$ ;  $\mathbf{Q} = \mathbf{B} + \mathbf{B}^T$
3. Extract  $\mathbf{Z}$  from  $\mathbf{B}$  matrix:  $\mathbf{Z} = [B_{23} - B_{32}B_{31} - B_{13}B_{12} - B_{21}]$
4. Form  $4 \times 4$  matrix  $\mathbf{K}$ :  $\mathbf{K}_{4 \times 4} = \begin{bmatrix} \mathbf{Q} - \mathbf{I}\sigma & \mathbf{Z}^T \\ \mathbf{Z} & \sigma \end{bmatrix}$   $\therefore \sigma = \text{tr}(\mathbf{B}), \mathbf{I} = \mathbf{I}_{3 \times 3}$
5. Find Eigen data of  $\mathbf{K}$ . The Eigenvector corresponding to the largest Eigen value of  $\mathbf{K}$  is  $\mathbf{q}$  vector:  
 $\mathbf{q}_{4 \times 1} = \begin{bmatrix} \mathbf{q}_v \\ q_s \end{bmatrix}$ ;  $4 \times 1$  unit quaternion representing best fit rotation from F to S.
6. Desired rotation matrix can be found from quaternion vector:  
 $\mathbf{C}_F^S = (q_s^2 - \mathbf{q}_v^T \mathbf{q}_v) \mathbf{I} + 2\mathbf{q}_v \mathbf{q}_v^T - 2q_s \mathbf{q}_v^\times$   $\therefore \mathbf{q}_v^\times = \begin{bmatrix} 0 & -q_{v3} & q_{v2} \\ q_{v3} & 0 & -q_{v1} \\ -q_{v2} & q_{v1} & 0 \end{bmatrix}$   
 $\mathbf{q}_v^\times$ : skew-symmetric matrix form of  $\mathbf{q}_v$ .
7. Attitude angles extraction:  
 $\mathbf{C}_S^T = \mathbf{C}_F^T \mathbf{C}_S^F$ , a transformation matrix (function of RPY) from b-frame to n-frame.  
 $\therefore \mathbf{C}_F^T$  is function of longitude, latitude (rover position).
8. Hence roll, pitch and heading angles can be derived from matrix ( $\mathbf{C}_S^T$ ) as:  
 $\phi = \text{asin}(\mathbf{C}_{S32}^T, \mathbf{C}_{S33}^T)$   
 $\theta = \text{asin}(\mathbf{C}_{S31}^T)$   
 $\psi = \text{atan2}(\mathbf{C}_{S21}^T, \mathbf{C}_{S11}^T)$

**4. Multi-Sensor Data Fusion: Design and Comparison of Non-linear Filters**

The attitude of the rover is propagated as given in Equations (5) in Section 2.3. Two nonlinear filters, the Extended Kalman filter and Unscented Kalman filter, were designed and compared in pursuit of better accuracy in this nonlinear environment. In designing EKF and UKF, we assumed low dynamic acceleration of the planetary rovers, which is a valid assumption for slow speed planetary rovers, and utilized the acceleration data from the inclinometer/accelerometer directly in the measurement equation. Hence both the process model and measurement model are nonlinear, involving sinusoidal terms. First, we present the predicted and updated parts of the EKF and UKF filters.

**4.1. Extended Kalman Filter and Unscented Kalman Filter Algorithm****4.1.1. Process Model**

We choose the state vector as follows:

$$\mathbf{x}_k = [\Psi \quad \mathbf{b}_{xyz}^g]^T \quad (9)$$

where  $\Psi = [\phi \quad \theta \quad \psi]^T$  are roll, pitch, yaw angles and  $\mathbf{b}_{xyz}^g$  represents the three gyro bias components. The discrete time nonlinear process model to be used for attitude propagation in EKF/UKF is given below:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k; \quad \mathbf{w}_k \sim \mathbb{N}(0, \mathbf{Q}_k) \quad (10)$$

where,  $\mathbf{x}_k \in \mathbb{R}^n$  is  $n \times 1$  state vector,  $\mathbf{w}_k \in \mathbb{R}^q$  is  $q \times 1$  state noise process vector, with covariance matrix  $\mathbf{Q}_k$ . The system Jacobean matrices, which are used in the EKF algorithm, are computed as:

$$\mathbf{F}_k = \left. \frac{\partial \mathbf{f}_k}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^+}; \mathbf{G}_k = \left. \frac{\partial \mathbf{f}_k}{\partial \mathbf{w}} \right|_{\hat{\mathbf{x}}_k^+} \quad (11)$$

Then, the time prediction step in EKF is:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^- &= \mathbf{f}_k(\hat{\mathbf{x}}_k^+, \mathbf{u}_k) \\ \mathbf{P}_{k+1}^- &= \mathbf{F}_k \mathbf{P}_k^+ \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \end{aligned} \quad (12)$$

The time prediction step in UKF does not require calculation of the Jacobean matrices, instead it generates sigma points ( $\hat{\mathbf{x}}_k^{(i)}$ ) based on the a priori state estimate and associated covariance matrix [26]. The UKF algorithm then propagates these sigma points directly through the nonlinear system dynamics, as in Equation (13):

$$\begin{aligned} \hat{\mathbf{x}}_k^{(i)} &= \hat{\mathbf{x}}_k^+ + \chi^i, \quad i = 1, \dots, 2N \\ \therefore \chi^i &= \left( \sqrt{N \mathbf{P}_k^+} \right)_i^T, \quad i = 1, \dots, N \\ \therefore \chi^{N+i} &= - \left( \sqrt{N \mathbf{P}_k^+} \right)_i^T, \quad i = 1, \dots, N \\ \hat{\mathbf{x}}_{k+1}^{(i)} &= \mathbf{f}_k(\hat{\mathbf{x}}_k^{(i)}, \mathbf{u}_k) \end{aligned} \quad (13)$$

Then, the transformed sigma points are combined to capture the mean and covariance of the state:

$$\hat{\mathbf{x}}_{k+1}^- = \frac{1}{2N} \sum_{i=1}^{2N} \hat{\mathbf{x}}_{k+1}^{(i)} \mathbf{P}_{k+1}^- = \frac{1}{2N} \sum_{i=1}^{2N} (\hat{\mathbf{x}}_{k+1}^{(i)} - \hat{\mathbf{x}}_{k+1}^-)(\hat{\mathbf{x}}_{k+1}^{(i)} - \hat{\mathbf{x}}_{k+1}^-)^T + \mathbf{Q}_k \quad (14)$$

#### 4.1.2. Measurement Update in EKF and UKF

In a measurement update, due to the low dynamics of the planetary rover, it is appropriate to embed the acceleration output directly in the measurement equation. The measurement model, using acceleration from the inclinometer and heading angle calculated in the *EASI/q-Method* algorithms, is given below:

$$\begin{aligned} \begin{bmatrix} f_x \\ f_x \\ f_x \\ \psi_m \end{bmatrix} &= \begin{bmatrix} \mathbf{C}_n^b \mathbf{g}^n \\ \psi_{ss} \end{bmatrix} + \mathbf{v} \Rightarrow \begin{bmatrix} f_x \\ f_x \\ f_x \\ \psi_m \end{bmatrix} = \begin{bmatrix} -g \sin \theta \\ g \sin \phi \cos \theta \\ g \cos \phi \cos \theta \\ \psi_{ss} \end{bmatrix} + \mathbf{v} \\ \mathbf{y}_{k+1} &= h(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1}; \mathbf{v}_{k+1} \sim \mathbb{N}(0, \mathbf{R}_k) \end{aligned} \quad (15)$$

where,  $\mathbf{y}_k \in \mathbb{R}^m$  is  $m \times 1$  observation vector and  $\mathbf{v}_k \in \mathbb{R}^r$  is  $r \times 1$  measurement noise process vector, with covariance matrix  $\mathbf{R}_k$ . Further it is assumed that:  $E[\mathbf{w}_k \mathbf{w}_j^T] = \delta_{kj} \mathbf{Q}_k$ ,  $E[\mathbf{v}_k \mathbf{v}_j^T] = \delta_{kj} \mathbf{R}_k$ ,  $E[\mathbf{v}_k \mathbf{w}_j^T] = 0$ ,  $\forall k, j$ .

The EKF will require measurement of the Jacobean matrix calculated as below:

$$\begin{aligned} \mathbf{H}_{k+1} &= \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k+1}^-} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \\ \text{where, } \mathbf{H}_{11} &= \begin{bmatrix} 0 & -g \cos(\theta) \\ g \cos(\phi) \cos(\theta) & -g \sin(\phi) \sin(\theta) \\ -g \sin(\phi) \cos(\theta) & g \cos(\phi) \sin(\theta) \end{bmatrix}; \mathbf{H}_{12} = [0_{3 \times 4}]; \mathbf{H}_{21} = [0_{1 \times 2}]; \mathbf{H}_{22} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (16)$$

The measurement update step of EKF proceeds as usual:

$$\begin{aligned} \mathbf{K}_{k+1} &= \mathbf{P}_{k+1}^{xy} / \mathbf{P}_{k+1}^{vv} \\ \hat{\mathbf{x}}_{k+1}^+ &= \hat{\mathbf{x}}_{k+1}^- + \mathbf{K}_{k+1} [\mathbf{y}_{k+1} - \hat{\mathbf{y}}_{k+1}] \\ \mathbf{P}_{k+1}^+ &= (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1}^- \end{aligned} \quad (17)$$

where,  $\mathbf{P}_{k+1}^{xy} = \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T$  and  $\mathbf{P}_{k+1}^{vv} = \mathbf{H}_{k+1} \mathbf{P}_{k+1}^- \mathbf{H}_{k+1}^T + \mathbf{R}_{k+1}$ .

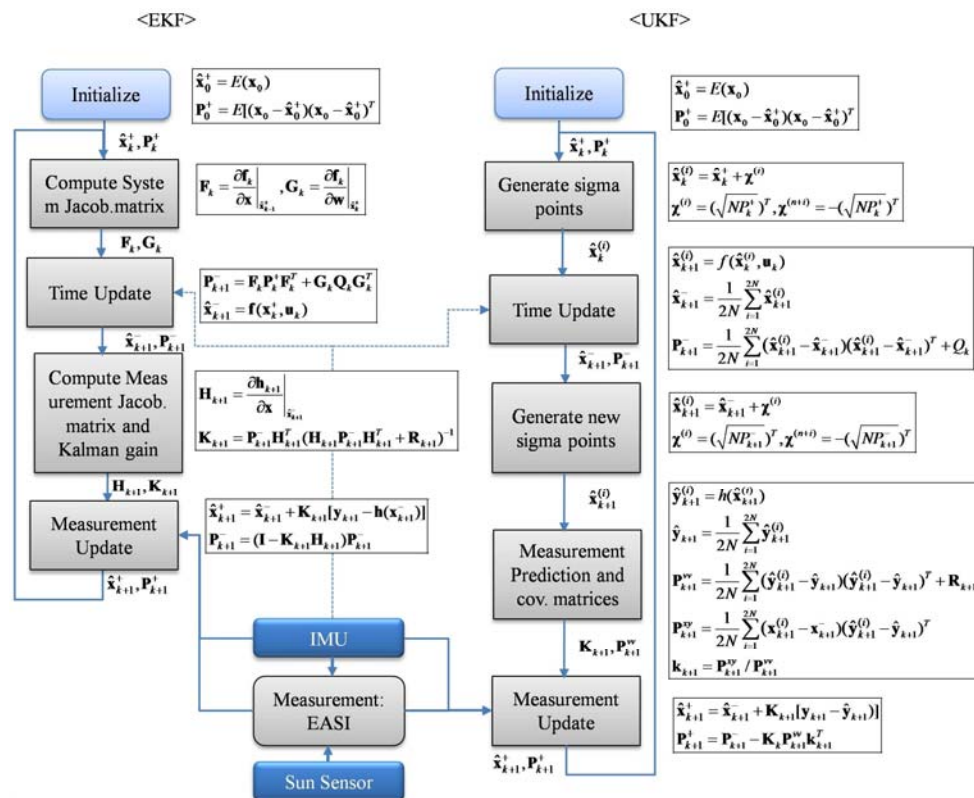
For the measurement update step of UKF, we can calculate sigma points from the freshly predicted state vector and transform the sigma points through the measurement model directly. The transformed sigma points are combined to produce the predicted measurement vector at time  $k + 1$ :

$$\begin{aligned}\hat{\mathbf{y}}_{k+1}^{(i)} &= \mathbf{h}(\hat{\mathbf{x}}_{k+1}^{(i)}) \\ \hat{\mathbf{y}}_{k+1}^- &= \frac{1}{2N} \sum_{i=1}^{2N} \hat{\mathbf{y}}_{k+1}^{(i)}\end{aligned}\quad (18)$$

To update the state and associated covariance matrix, the UKF algorithm proceeds as follows. It first calculates the covariance ( $\mathbf{P}_{k+1}^{vv}$ ) of the predicted measurement vector and adds the noise covariance matrix to account for sensor measurement noise. Then, the cross covariance matrix ( $\mathbf{P}_{k+1}^{xy}$ ) between the predicted state and predicted measurement vector is calculated as:

$$\begin{aligned}\mathbf{P}_{k+1}^{vv} &= \frac{1}{2N} \sum_{i=1}^{2N} (\hat{\mathbf{y}}_{k+1}^{(i)} - \hat{\mathbf{y}}_{k+1}^-)(\hat{\mathbf{y}}_{k+1}^{(i)} - \hat{\mathbf{y}}_{k+1}^-)^T + \mathbf{R}_{k+1} \\ \mathbf{P}_{k+1}^{xy} &= \frac{1}{2N} \sum_{i=1}^{2N} (\hat{\mathbf{x}}_{k+1}^{(i)} - \hat{\mathbf{x}}_{k+1}^-)(\hat{\mathbf{y}}_{k+1}^{(i)} - \hat{\mathbf{y}}_{k+1}^-)^T\end{aligned}\quad (19)$$

After this step, the measurement update in UKF takes a form similar to EKF, Equation (17). The whole process flowchart along with the corresponding equations for both nonlinear filters is given in Figure 6.



**Figure 6.** Algorithm flow of Extended Kalman filter and Unscented Kalman filter, with corresponding equations against each processing block.

#### 4.2. Position Update

In this work we have calculated the rover position using the traditional *wheel-odometry* method, *gyro-odometry* and by combining the wheel encoder data and the output of the *EASI* algorithm, called *EASI-odometry*. However, even if we provide accurate attitude estimates, the position error will grow due to errors in the dead reckoning process when using wheel encoder data to get change in position, as given in Equations (20) and (21),

$$\begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} + \mathbf{C}_b^n \begin{bmatrix} v_x^{wheel} \\ 0 \\ 0 \end{bmatrix} \Delta t \quad (20)$$

$$\mathbf{P}_k = \mathbf{P}_{k-1} + \Delta \mathbf{P}; \mathbf{P}_k \in \mathbb{R}^{3 \times 1} \quad (21)$$

where  $v_x^{wheel}$  is the forward velocity from wheel encoder and  $\Delta \mathbf{P}$  is the change in position during time  $\Delta t$ . The transformation matrix ( $\mathbf{C}_b^n$ ) can be composed of just the heading angle in the case of wheel-odometry, three propagated attitude angles in the case of gyro-odometry [27] or stabilized attitude in the case of *EASI-odometry*.

For stabilized position estimation (*i.e.*, non-drifting position), we need some external reference to correct the position errors in *PositionEKF*. The *PositionEKF* is designed to update the position only, if position measurements are available from any other external absolute sensor/system. For example, in this work, we have used GPS absolute position measurements (with 5m accuracy) after every 30 s, to simulate absolute position reference data in planetary environment. The measurement model for position updates in *PositionEKF* can be written as:

$$\mathbf{y}_k = \mathbf{h}(\mathbf{P}_k) + \boldsymbol{\eta}_k; \boldsymbol{\eta}_k \in \mathbb{R}^{3 \times 1} \quad (22)$$

Note that position measurements must be transformed to local-tangent frame (n-frame) before incorporating in estimator frame work, hence it becomes a non-linear function  $\mathbf{h}(\cdot)$  with noise vector  $\boldsymbol{\eta}_k$ .

This infrequent position measurement is chosen because on a planetary rover, absolute position measurements are not as frequently available as in terrestrial applications [3,28]. Position is propagated using wheel encoder data and attitude estimated from the *EASI* algorithm, and the position is updated if any source of absolute position becomes available, *e.g.*, from the onboard map matching algorithm (DEM), rover tracking from the orbiter circling the planet, direct rover tracking from an Earth-based station, *etc.* None of these external sources of rover position estimation can provide frequent position measurements to update the localization data in *PositionEKF* due to their limitations. However, infrequent position measurements are sufficient if we have precise and well calibrated wheel encoders and there is no wheel slippage or rover skidding during motion. The divergence in localization due to odometry errors may be reduced by incorporating global position information in a position EKF framework, as shown in Figure 7.

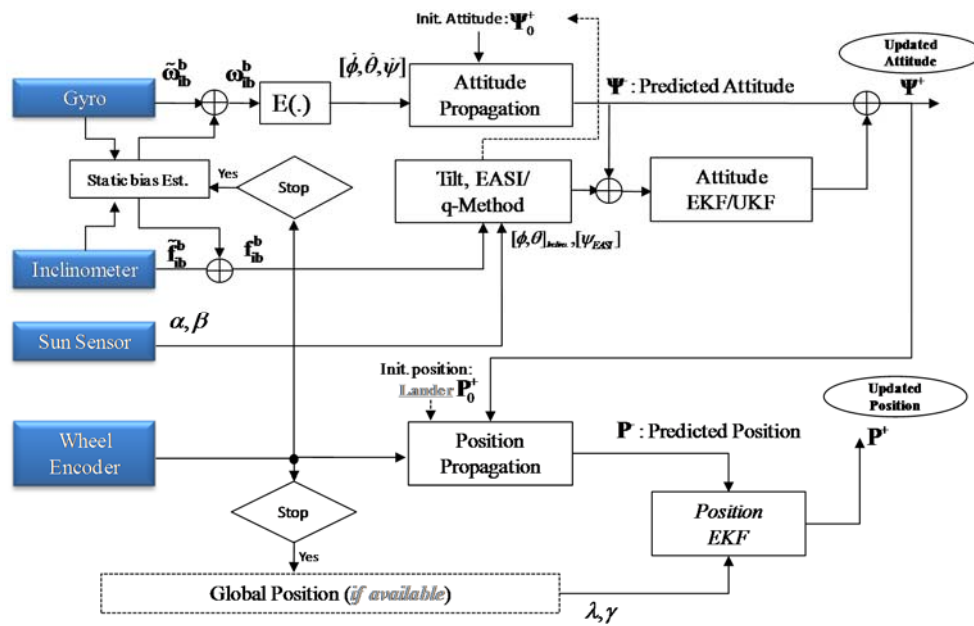


Figure 7. Sensor fusion for planetary rover localization: Block diagram.

#### 4.3. Hardware Description of Navigation Sensor Module for Micro Planetary Rover

We have developed a low cost, low weight and low power consumption navigation module called the Kitech Advanced Intelligent Rovers Sensor System (KAIROSS), as shown in Figure 8 below. This prototype unit uses two navigation sensors, a ISS-D60 (Sun sensor, Solar MEMS Technologies, Sevilla, Spain) and STIM300 (IMU, Sensoror AS, Horten, Norway), based on MEMS technology. The Sun sensor is a PSD type sensor with two orthogonal axes and a wide field of view. The IMU consists of a 3-axis accelerometer, 3-axis gyros and 3-axis inclinometer, with onboard calibration and temperature compensation routines. The main parameters of KAIROSS and the sensors used in this work are given in Table 1.

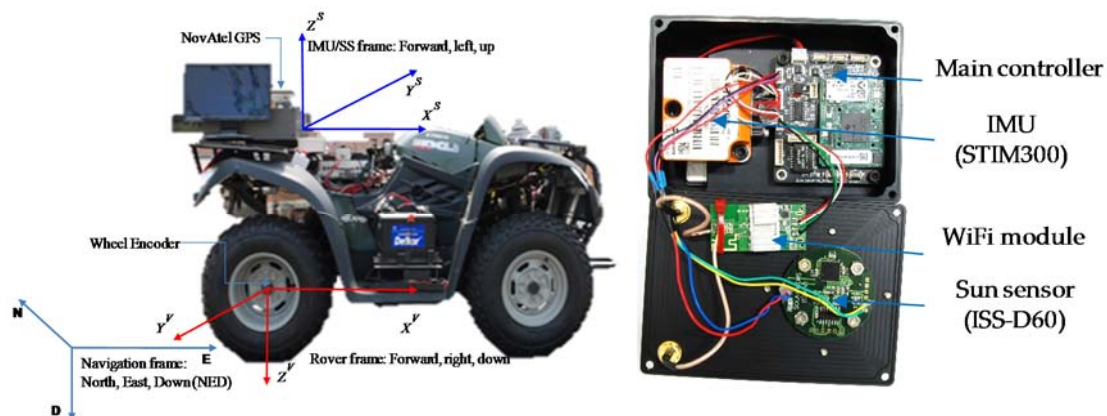


Figure 8. Left: Test rover. Right: Exploded view of the sensor module (KAIROSS) used for experiments.

**Table 1.** Sensor parameters in *KAIROSS*.

Sensors	Parameters	Wt. (g)	Pwr. (w)@5V	Size (cm)
Sun Sensor (ISS-D60)	Accuracy: $0.4^\circ$ ; 2-axis; FOV: $120^\circ \times 120^\circ$	100	0.165	$r = 4, h = 2.1$
MEMS IMU (STIM 300)	Bias stability: $0.5 (^\circ/h)$ ; ARW: $0.5 (^\circ/\sqrt{h})$	55	2	$3.9 \times 4.5 \times 2.2$
<i>KAIROSS</i>	-	270	3.2 (Load)	$11 \times 8 \times 4.3$

In *KAIROSS*, we have used the OMAP-L138 processor from Texas Instruments (TI, Dallas, TX, USA) for on-board processing, which has a dual-core SoC, a ARM926EJ-S RISC MPU, and C674x DSP, with 8MB SPI flash memory for kernel image for stand-alone system, 10/100Mbps Ethernet peripheral for data and commands transfer, 3 UARTs for getting data from the sensors, USB2.0 for WiFi, I2C bus interface, and DMA, *etc.* This system has two separated processes. Getting data from the sensor is handled by the DSP processor so that it ensures exact time stamps; and on the ARM side, it runs the Linux kernel and handles communication with the main control center. *KAIROSS* can be operated with an external/internal battery pack and weighs only 270 grams.

The test rover used for the experiments is an MXU 500i (KYMCO, Spartanburg, SC, USA), an off-road rover that can be remotely controlled, and which was modified for autonomous mode capabilities as a UGV. The navigation system is installed on top of the test rover.

## 5. Experimental Set-up and Results Discussion

To verify our proposed algorithms, the experiments are conducted on an Earth-based test rover. The differences in terms of rover navigation between the Earth-based tests and actual rover (e.g., lunar rover) is that the Earth's rotation angular velocity is about 27 times faster than that of the Moon, which results in a slower interval of Sun observations. The other major difference is the radius of the planet, which affects the positional accuracy as estimated on Earth-based test rovers. The Sun illumination and Sun appearance duration time may also be different than those on Earth [29]. We considered possible motion scenarios commonly encountered by planetary rovers. These include:

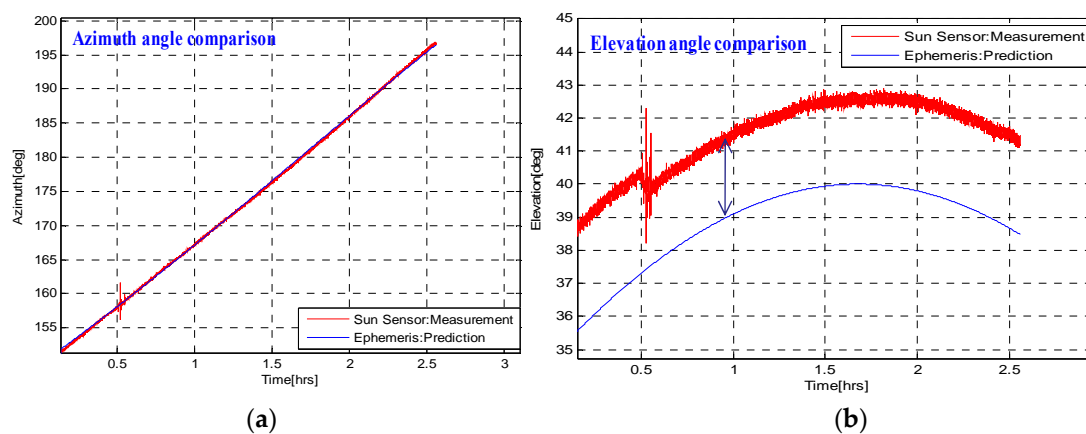
- A. Long time (e.g., hours) stationary state: Planetary rovers remain stationary most of the time unless they are commanded to move forward or perform some task.
- B. Turn in place (*TiP*) motion: This is a very common motion scenario on a planet with many unforeseen obstacles. The rover has to avoid them by performing a turn-in-place motion to reach the goal with minimum energy and time.
  - (i) *TiP* in large steps: Turn-in-place rotation test of the rover in steps of  $90^\circ$ , both clockwise (CW) and then counter-clockwise (CCW) was performed. This test was conducted to check the capabilities of the developed algorithm to track the changing absolute heading angle when the rover takes large turns to avoid obstacles on the planet surface.
  - (ii) *TiP* in small steps: This is the *TiP* test in steps of  $20^\circ$ , to check the small angle change tracking of the algorithm.
  - (iii) Continuous *TiP*: Sometimes the rover has to turn about continuously to change direction. This test verifies the algorithm's ability to track absolute angle change during continuous turning of the rover from  $0-360^\circ$ , in CW and CCW rotations.
- C. With arbitrary attitude: Most of the time rovers are commanded to move forward a few centimeters and then look around, and take a curved path, with an arbitrary attitude, around any encountered obstacle or target rock, to take an appropriate action.



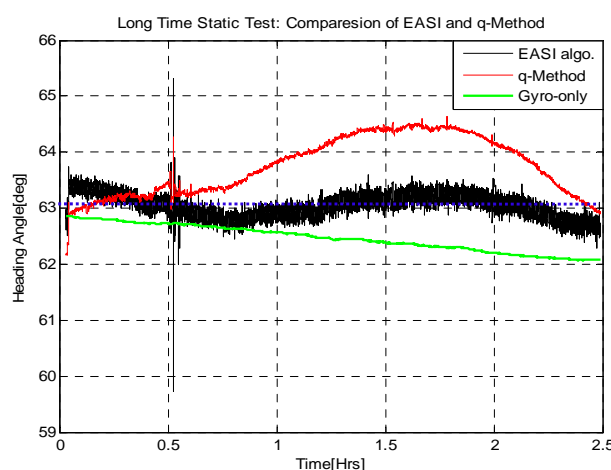
### 5.1. Long-Time Static Test

For Test (A), we kept the sensor module stationary for about 2.5 h at a fixed location (longitude:  $126.841833^\circ$  and latitude:  $37.293353^\circ$ ) in a particular direction on 27 October 2014 and captured data from *KAIROSS*. The roll and pitch angles were  $4.5^\circ$  and  $6^\circ$ , respectively during this static test. The purpose of this experiment was to test the long term stability of heading estimation algorithms, both *EASI* and *q-Method*, using onboard sensors only. The position (longitude, latitude) of the rover was taken from *Google-Earth* and time/date was the onboard system time. In actual planetary rovers, however, the absolute position may be acquired from other sources.

The results of the long time stationary condition are shown in Figures 9 and 10. In Figure 9, we compare the azimuth and elevation angles as predicted by the Sun Ephemeris data with that of the Sun sensor measurements, from 11:00 AM to 13:30 PM, a 2.5 h time duration. Both azimuth angles match as expected, but there is a fixed offset of about  $3^\circ$  in the elevation angles. This offset is due to the atmospheric effect as the Sun rays are refracted when entering the Earth's atmosphere. This effect is called "*Albedo Effect*"; however, the *Albedo Effect* is absent in atmosphere-free planets, which is our intended application.



**Figure 9.** Azimuth angle (a) and Elevation angle (b) comparison of the *EASI* algorithm and Sun Ephemeris data.



**Figure 10.** Rover heading angle estimation during the 2.5 h static test. Comparison of the *EASI* algorithm and the Optimization based *q-Method*.

The final outputs of the two heading estimation algorithms are plotted in Figure 10. The absolute heading angle determined by the *EASI* algorithm is quite smooth with a mean error of  $0.26^\circ$ , standard

deviation of  $0.21^\circ$  and a maximum error of less than  $1^\circ$ . However, the absolute heading estimate provided by the *q-Method* has an error of about  $2^\circ$ . This proves that the proposed algorithm (*EASI*) has better performance than the optimization based *q-Method* in this long term static test. In addition, the heading angle of *EASI* does not diverge, though it is noisier compared to the gyro-only solution, which inherently diverges as a result of the dead reckoning process. A quantitative comparison of the heading estimation algorithms is given in Table 2.

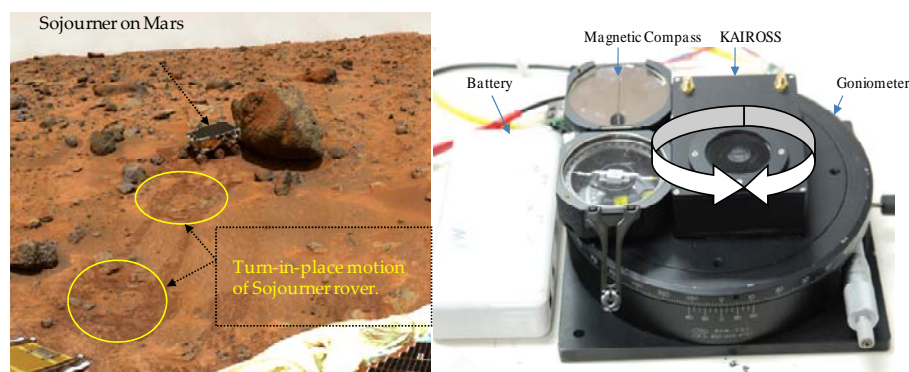
**Table 2.** Statistical comparison of heading estimation algorithms.

Algo.	Error ( $^\circ$ )	Mean Error	Std. dev.	Max. Error	RMS Error
<i>q-Method</i>		0.76	0.53	1.8	0.93
<i>EASI</i>		0.26	0.21	0.7	0.34
Gyro-only		0.84	0.24	0.5	1.16

## 5.2. Turn-In-Place (TiP) Tests

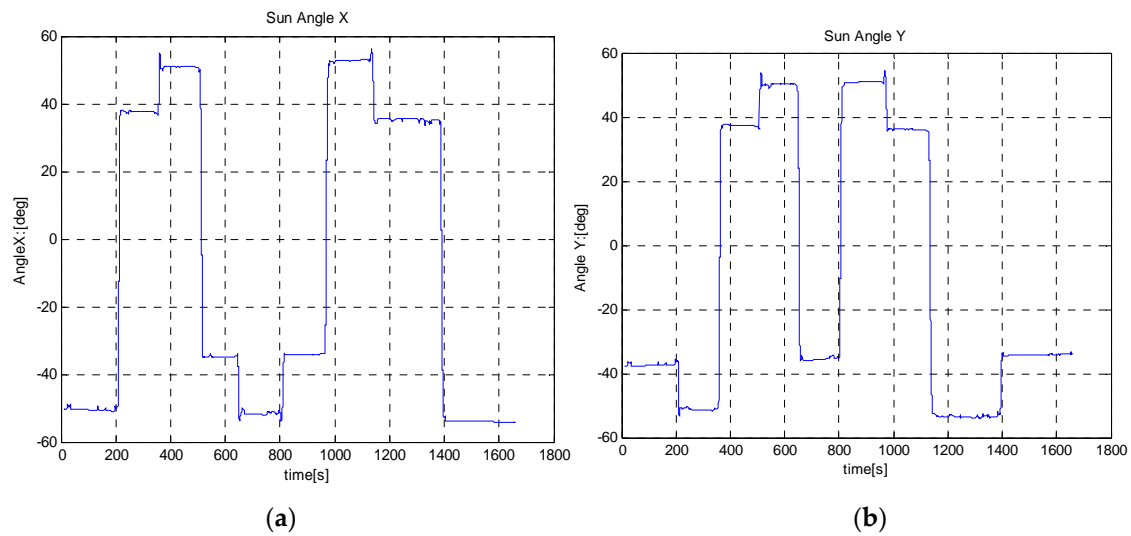
The second motion scenario, *Turn-In-Place (TiP)* is a very important type of motion from a planetary rover's perspective, as the rover has to take sharp turns to avoid unforeseen obstacles.

To test the *TiP* motion scenario (B), a sensor set up composed of *KAIROSS*, a Goniometer and a high precision magnetic compass was used, as shown in Figure 11.

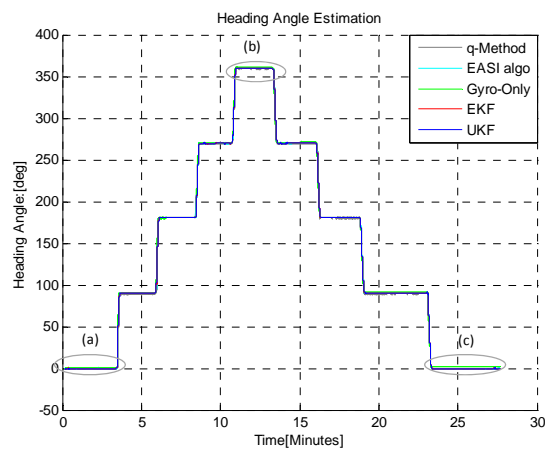


**Figure 11.** Turn-in-place motion: (Left) Actual Mars rover (Sojourner) took two *TiPs* before reaching the targeted rock [30]; (Right) Sensors set-up for *TiP* test (B), both CCW and CW rotation through  $0\text{--}360^\circ$ .

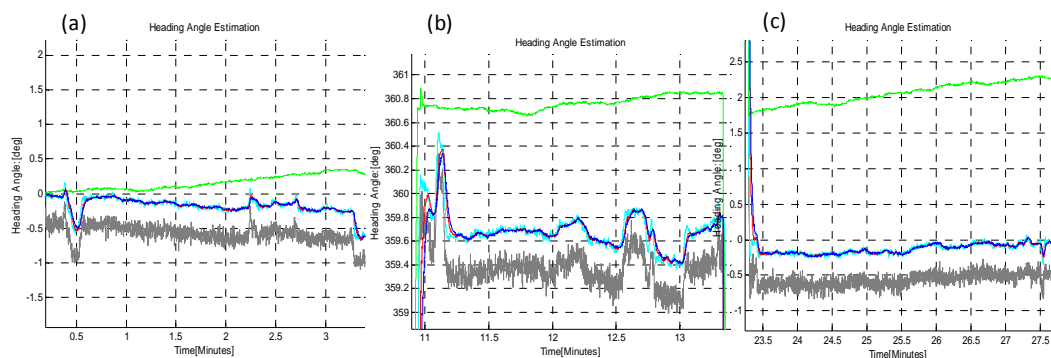
In the first test (B-i), the sensor module was rotated through large steps of  $90^\circ$ , in CW and then CCW. For each step, the rover rested for about 2 min before performing the next rotation and completed a full  $0\text{--}360^\circ$  rotation and back. The output of the absolute heading estimation algorithms were compared against the reference value taken from the Goniometer readings, and the absolute heading angle measured by a very precise magnetic compass. The results of this test are shown in Figures 12–14 and in Table 3.



**Figure 12.** Sun sensor output during Test (B-i) with  $90^\circ$  rotation steps in the CW and CCW directions. (a) Angle-x is the angle between z-axis and projection of Sun ray on x-z plan; (b) Angle-y is the angle between z-axis and projection of Sun ray on y-z plan.



**Figure 13.** *TiP* with large steps ( $90^\circ$ ) in CW and CCW. Comparison of heading estimation algorithms.

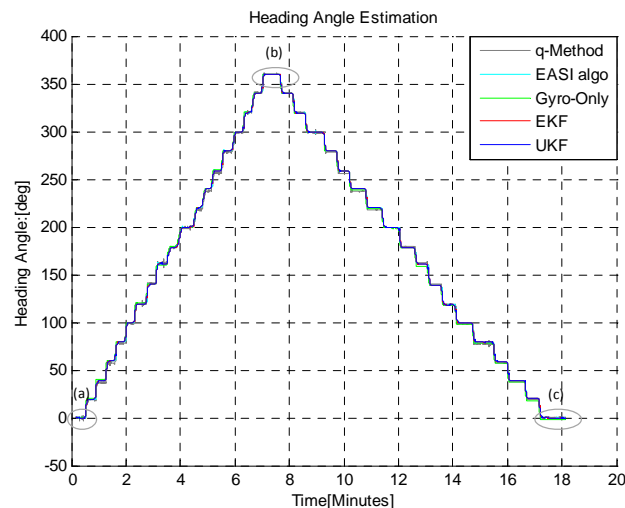


**Figure 14.** Zoom-in view of heading angle, (a) at start; (b) in middle and; (c) at the end of test.

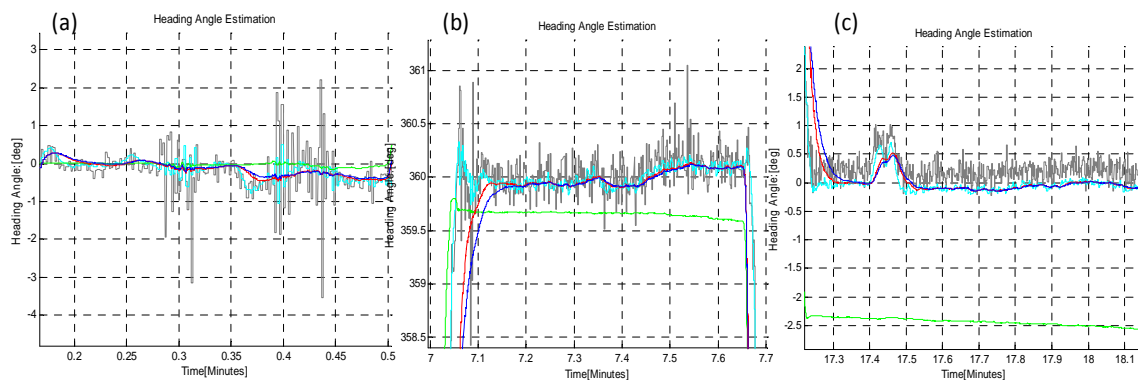
**Table 3.** Statistical comparison of heading estimation algorithms in the 1st *TiP* test (B-i).

Algo.	Error (°)	Start (0°)		Mid (360°)		End (0°)		
		Mean	Std. dev.	Mean	Std. dev.	Error	Mean	Std. dev.
<i>EASI</i>		−0.158	0.095	359.65	0.112	0.348	−0.134	0.067
<i>q-Method</i>		−0.163	0.117	359.74	0.125	0.256	−0.165	0.090
EKF		−0.158	0.090	359.65	0.105	0.349	−0.134	0.063
UKF		−0.154	0.085	359.65	0.102	0.348	−0.113	0.052
Gyro-only		0.152	0.103	360.75	0.055	0.759	2.062	0.145

In the second *TiP* test (B-ii), the sensor plate was rotated in short steps of  $20^\circ$ , to ensure that the developed algorithms can also reliably track small changes in absolute heading angle, while performing the turn-in-place rotation, as shown in Figures 15 and 16. The zoom-in view (a,b,c) and statistical comparison are shown in Figure 16 and in Table 4. The *q-Method* has a large mean error and standard deviation as compared to the *EASI* algorithm. The *q-Method* does not track heading angle well in this quasi-dynamic test due to the dynamic motion, because it needs multiple measurements from the same position to perform an optimization operation.



**Figure 15.** *TiP* with small steps ( $20^\circ$ ) in CW and CCW rotation. The EKF and UKF estimates end up with close to the initial heading angle but the gyro-only solution shows clear divergence, more than  $2^\circ$  at the end of this experiment.

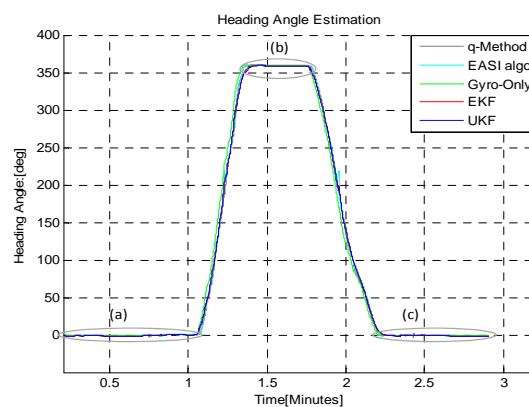


**Figure 16.** Zoom-in view in *TiP* test (B-ii), with small increments ( $20^\circ$ ), with both CW and CCW rotation through  $0\sim 360^\circ$ , at the start (a); mid (b) and end (c) point of rotation. Clearly, the *q-Method* has large errors as compared to *EASI*.

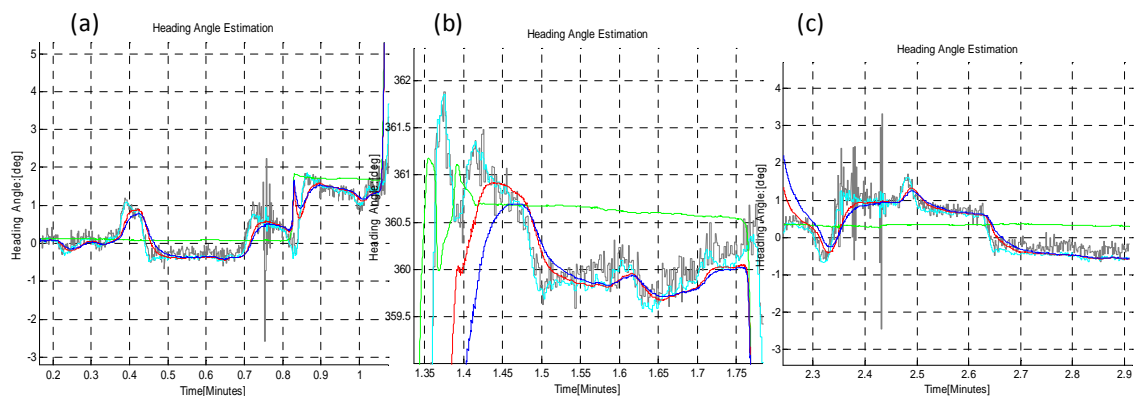
**Table 4.** Statistical comparison of heading estimation algorithms in *TiP* test (B-ii).

Algo.	Error (°)	Start (0°)		Mid (360°)		End (0°)	
		Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
<i>EASI</i>		0.241	0.265	360.410	0.093	0.414	0.063
<i>q-Method</i>		0.605	0.625	360.860	0.180	0.995	0.148
EKF		0.255	0.192	360.400	0.077	0.333	0.049
UKF		0.179	0.278	360.400	0.073	0.231	0.038
Gyro-only		0.324	0.334	360.040	0.022	-2.073	0.040

In the third Turn-in-place test (B-iii), the sensor plate was rotated continuously in order to check the continuous tracking of absolute angle by both algorithms, as shown in Figure 17, Figure 18 and Table 5. This scenario of motion may sometimes be required to turn the rover back because of some large encountered obstacle, for example. The statistical comparison at the start (a), mid (b) and at the end (c) of this test shows that *EASI* algorithm has better performance against the *q-Method*, and a little better performance in UKF as compared to EKF is also observed in this application.



**Figure 17.** Continuous rotation (0~360°) in CW and CCW direction in the *TiP* test (B-iii). The *EASI* algorithm output gives a noisy but stable heading angle. EKF and UKF show similar performance in this short time continuous rotation experiment. The *q-Method* does not perform well, as compared to the *EASI*, due to sensor's continuous motion.



**Figure 18.** Zoom-in view of continuous Turn-in-place test (B-iii): Start (a); mid (b) and at the end (c) of rotation in CW and CW through 0~360° rotation. The *q-Method* heading estimate is noisy and has large errors due to the dynamic condition of the sensor.

**Table 5.** Statistical comparison of heading estimation algorithms in the continuous *Tip* test.

Algo.	Error (°)	Start (0°)		Mid (360°)		End (0°)	
		Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
<i>EASI</i>		0.332	0.676	360.111	0.411	0.191	0.657
<i>q-Method</i>		0.651	0.694	360.297	0.414	0.427	0.685
EKF		0.326	0.659	360.101	0.396	0.285	0.624
UKF		0.301	0.651	360.026	0.307	0.234	0.634
Gyro-only		0.495	0.698	360.632	0.055	0.532	0.022

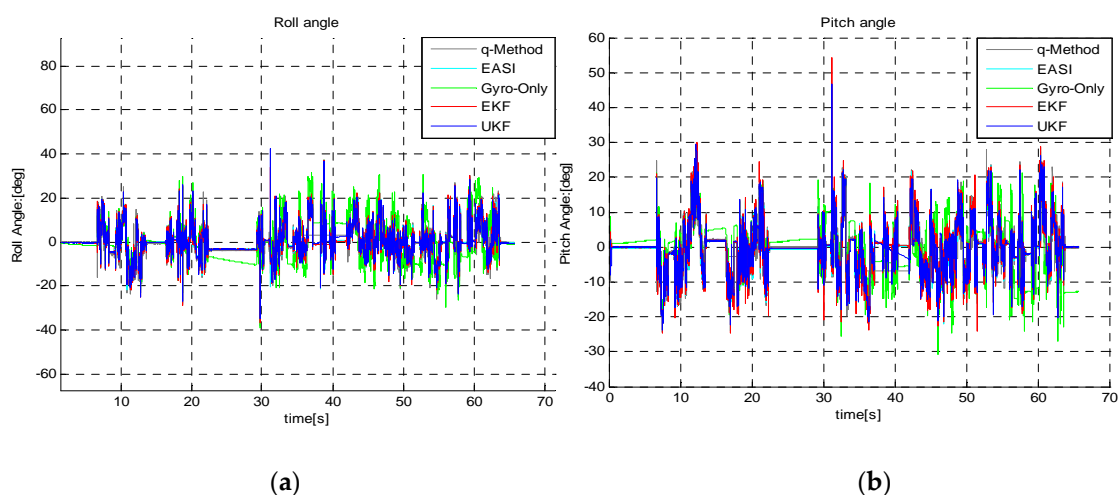
### 5.3. Long Time Dynamic Test

To test the algorithm's performance under the dynamic condition of the rover (Test-C), the sensor module was fixed on a hand-driven cart and moved slowly (e.g., 2 cm/s.) in a stop-and-go fashion and with arbitrary roll, pitch and heading. At the start of the dynamic test, the sensor module was kept static for some time to capture the turn-on bias of the IMU. During motion, the Sun obstruction condition was also tested by blocking the Sun radiation for some time.

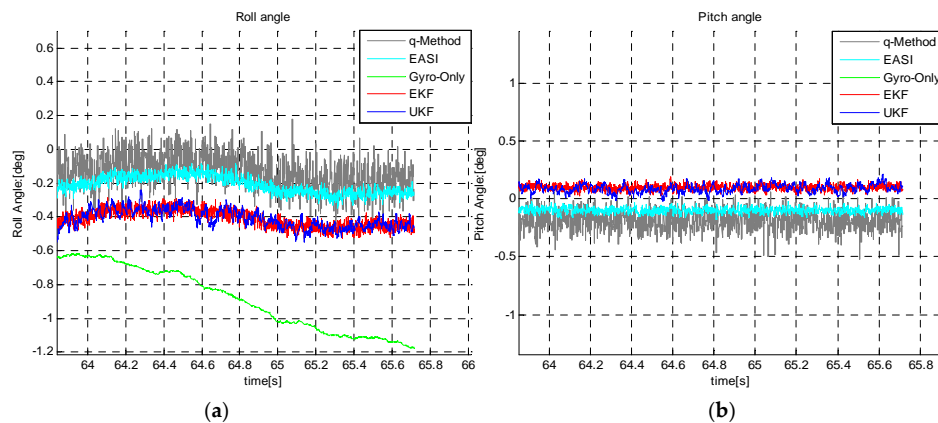
In this test, the sensor module was placed at a certain initial orientation, and it was arranged that the module would end up in same orientation at the end of this long dynamic test. Hence, the initial and final orientation angles were taken as reference values for error calculations. We drove our sensor module with different speeds and with an arbitrary attitude for about an hour. The motion was in a stop-and-go fashion in order to mimic the planetary rover's motion. The integrated (or fused) results are expected to show superior performance in terms of accuracy and robustness in comparison to a single navigation sensor.

The EKF and UKF were compared for accuracy and robustness by simulating a scenario in which the Sun radiations were obstructed for some reason, such as the rover being under the shadow of a big rock, or when the Sun sensor is occluded by the rover's own shadow. In that case the Sun's radiation was below the critical value of 300 W/m<sup>2</sup>. In Sun obstruction situations, the absolute heading information from *EASI* or *q-Method* is not available to update the EKF/UKF.

Roll and pitch angles propagated by the gyro-only method show clear divergence (Figure 19), whereas the EKF/UKF has almost the same performance, as is evident from the end-trajectory zoomed-in views (Figure 20). The numerical comparison of roll/pitch is given in Table 6, below.

**Figure 19.** (a) Roll angle and (b) pitch angle estimation during the long dynamic test.





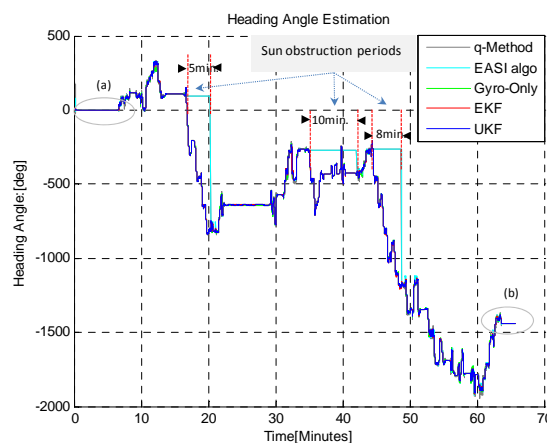
**Figure 20.** End trajectory Roll angle (a) and pitch angle (b) estimation during long time dynamic test.

**Table 6.** Statistical comparison of estimated tilt angles in the long-time dynamic test.

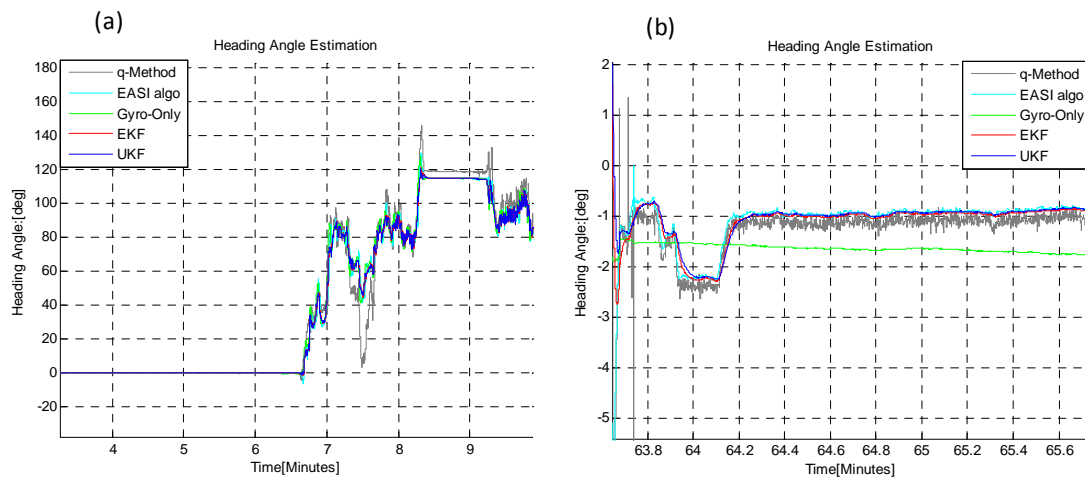
Algo.	Error (°)	Roll				Pitch			
		Stat		End		Start		End	
		Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
EASI	−0.042	0.026	−0.111	0.055	−1.192	0.021	−1.097	0.021	
<i>q-Method</i>	−0.061	0.088	−0.238	0.105	−1.602	0.082	−1.689	0.089	
EKF	−0.268	0.029	−0.413	0.051	−0.982	0.025	−0.898	0.022	
UKF	0.269	−0.036	−0.413	0.055	−0.980	0.038	−0.909	0.037	
Gyro-only	−0.542	0.057	−0.937	0.163	0.050	0.041	−13.875	0.066	

The estimated heading angle provided by different algorithms is given in Figure 21. There were Sun obstruction periods of different time durations during this long-time dynamic test, in order to check the capability of the estimators, EKF/UKF during Sun obstruction periods.

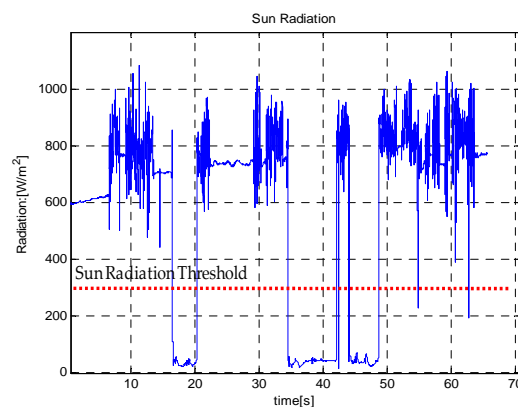
Three Sun-obstruction periods of different time lengths (5, 10 and 8 min) were considered during the motion, to check the robustness of the heading estimation output of the EKF/UKF algorithms. It was noted that during Sun obstruction periods, neither the *EASI* nor the *q-Method* work because of the Sun sensor limitations. However, the fused data of IMU and *EASI* (i.e., the EKF/UKF) provided heading angle tracking even during the Sun obstruction periods. The zoom-in view of the start and end heading angle is given in Figure 22. The Sun radiation data with threshold for this experiment is shown in Figure 23. The quantitative comparison of heading estimation algorithms is given in Table 7.



**Figure 21.** Heading angle (unwrapped) estimated by different algorithms during the one hour dynamic test.



**Figure 22.** Zoom-in view of Start (a) and End (b) heading angle estimation in the dynamic test of 1 h. The *q-Method* is not in agreement with the other algorithms in the dynamic condition. However, in the static condition (e.g., before 6.5 min and after 63.8 min), the *q-Method* also performs nearly as well, as expected.



**Figure 23.** Sun radiation during the long-time dynamic test. Note that below a threshold of  $300 \text{ W/m}^2$ , the sensor output is not reliable for absolute heading calculation.

**Table 7.** Statistical comparison of heading estimation algorithms in the dynamic test (only end-trajectory errors are compared).

Algo.	Error (°)	Mean	Std. Dev.	Max. Error
EASI		−0.912	0.090	1.091
<i>q-Method</i>		−1.186	0.096	1.7105
EKF		−0.954	0.044	1.067
UKF		−0.928	0.041	1.017
Gyro-only		−1.664	0.042	1.755

The comparison of *EASI* and *q-Method* shows that in the static condition, both absolute heading estimation algorithms work well as expected, but in the dynamic case, the *q-Method* fails to estimate heading reliably, whereas the *EASI* has good performance both in the static and in the dynamic situation too. The reason is that *EASI* is a *one-shot* algorithm whereas the *q-Method* requires the collection of a number of measurements before applying some kind of optimization method.

In Figures 24 and 25 we have summarized the statistical results of all five tests conducted in this study, providing the mean error and standard deviation of all the algorithms compared in each test.

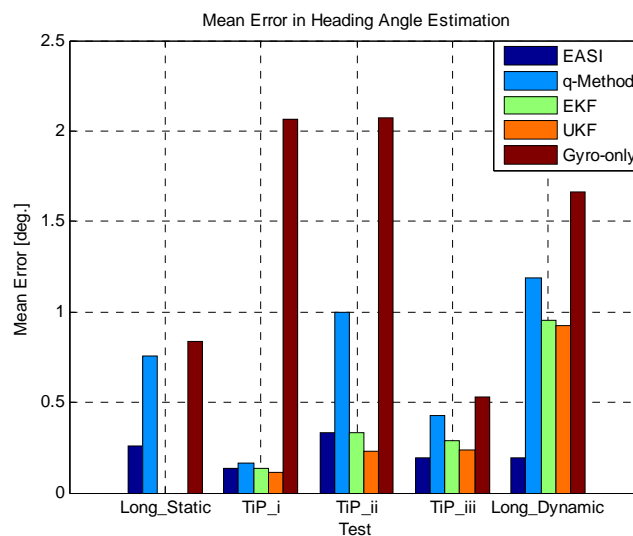


Figure 24. Mean error in all five tests.

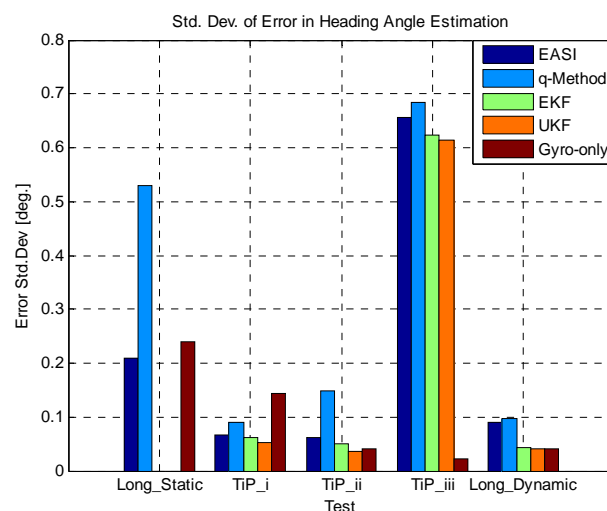
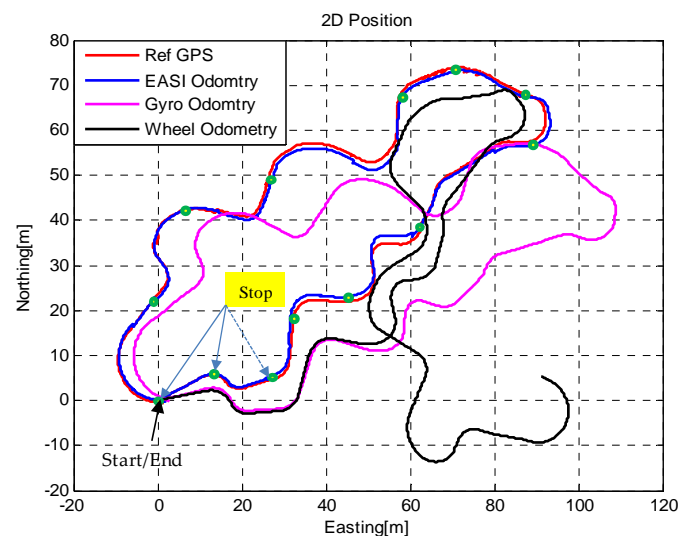


Figure 25. Standard deviation of error in all five tests.

Both the qualitative and quantitative results confirm that the developed algorithm, *EASI* outperforms the conventional optimization-based *q-Method*, especially in the dynamic condition. It was also demonstrated in this work that multisensory fusion has higher accuracy and robustness than the individual sensor/methods, in estimating attitude for the planetary test rover.

#### 5.4. Planetary Rover's Localization

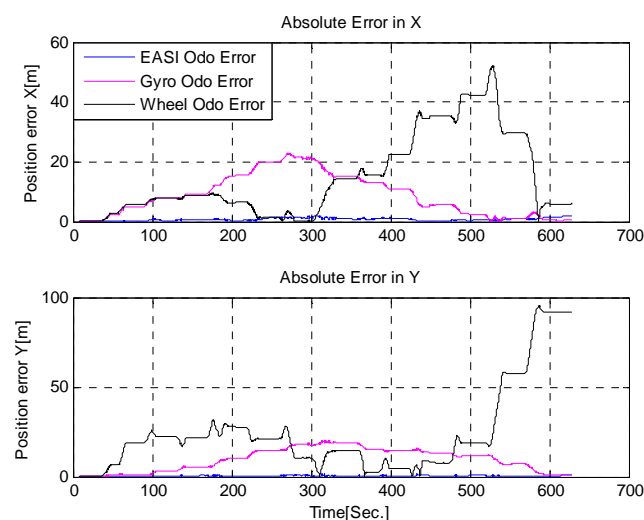
For completeness, we also developed an integrated position estimation algorithm for the planetary rover, shown in Figure 26. For comparison, three localization methods were considered: conventional *wheel-odometry* as for ground robots, *gyro-odometry* and *EASI-odometry*. In *wheel-odometry*, the forward position change is calculated for every motion from pre-calibrated encoder data, and heading angle is calculated from steering wheel encoder data. However, this kind of motion estimation is only valid over a limited period of time and on flat surfaces. Depending on surface roughness, this method is not reliable for long time operations.



**Figure 26.** 2D localization plot: Comparison of position estimations using different localization techniques for the planetary rover.

However, the performance of *wheel-odometry* can be enhanced by including the gyro data in the attitude calculation. Instead of using only heading angle from the steering wheel encoder, in *gyro-odometry*, 3D attitude angles are used to propagate wheel position data. This type of position estimation has good performance as compared to wheel encoders only, but still depends on the accuracy of the gyros used. During long time operations, *gyro-odometry* also suffers from drift and position estimates become no longer usable, and cross-track errors especially have increased error in estimating heading angle.

To reduce errors in *gyro-odometry*, we must compensate attitude errors by some absolute means, e.g., from the inclinometer and Sun sensor data used in this work. By combining the Sun sensor and IMU, called *EASI-odometry*, the errors in localization of the rover are also greatly reduced. The localization estimates were compared with the high-performance NovAtel GPS module with 5 m accuracy. The absolute error in position estimates are shown in Figure 27 below. The quantitative comparison of the three position estimation methods is given in Table 8.



**Figure 27.** Absolute error in position estimation.

The quantitative comparison of position estimates shows that the position errors of the *EASI-odometry* were less than 3m when compared with a NovAtel GPS module in this 10 min and about 200 m run.

**Table 8.** Statistical comparison of different localization methods.

Algo.	Error (°)	Abs. Error (max)		2D Abs. Error	RMS Error		2D RMS Error
		x	y	$\sqrt{x^2 + y^2}$	x	y	$\sqrt{x^2 + y^2}$
Wheel Odo.		52.1	95.3	108.6	20.2	34.3	39.8
Gyro Odo.		22.8	20.5	30.7	11.3	11.6	16.2
<i>EASI Odo.</i>		1.8	1.6	2.4	0.9	0.6	1.1

## 6. Conclusions

In this work, we have presented methods for estimating attitude and position using different algorithms which only employ on-board low cost, low power MEMS sensors. Two absolute heading estimation algorithms were developed and compared for accuracy and reliability. The *EASI* algorithm outperformed the traditionally used optimization-based *q-Method* in rover's in-motion condition. However, in static tests, both algorithms exhibited comparable performance. The drawback of the absolute heading angle determination algorithms is that their output is noisier and the Sun sensor does not produce any measurements below a certain Sun light radiation (<300 W/m<sup>2</sup>). At the same time, the gyro-only attitude determination has an inherent limitation during long time operations, as the attitude angles start drifting after some time.

To cope with this issue, the output of the absolute attitude estimation algorithm was fused with the relative attitude sensor, *i.e.*, MEMS gyros, to reduce error drift in attitude with the passage of time. The fused attitude solution provides a reliable heading angle in the event of Sun radiation blockage and has better accuracy as compared to absolute heading algorithms alone. Furthermore, the fused attitude is non-drifting and less noisy as compared to the gyro-only or *EASI/q-Methods*.

For the sensor fusion in this nonlinear problem, we compared two commonly known filters, the Extended Kalman filter and Unscented Kalman filter. The UKF has a little better performance than EKF in this application; however, both filters have error <1°, in heading angle estimation. For completeness, we also compared the localization results of three algorithms, and it was shown that when the localization algorithm was integrated with the *EASI* algorithm, the errors in the position were also reduced to less than 3 m in a 200 m run, which is about 1.5% of the total distance travelled. As a future research work, the absolute heading estimation algorithm will be integrated with other localization systems, *e.g.*, visual odometry, DEM matching, *etc.*

**Author Contributions:** Muhammad Ilyas conceived the main idea and wrote this paper. Beomjin Hong and Seung-Ho Baeg designed the hardware and data acquisition module for experiments conducted in this work. Paper revision and help in data analysis and visualization was conducted by Kuk Cho. All of the work is done under the supervision of Sangdeok Park and Seung-Ho Baeg.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Jet Propulsion Laboratory. Available online: [http://mars.jpl.nasa.gov/mer/mission/spacecraft\\_rover\\_wheels.html](http://mars.jpl.nasa.gov/mer/mission/spacecraft_rover_wheels.html) (accessed on 19 May 2016).
2. Maimone, M.; Biesiadeck, J.; Edward, T.; Yang, C.; Chris, L. Surface navigation and mobility intelligence on the Mars Exploration Rovers. In *Intelligence for Space Robotics*; Howard, A.M., Tunstel, E.W., Eds.; TSI Press: San Antonio, TX, USA, 2006; Chapter 3; pp. 45–69.
3. Bekey, G.A. *Autonomous Robots: From Biological Inspiration to Implementation and Control*; The MIT Press: Cambridge, MA, USA, 2005.

4. Mars Global Surveyor. Available online: <http://mgs-mager.gsfc.nasa.gov/Kids/magfield.html> (accessed on 9 March 2016).
5. Titterton, D.H.; Weston, J.L. *Strapdown Inertial Navigation Technology*; The Institution of Engineering and Technology: London, UK, 2004.
6. Matthies, L.; Maimone, M.; Johnson, A.; Yang, C.; Willson, R.; Villalpando, C.; Goldberg, S.; Huertas, A. Computer vision on Mars. *Int. J. Comput. Vis.* **2007**, *75*, 67–92. [[CrossRef](#)]
7. Maimone, M.; Biesiadecki, J.; Tunstel, E.; Yang, C. *Intelligence for Space Robotics*; Howard, A.M., Tunstel, E.W., Eds.; TSI Press: San Antonio, TX, USA, 2006.
8. Doraiswami, R.; Price, S. A Robust Position Estimation Scheme Using Sun Sensor. *IEEE Trans. Instrum. Meas.* **1998**, *47*, 595–603. [[CrossRef](#)]
9. Volpe, R. Mars rover navigation results using sun sensor heading determination. In Proceedings of the IEEE/RSJ International conference on Intelligent Robot and Systems, Kyongju, Korea, 17–21 October 1999.
10. Trebi-Ollennm, A.; Huntsberge, T.; Yang, C.; Baumgartner, E.T. Design and analysis of a sun sensor for planetary rover absolute heading detection. *IEE Trans. Robot. Autom.* **2001**, *17*, 939–947. [[CrossRef](#)]
11. Pingyuan, C.; Fuzhan, Y.; Hutao, C. Attitude and position determination scheme of lunar rovers basing on the celestial vectors observation. In Proceedings of the IEEE International Conference on Integration Technology, Shenzhen, China, 20–24 March 2007; pp. 538–543.
12. Davenport, P. *A Vector Approach to the Algebra of Rotations with Applications*; National Aeronautics and Space Administration: Washington, DC, USA, 1968.
13. Furgale, P.; Enright, J.; Barfoot, T. Sun sensor navigation for planetary rovers: Theory and field testing. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 1631–1647. [[CrossRef](#)]
14. Lambert, A.; Furgale, P.; Barfoot, T.; Enright, J. Field testing of visual odometry aided by a sun sensor and inclinometer. *J. Field Robot.* **2012**, *29*, 426–444. [[CrossRef](#)]
15. Enright, J.; Furgale, P.; Barfoot, T. Sun sensing for planetary rover navigation. In Proceedings of the IEEE Aerospace Conference, Big Sky, MY, USA, 7–14 March 2009; pp. 1–12.
16. Linda, M.M. MEMS for space applications. *Proc. SPIE Design Test Microfabr. MEMS MOEMS* **1999**, *3680*. [[CrossRef](#)]
17. Sebastien, C.; Martella, P.; Durrant, D.; Bertsch, N.; Dussy, S. Development of the European IMU for Space Applications. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, IL, USA, 10–13 August 2009.
18. Khaled, S.A.; Vanelli, C.A.; Biesiadecki, J.J.; Maimone, M.W.; Cheng, Y.; San Martin, A.M.; Alexander, J.W. Attitude and position estimation on the Mars exploration rovers. In Proceedings of the 2005 IEEE International Conference on Systems, Man, and Cybernetics—SMC, Waikoloa, HI, USA, 10–12 October 2005; pp. 20–27.
19. Sukkarieh, S. Low Cost, High Integrity, Aided Integrated Navigation Systems for Autonomous Land Vehicles. Ph.D. Thesis, The University of Sydney, Sydney, Australia, 2000.
20. Ilyas, M.; Cho, K.; Baeg, S.-H.; Sangdeok, P. Absolute navigation information estimation for micro planetary rovers. *Int. J. Adv. Robot. Syst.* **2016**, *13*. [[CrossRef](#)]
21. Groves, P.D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*; Artech House: Boston, MA, USA; London, UK, 2008.
22. Kim, J. Autonomous Navigation for Airborne Application. Ph.D. Thesis, The University of Sydney, Sydney, Australia, 2004.
23. Reda, I.; Andreas, A. *Solar Position Algorithm for Solar Radiation Application*; National Renewable Energy Laboratory: Golden, CO, USA, 2008.
24. SolarMems. ISS-DX Technical Specifications. Available online: <http://www.solar-mems.com/en/> (accessed on 19 May 2016).
25. Wahba, G. A least square estimate of spacecraft attitude. *SIAM Rev.* **1965**, *7*, 409. [[CrossRef](#)]
26. Simon, D. *Optimal State Estimation*; Wiley-Interscience: New York, NY, USA, 2006.
27. Reina, G. Cross-Coupled Control for All-Terrain Rovers. *Sensors* **2013**, *13*, 785–800. [[CrossRef](#)] [[PubMed](#)]
28. Rossmann, J.; Schlette, C.; Emde, M.; Sondermann, B. Discussion of a self-localization and navigation unit for mobile robots in extraterrestrial environments. In Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Sapporo, Japan, 29 August–1 September 2010; pp. 46–53.



29. Penga, Y.; Xie, L.; Liu, J. Simultaneous celestial positioning and orientation for the lunar rover. *Aerosp. Sci. Technol.* **2014**, *34*, 45–54.
30. Mars Pathfinder. Available online: [http://www.nasa.gov/mission\\_pages/mars-pathfinder/](http://www.nasa.gov/mission_pages/mars-pathfinder/) (accessed on 19 May 2016).



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).