*Sensors* **2014**, *14*, 8167-8188; doi:10.3390/s140508167

OPEN ACCESS

*sensors*

**ISSN 1424-8220**

www.mdpi.com/journal/sensors

*Article*

# Sensor Saturation Compensated Smoothing Algorithm for Inertial Sensor Based Motion Tracking

**Quoc Khanh Dang and Young Soo Suh ***

Department of Electrical Engineering, University of Ulsan, Mugeo-dong, Namgu, Ulsan 680-749, Korea; E-Mail: khanhdq86@yahoo.com

***** Author to whom correspondence should be addressed; E-Mail: yssuh@ulsan.ac.kr; Tel.: +82-52-259-2196.

*Received: 11 March 2014; in revised form: 29 April 2014 / Accepted: 29 April 2014 / Published: 6 May 2014*

**Abstract:** In this paper, a smoothing algorithm for compensating inertial sensor saturation is proposed. The sensor saturation happens when a sensor measures a value that is larger than its dynamic range. This can lead to a considerable accumulated error. To compensate the lost information in saturated sensor data, we propose a smoothing algorithm in which the saturation compensation is formulated as an optimization problem. Based on a standard smoothing algorithm with zero velocity intervals, two saturation estimation methods were proposed. Simulation and experiments prove that the proposed methods are effective in compensating the sensor saturation.

**Keywords:** sensor saturation compensation; smoothing algorithm

## 1. Introduction

In motion tracking, there are many ways to estimate the trajectory of a moving object. Moving objects can be tracked accurately by using visual devices such as camera systems [1,2]. In [1], with 40 landmarks attached on the body, Hong *et al.* used the Eagle Digital motion capture system with seven cameras to analyze 14 angles and one ratio of gait features. In [2], Lee and Grimson investigated person identification and gender classification based on moments computed from the silhouette of walking people. However, these camera systems are limited in their setup ranges and sometimes have high implementation costs. Moreover, the angle views of the cameras are also limited and they are easily affected by illumination. Due to these reasons, for long distance or outdoor measurements, motion tracking based on camera systems seems to be a difficult task.

To avoid the mentioned disadvantages, inertial measurement units (IMU) can be used instead as wearable devices. IMUs are widely used due to their small size and low cost. With the development of the technology, IMUs are now becoming more accurate. In [3], Tadano *et al.* proposed a method using quaternion calculations from seven sensor units consisting of a tri-axial acceleration and gyro sensors. The quaternions, which are computed from the sensors attached on limbs and waist, are used in a gait wire frame model to generate the gait animation. To increase the accuracy, the IMUs are used with other aiding devices such as cameras in [4,5] or force sensors in [6].

However, IMUs still have their own limitations such as susceptibility to noise and limited dynamic range. The accuracy of inertial sensor-based estimation can be improved by using zero velocity updates as in [7], or taking advantage of the relative position and attitude of multiple sensors [8,9]. In [7], a robot arm control with an automatic calibration function based on inertial sensors is proposed. The authors state that the drift of the sensors is clearly removed by applying zero velocity updates. In [8], Helten *et al.* introduce another way which takes into account the relative position and attitude of multiple sensors to improve the accuracy of human motion estimation. Using the same method, Tao *et al.* [9] estimate limb movement. They also use the limb biomechanical model characteristics to provide constraints for sensors' relations. However, the other drawback of the inertial sensor, the saturation, has not been considered. Saturation is a state in which the signal that needs to be measured is larger than the dynamic range of the sensor. When that happens, the output of the sensor becomes the limiting value of the sensor range. This induces a considerable error between the true and estimated values during motion tracking. In this paper, we propose two methods for estimating the sensor saturation. Assuming that the motion is between not moving intervals (called zero velocity intervals), we formulate a saturation estimation problem as an optimization problem by modifying a smoother used in [10,11]. In the proposed methods, some state constraints mentioned in [12] could be added to increase the accuracy of the algorithm.

The paper is organized in five main sections and a conclusion. Section 2 points out the problem formulation. In Section 3, a standard smoothing algorithm with zero velocity intervals is described in detail. Sections 4 and 5 propose some methods for sensor saturation estimation. Some experiments to verify the proposed methods are given in Section 6. The last section concludes the paper.

## 2. Problem Formulation

We consider a moving object case where an IMU is attached on the object. There are two coordinate systems in this paper: the navigation coordinate frame and the body coordinate frame. The $z$ axis of the navigation coordinate frame coincides with the local vertical. The choice of $x$ axis is arbitrary. The body coordinate frame is defined as a frame with three axes coinciding with the three axes of the inertial measurement unit. The subscripts $b$ (body) and $n$ (navigation) are used to emphasize that a vector or matrix belongs to the body or navigation coordinate frame, respectively.

Our goal is to estimate attitude (expressed using the quaternion), position and velocity of the object from the sensor data. Let $q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix} \in R^4$ be a quaternion representing the rotation relationship between the navigation coordinate frame and the body coordinate frame. Let $C(q) \in R^{3\times3}$

be the rotation matrix corresponding to the quaternion $q$ [13], and $r \in R^3$ and $v \in R^3$ be the position and velocity of the object, respectively. We have the following basic equations [14]:

$$\begin{aligned} \dot{q} &= \frac{1}{2}\Omega(\omega_b)q \\ \dot{v}_n &= a_n = C^T(q)a_b \\ \dot{r}_n &= v_n \end{aligned} \tag{1}$$

where $a_n \in R^3$ and $a_b \in R^3$ are the acceleration made by forces other than gravitational field in the navigation coordinate frame and body coordinate frame, respectively. The symbol $\Omega$ is defined by:

$$\Omega(\omega) \triangleq \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$$

where $\omega_b = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ is the body angular rate.

The inertial measurement unit used in this paper consists of three axis gyroscopes and accelerometers. Let $y_g \in R^3$ be the gyroscope output and $y_a \in R^3$ be the accelerometer output. They satisfy the following relationship [15]:

$$\begin{aligned} y_g &= \omega_b + n_g \\ y_a &= a_b + C(q)\bar{g} + n_a \end{aligned} \tag{2}$$

where $n_g$ and $n_a$ are zero mean white Gaussian sensor noises with covariances $R_g = E\{n_g n_g^T\}$ and $R_a = E\{n_a n_a^T\}$, and $\bar{g} = [0 \quad 0 \quad g]^T$ The symbol $g$ denotes the gravitational acceleration. It is assumed the sensor bias is already compensated using the standard calibration algorithm [16].

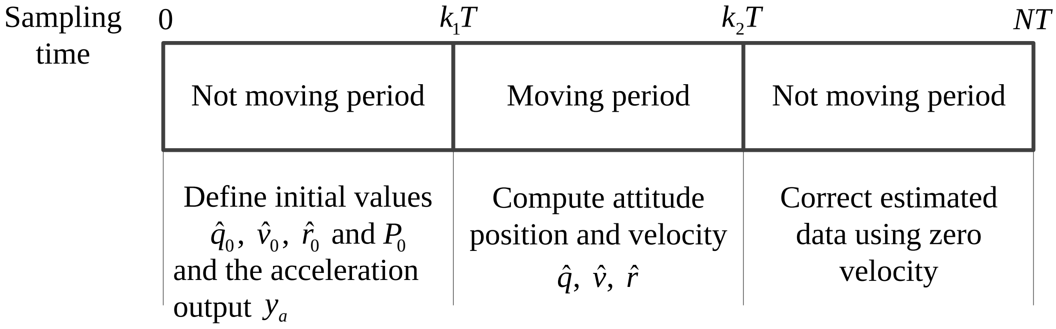
In summary, our goal is to estimate the quaternion, velocity and position of a moving object using the accelerometer and gyroscope data where there could be sensor saturation. Since a smoother is used instead of a filter, we note that the proposed method is offline analysis of attitude and position.

## 3. Standard Smoothing Algorithm with Zero Velocity Intervals

In this section, a standard smoothing algorithm is formulated in the quadratic optimization problem. A general method of formulating the smoothing problem in the optimization problem is given in [10]. In this section, we apply the result in [10] to an attitude and position estimation problem, where there are zero velocity intervals. Sensor saturation compensation is not considered in this section and will be discussed in Section 4.

We assume that the motion is a short movement consisting of a moving interval and two not moving intervals (see Figure 1). This type of movement can be found in walking [4,5], golf swing [17] and so on. The sensor data are sampled with the sampling period $T$. It is assumed that there are total $N$ sampling sensor data and the moving interval starts at $k_1 T$ and stop at $k_2 T$ ($k_1 < k_2 < N$). The rest are the zero velocity intervals. We use the subscript $k$ to describe a variable is expressed in the discrete time $k$. For example, a gyroscope output data at time $k$ is denoted by $y_{g,k}$.

**Figure 1.** Standard inertial navigation algorithm with zero velocity correction.



| Sampling time | 0 | $k_1T$ | $k_2T$ | $NT$ |
|---|---|---|---|---|
| | Not moving period | Moving period | Not moving period | |
| | Define initial values $\hat{q}_0$, $\hat{v}_0$, $\hat{r}_0$ and $P_0$ and the acceleration output $y_a$ | Compute attitude position and velocity $\hat{q}$, $\hat{v}$, $\hat{r}$ | Correct estimated data using zero velocity | |

Using the gyroscope output $y_g$ in Equation (1), the $q$, $v$ and $r$ can be estimated by the following equation ("$\wedge$" denotes for estimation):

$$\begin{aligned} \dot{\hat{q}} &= \frac{1}{2}\Omega\left(y_g\right)\hat{q} \\ \dot{\hat{v}}_n &= a_n = C^T\left(\hat{q}\right)y_a - \bar{g} \\ \dot{\hat{r}}_n &= \hat{v}_n \end{aligned} \tag{3}$$

The initial values of position $\hat{r}_0$ and velocity $\hat{v}_0$ are assumed to be zero due to the fact that the object is not moving at the not moving period and the origin of the navigation coordinate frame coincides with the starting point of the object. The initial quaternion $\hat{q}_0$ is obtained from the accelerometer data using the following (note that $a_b = 0$ during the zero velocity intervals):

$$y_a = C\left(\hat{q}_0\right)\bar{g}$$

The heading in $\hat{q}_0$ is not determined and can be chosen arbitrarily. Since the noise terms are included in $y_g$ and $y_a$ in Equation (3), $\hat{q}$, $\hat{v}$ and $\hat{r}$ are different from the true $q$, $v$ and $r$. The error in $\hat{q}$, $\hat{v}$ and $\hat{r}$ are estimated using a smoothing algorithm. Thus a smoothing algorithm is not used for directly estimating $q$, $v$ and $r$ but for estimating errors in $\hat{q}$, $\hat{v}$ and $\hat{r}$. Once the errors are estimated, we can update $\hat{q}$, $\hat{v}$ and $\hat{r}$ to obtain more accurate estimates. A multiplicative error is used for the error in $\hat{q}$ [18]. A small error in $\hat{q}$ is denoted by $q_e$. $q_e$ is assumed to satisfy the following:

$$q = \hat{q} \otimes q_e \tag{4}$$

or in matrix expression:

$$C\left(q\right) = C\left(q_e\right)C\left(\hat{q}\right) \tag{5}$$

Since we assume $q_e$ is small, it can be approximated by $q_e \approx \begin{bmatrix} 1 \\ \overline{q}_e \end{bmatrix} \in \begin{bmatrix} R \\ R^3 \end{bmatrix}$, and ⊗ is the quaternion multiplication. Therefore, the quaternion error in $q \in R^4$ is represented by $\overline{q}_e = \begin{bmatrix} \overline{q}_{e,1} & \overline{q}_{e,2} & \overline{q}_{e,3} \end{bmatrix}^T \in R^3$. Assuming $\overline{q}_e$ is small, $C(q_e)$ can be approximated by the following [19]:

$$C(q_e) \approx \begin{bmatrix} 1 & 2\overline{q}_{e,3} & -2\overline{q}_{e,2} \\ -\overline{q}_{e,3} & 1 & 2\overline{q}_{e,1} \\ 2\overline{q}_{e,2} & -2\overline{q}_{e,1} & 1 \end{bmatrix} = I - 2[\overline{q}_e \times] \tag{6}$$

where $[p\times]$ ( $p = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}^T \in R^3$) is defined by:

$$[p\times] \triangleq \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}$$

For the error in $\hat{v}$ and $\hat{r}$, an additive error model is used. We use symbols $v_e$ and $r_e$ to denote the velocity error and position error, respectively:

$$\begin{aligned} v_e &\triangleq v_n - \hat{v} \\ r_e &\triangleq r_n - \hat{r} \end{aligned} \tag{7}$$

The estimation error in $\hat{q}, \hat{v}$ and $\hat{r}$ can be expressed using the following state:

$$x(t) \triangleq \begin{bmatrix} \overline{q}_e \\ r_e \\ v_e \end{bmatrix} \in R^9$$

This $x(t)$ is estimated using a smoother. To do that, we derive a differential equation for $x(t)$. From the assumption that $\overline{q}_e$ and $\omega - y_g$ is small, we obtain the follow (see [19] for the derivation):

$$\dot{x}(t) = Ax(t) + \begin{bmatrix} -\frac{1}{2} n_g \\ 0 \\ -C^T(\hat{q}) n_a \end{bmatrix} \tag{8}$$

where:

$$A \triangleq \begin{bmatrix} [-y_g \times] & 0 & 0 \\ 0 & 0 & I \\ -2C^T(\hat{q})[y_a \times] & 0 & 0 \end{bmatrix}$$

and:

$$E\left\{\begin{bmatrix} -\frac{1}{2}n_g \\ 0 \\ -C^T(\hat{q})n_a \end{bmatrix}\begin{bmatrix} -\frac{1}{2}n_g \\ 0 \\ -C^T(\hat{q})n_a \end{bmatrix}^T\right\} = \begin{bmatrix} 0.25R_g & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & C^T(\hat{q})R_aC(\hat{q}) \end{bmatrix}$$

Since the sensor sampling period is *T*, Equation (8) is discretized with the sampling period *T* [20]:

$$x_{k+1} = A_{d,k}x_k + w_k \tag{9}$$

where $A_{d,k} \approx \exp(A(kT)T)$ and $Q_{d,k} = E\left[w_k w_k^T\right]$.

Since no external sensors other than the inertial sensor are used, there is no physical measurement during the motion. However, the fact that the velocity is zero during the not moving period can be used as a virtual measurement. In motion analysis, it is assumed that the object is not moving when the gyroscope and the variation of the accelerometer are smaller than threshold values for some specified time. Thus there will be a chance that a moving interval is detected as a zero velocity interval. To reflect this fact, a small noise $n_{v,k}$ is added in the following equation:

$$\begin{aligned} 0 - \hat{v} &= v_e + n_v \\ &= \begin{bmatrix} 0_3 & 0_3 & I_3 \end{bmatrix} x + n_v \end{aligned} \tag{10}$$

The noise $n_{v,k}$ at time $k$ is modeled as a Gaussian white noise with the covariance $R_{v,k} = E\left\{n_{v,k}n_{v,k}^T\right\}$. Equation 10 can be rewritten in form of following expression in discrete time at time $k$:

$$z_k = H_k x_k + n_{v,k} \tag{11}$$

where $z_k = \left[0 - \hat{v}_k\right]$ and $H_k = \begin{bmatrix} 0_3 & 0_3 & I_3 \end{bmatrix}$. Equation (10) can be used during the not moving interval.

We introduce a set $Z_m$ which consist of the discrete time indices belonging to the zero velocity intervals. That is, if $k \in Z_m$, then we can use the Equation (11).

A smoother problem to estimate $x_k$ can be formulated as the following optimization problem [10,11]:

Find $x_k$ and $w_k$ for $0 \le k \le N$ that minimize:

$$J(x_k, w_k) = \frac{1}{2}\sum_{k=1}^{N-1} w_k^T Q_{d,k}^{-1} w_k + \frac{1}{2}\sum_{k \in Z_m}(z_k - H_k x_k)^T R_k^{-1}(z_k - H_k x_k) + \frac{1}{2}(x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) \tag{12}$$

subject to $x_{k+1} = A_{d,k}x_k + w_k$. It is assumed that the initial value $\hat{x}_0$ and the initial covariance error $P_0$ are given. The method to choose these initial values will be discussed Section 5. The Equation (12) is posed in the maximum a posterior form. The joint a posterior probability density of $x_0$ and $w_0, \ldots, w_{k-1}$ conditioned on $\hat{x}_0, y_1, \ldots y_k$ is proportional to $\exp(-J)$, which implies that minimizing $J$ means maximizing this probability density.

By inserting the constraint $x_{k+1} = A_{d,k}x_k + w_k$ into Equation (12), we can remove the variable $w_k$ from the optimization problem:

$$J\left(x_k\right)=\frac{1}{2}\sum_{k=1}^{N-1}\left(x_{k+1}-A_d x_k\right)^T Q_{d,k}^{-1}\left(x_{k+1}-A_d x_k\right)+\frac{1}{2}\sum_{k\in Z_m}\left(z_k-H_k x_k\right)^T R^{-1}\left(z_k-H_k x_k\right) +\frac{1}{2}\left(x_0-\hat{x}_0\right)^T P_0^{-1}\left(x_0-\hat{x}_0\right). \quad (13)$$

Let the optimization variable $\bar{x}$ be defined by $\bar{x}=\begin{bmatrix} x_0^T & x_1^T & \ldots & x_N^T \end{bmatrix}^T \in R^{9(N+1)\times 1}$, then the matrix form of the optimization will be:

$$J\left(\bar{x}\right)=\frac{1}{2}\bar{x}^T M_1 \bar{x}+M_2\bar{x}^T+M_3 \quad (14)$$

where $M_1 \in R^{9(N+1)\times 9(N+1)}, M_2 \in R^{1\times 9(N+1)}, M_3 \in R$ can be computed from Equation (13). Note that Equation (14) is a quadratic function of $\bar{x}$, which can be computed efficiently using the quadratic optimization method [21]. Minimizing Equation (14) will provide a set of estimation error. From these values, $\hat{q},\hat{v}$ and $\hat{r}$ can be updated using Equations (4) and (7).

Let the minimum solution to the problem. Equation (14) be defined by $J^*$. Note that $J^*$ depends on $y_{g,k}$, $y_{a,k}$ $\left(0\le k\le N\right)$, $\hat{q}_0$, $\hat{r}_0$, $\hat{v}_0$ and $P_0$. For later use, we denote $J^*$ by a function $f$ as follows:

$$J^*=f\left(y_{g,k},y_{a,k},\hat{q}_0,\hat{r}_0,\hat{v}_0,P_0\right)=\min_{\bar{x}} J\left(\bar{x}\right) \quad (15)$$

In Equation (14), constraints can be easily added to improve the accuracy. For example, in gait analysis, while walking on a plane that assumed to be parallel with the *xy* plane of the local navigation coordinate frame, we can make use of zero $z$ axis position as following:

$$0-\begin{bmatrix}0 & 0 & 1\end{bmatrix}\hat{r}_k=\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}x_k+n_{r,k}$$

and the constraint of nonnegative $z$ axis position is given by:

$$\begin{bmatrix}0 & 0 & 1\end{bmatrix}\hat{r}_k+\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}x_k\ge 0.$$

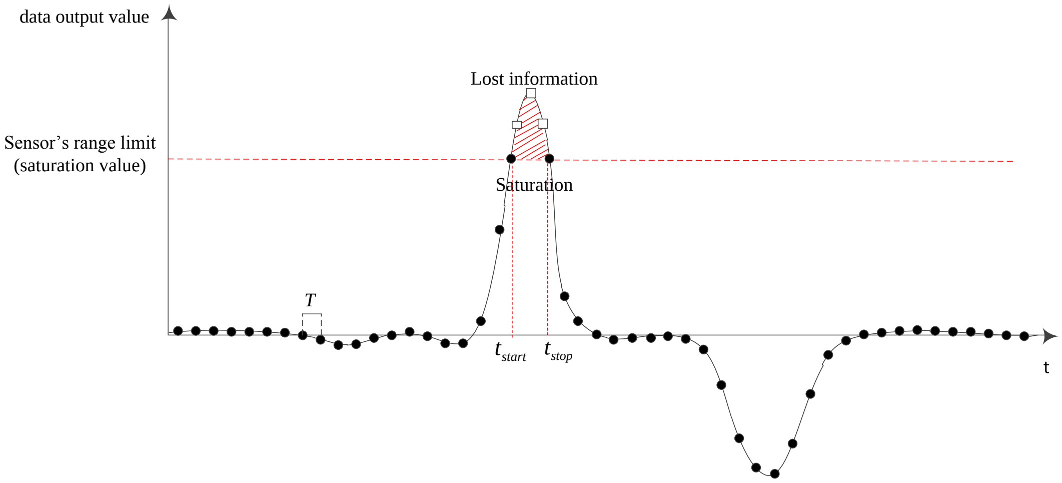
## 4. Sensor Saturation Estimation

The sensor saturation occurs when the measured values are over the sensors' dynamic ranges (Figure 2). While it is difficult to know sensor noise values ($n_g$ and $n_a$ in Equation (7)), it is not difficult to know when the sensor saturation occurs.

Each sensor has its own range of measurement. When the measured values are larger than the measurement limit, the saturation happens. Even if the saturation happens in a short time, it could lead to a large accumulated error since the lost information is in a large magnitude data area. The data loss due to the saturation has a considerable influence to the result, especially when the integration is used in data processing.

The saturation can be avoided by using large dynamic range sensors. Usually, large measuring range (with the same sensor resolution) sensors tend to be expensive. Instead of using an expensive large range sensor, we can use a low cost one with smaller measuring limit along with applying a saturation compensation algorithm.

**Figure 2.** Sensor Saturation.



The saturation can happen in gyroscopes and accelerometers in three axes. Symbols $y_{g,x,k}$, $y_{g,y,k}$, $y_{g,z,k}$ are used to denote three elements in $y_{g,k} \in R^3$. Similarly, $y_{a,x,k}$, $y_{a,y,k}$ and $y_{a,z,k}$ are used for $y_{a,k} \in R^3$. We use the fact that the value of the sensor output is smaller than the saturation value outside the saturation interval. In the saturation interval, we assume that the sensor output is equal to the saturation value. Let $y_{g,sat}$ be the saturation value of a gyroscope and $S_{g,x}$ be the set of gyroscope $x$ saturation inertial indexes:

$$\begin{cases} |y_{g,x,k}| = y_{g,x,sat} & \text{if } k \in S_{g,x} \\ |y_{g,x,k}| < y_{g,x,sat} & \text{if } k \notin S_{g,x} \end{cases}. \tag{16}$$

Similarly, we can define $S_{g,y}$, $S_{g,z}$, $y_{a,sat}$ (saturation value of an accelerometer), $S_{a,x}$, $S_{a,y}$ and $S_{a,z}$. Denote $\delta_{g,x,k}$, $\delta_{g,y,k}$, $\delta_{g,z,k}$, $\delta_{a,x,k}$, $\delta_{a,y,k}$, $\delta_{a,z,k}$, the compensation values of gyroscopes and accelerations in $x, y, z$ axes at the time $k$, respectively. The compensated sensor value ($\bar{y}_{g,k}$ and $\bar{y}_{a,k}$) is the sum of sensor output value and compensation value. For example, the compensated $x$ axis gyroscope value is given by:

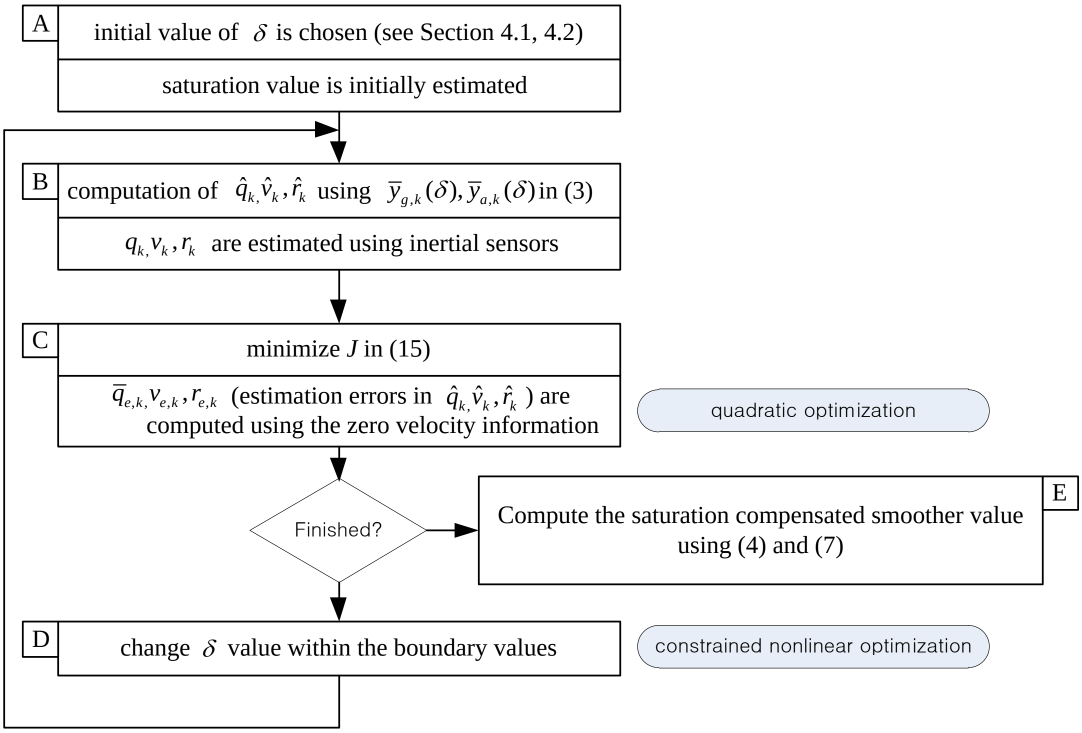
$$\bar{y}_{g,x,k} = \begin{cases} y_{g,x,k} + \delta_{g,x,k} & \text{if } k \in S_{g,x} \\ y_{g,x,k} & \text{if } k \notin S_{g,x} \end{cases} \tag{17}$$

Let $\delta$ be set of $\delta_{g,x,k}$, $\delta_{g,y,k}$, $\delta_{g,z,k}$, $\delta_{a,x,k}$, $\delta_{a,y,k}$ and $\delta_{a,z,k}$ variables. For example, if $S_{g,x} = \{5,6,7\}$, $S_{a,y} = \{8,9,10\}$, and $S_{g,y} = S_{g,z} = S_{a,x} = S_{a,z} = \varnothing$ then $\delta$ is given by:

$$\delta = \begin{bmatrix} \delta_{g,x,5} & \delta_{g,x,6} & \delta_{g,x,7} & \delta_{a,y,8} & \delta_{a,y,9} & \delta_{a,y,10} \end{bmatrix}^T \in R^6.$$

Now the standard smoother algorithm in Section 3 is modified using $\bar{y}_{g,k}$ and $\bar{y}_{a,k}$. The flowchart of the algorithm is given in Figure 3. The first step (step A in Figure 3) of the algorithm is the initial estimation of $\delta$, which is explained in Sections 4.1 and 4.2. Given $\delta$ values, the standard smoother algorithm (steps B and C) is applied to compute the smoother. This standard algorithm (given in Section 3) can be formulated as the quadratic optimization algorithm. How good the computed smoother is can be evaluated using the computed $J$ value in Equation (15). This process (steps B and C) is repeated by changing $\delta$ values. The algorithm finishes if the minimum value of $J$ is found. We note that $\delta$ optimization can be formulated as a constrained nonlinear optimization problem. Once the minimization process is done, we can compute the saturation compensated smoother values (step E).

**Figure 3.** Proposed saturation compensated smoothing algorithm.


A
initial value of $\delta$ is chosen (see Section 4.1, 4.2)
saturation value is initially estimated
B
computation of $\hat{q}_{k,}\hat{v}_k,\hat{r}_k$ using $\bar{y}_{g,k}(\delta),\bar{y}_{a,k}(\delta)$ in (3)
$q_{k,}v_k,r_k$ are estimated using inertial sensors
C
minimize $J$ in (15)
$\bar{q}_{e,k,}v_{e,k},r_{e,k}$ (estimation errors in $\hat{q}_{k,}\hat{v}_k,\hat{r}_k$ ) are computed using the zero velocity information
quadratic optimization
Finished?
E
Compute the saturation compensated smoother value using (4) and (7)
D
change $\delta$ value within the boundary values
constrained nonlinear optimization

Consider the following function (from Equation (15)):

$$f\left(\bar{y}_{g,k}(\delta),\bar{y}_{a,k}(\delta),\hat{q}_0,\hat{r}_0,\hat{v}_0,P_0\right) \tag{18}$$

Assuming $\hat{q}_0,\hat{r}_0,\hat{v}_0,P_0,y_{g,k}$ and $y_{a,k}$ are constant, the function $f$ in Equation (18) depends on $\delta$. In this section, $f$ is minimized with respect to $\delta$. Two different methods are proposed. The first method directly minimize $\delta$ with respect to all possible combination of $\delta$. The second method uses the geometric structure of the sensor saturation.

### 4.1. Method 1: Direct Estimation of $\delta$

In the first method, the following optimization problem is solved:

$$\min_{\delta} f\left(\bar{y}_{g,k}(\delta),\bar{y}_{a,k}(\delta),\hat{q}_0,\hat{r}_0,\hat{v}_0,P_0\right)=\min_{\delta}\min_{\bar{x}} J\left(\bar{x}\right) \tag{19}$$

subject to:

$$\begin{cases}\delta\le\delta_b & \text{if } \delta\ge 0\\ \delta\ge-\delta_b & \text{if } \delta\le 0\end{cases} \tag{20}$$

In Equation (20), "≤" and "≥" represent element-wise inequalities and $\delta_b$ is a set of bounding positive values of elements in $\delta$. Choice of $\delta_b$ depends on applications. For example, in knee gyroscope data, the maximum angular velocity of a human knee varies from 213 to 1,087 $°/s$ [22]; in gait acceleration, the maximum foot acceleration is around 11.82 $m/s^2$ for walking case [23]. Therefore, the gyroscope data upper bound can be chosen as $\delta_b = [1{,}087 - y_{g,sat}\ 11.82 - y_{a,sat}]^T$. From Equations (16) and (17), it is easily to be seen that when $y_g, y_a \geq 0$, we have $\delta \geq 0$ because gyroscope compensated values ($\bar{y}_g, \bar{y}_a$) are larger than saturation value ($y_{g,sat}, y_{a,sat}$). The initial value of $\delta$ can be chosen as a set of 0 and varies in a range which satisfies the condition in Equation (20). With each set of $\delta$, one value of $\bar{x}$ is obtained. The minimum value of $\delta$ which makes $\bar{x}$ minimizes the quadratic problem (14) is chosen.

### *4.2. Method 2: Estimation of $\delta$ Using Geometric Form*

In this method, the saturated sensor data region is approximated by a triangle (see Figure 4a) and quadratic function (see Figure 4b). Using three data before the saturation interval ($x_{k-2}, x_{k-1}, x_k$ in Figure 4a), we can obtain a quadratic function. In the next step, a line $l_1$ starting from $x_k$ with the slope $\xi'(k)$ is generated. With the same procedure (quadratic function $g(t)$ for $x_h, x_{h+1}, x_{h+2}$), we generate a line $l_2$. $l_1$ and $l_2$ are the reconstruction part of sensor saturation. The intersection point of $l_1$ and $l_2$ (at the time $t_m$) is the maximum value ($x_m$) for compensation. Now consider $l_1$ is created based on $(x_k, k)$ and $(x_m, t_m)$ points, $l_2$ is created based on $(x_h, h)$ and $(x_m, t_m)$ points. Changing the height of the intersection from saturation value to maximum compensation value $x_m$ we can generate several sets of basic lines $l_1, l_2$ which contain compensation values. The intersection point is defined as following:

$$\begin{aligned} t_m &= \frac{g(h)-\xi(k)-(h-k)g'(h)}{\xi'(k)+g'(h)} \\ x_m &= \xi(k)+t_m\xi'(t_m) \end{aligned} \tag{21}$$

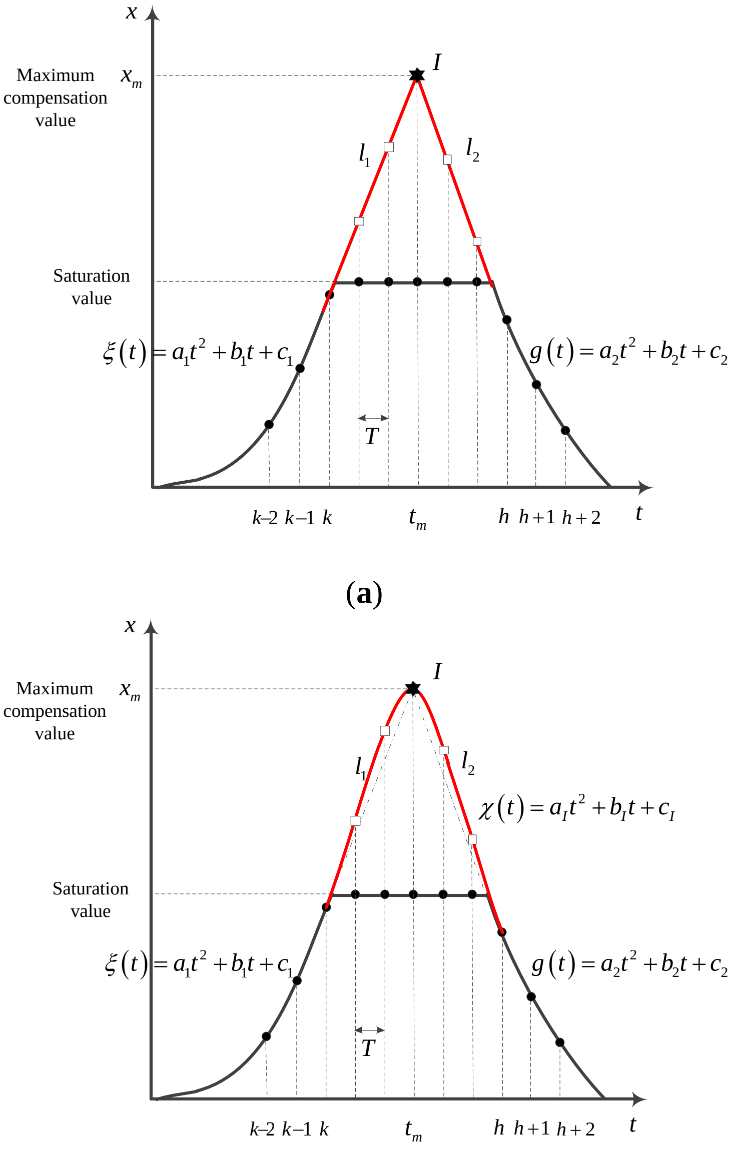
The other compensation value is defined by:

$$\begin{aligned} x_t &= \xi(k)+\frac{x_m-\xi(k)}{t_m-k}(t-k) \quad \text{if } k<t<t_m \\ x_t &= g(h)+\frac{g(h)-x_m}{h-t_m}(t-h) \quad \text{if } t_m<t<h \end{aligned} \tag{22}$$

Using this method, sets of sensor's compensation values $\delta$ can be generated by changing the intersection point's value from saturation value to maximum compensation value. In this case, $\delta$ only depends on intersection point. Denote $I(x_I, t_I)$ the intersection point. Once $I$ is created, the rest compensation data will be obtained based on $l_1$ and $l_2$ lines using Equation (22). The optimization problem (19) will subject to $\delta$ value which relates to the intersection point.

With the same idea, a quadratic approximation can be used to generate the lost information (see Figure 3b. Firstly the maximum compensation value can be generated using the same procedure in triangle approximation process. After this step, a quadratic function $\chi(t)$ is formed up based on $(x_k, t_k), (x_h, t_h)$ and $(x_m, t_m)$ points. In saturation parts, the compensation value at time $t_k$ is $\chi(t_k)$.

**Figure 4.** Estimation of $\delta$ using geometric form. (**a**) Triangle approximation; (**b**) Quadratic approximation.



(**a**)

(**b**)

## 5. Sensor Saturation Compensation Algorithm with Multiple Zero Velocity Intervals

Section 4 introduced two methods to compensate sensor saturation in a standard movement situation which contains two zero velocity intervals at the beginning and ending of a moving interval. In this section, we apply the Section 4 methods to multiple zero velocity interval movements. A movement with multiple zero velocity intervals is displayed in Figure 5. In this movement, moving intervals are interposed by zero velocity intervals. An example of this movement can be found in gait analysis in [4,5]. In the characteristic human gait, one foot is assumed to be not moving when the the center of mass is put on the corresponding leg to move the other. Therefore, during the walking process, there are zero velocity intervals separated by moving intervals for each foot.

We divide the movement into segments based on zero velocity intervals so that between two zero velocity intervals there is data from one moving period (see Figure 5). If there are saturations in the movement, we can apply the compensation methods in Section 4 to each segment. The information of the last sampling in the prior segment will be used as the initial value for the following one. The initial error covariance for each segment can be estimated using a Kalman filter with a zero velocity measurement update for previous segment. Note that for the first segment the position and velocity errors are assumed to be zero due to the fact that the initial body coordinate frame coincides with the navigation coordinate frame. The initial covariance of the first segment is also assumed to be small. Repeating the smoothening processes until the last segment, the whole smoothed data is obtained.

**Figure 5.** A movement with multiple zero velocity intervals.

| Segment 1 | | Segment 2 | | | | Segment N | | |
|---|---|---|---|---|---|---|---|---|
| Zero-velocity interval | Moving interval | Zero-velocity interval | Moving interval | Zero-velocity interval | • • • | Zero-velocity interval | Moving interval | Zero-velocity interval |

## 6. Simulation and Experiments

In this section, one simulation result and two experimental results are given to verify the proposed algorithm. First, a simulation is done to verify the proposed method. The sensor is assumed to be located at the end of a bar, which is rotated along the body $y$ axis. There is a sensor's saturation in the gyroscope $y$ axis data as in Figure 6. The saturated data (green "—" line) is obtained with $y_{g,y,sat} = 7.5$. This saturated data is used as an input data for the compensation algorithms in Section 4.

**Figure 6.** Simulation gyroscope $y_{g,y}$ data with and without saturation ($y_{g,y,sat} = 7.5$).

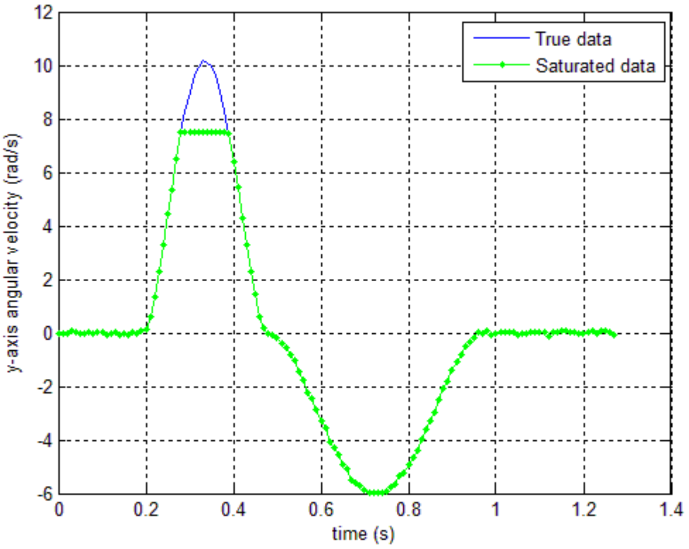
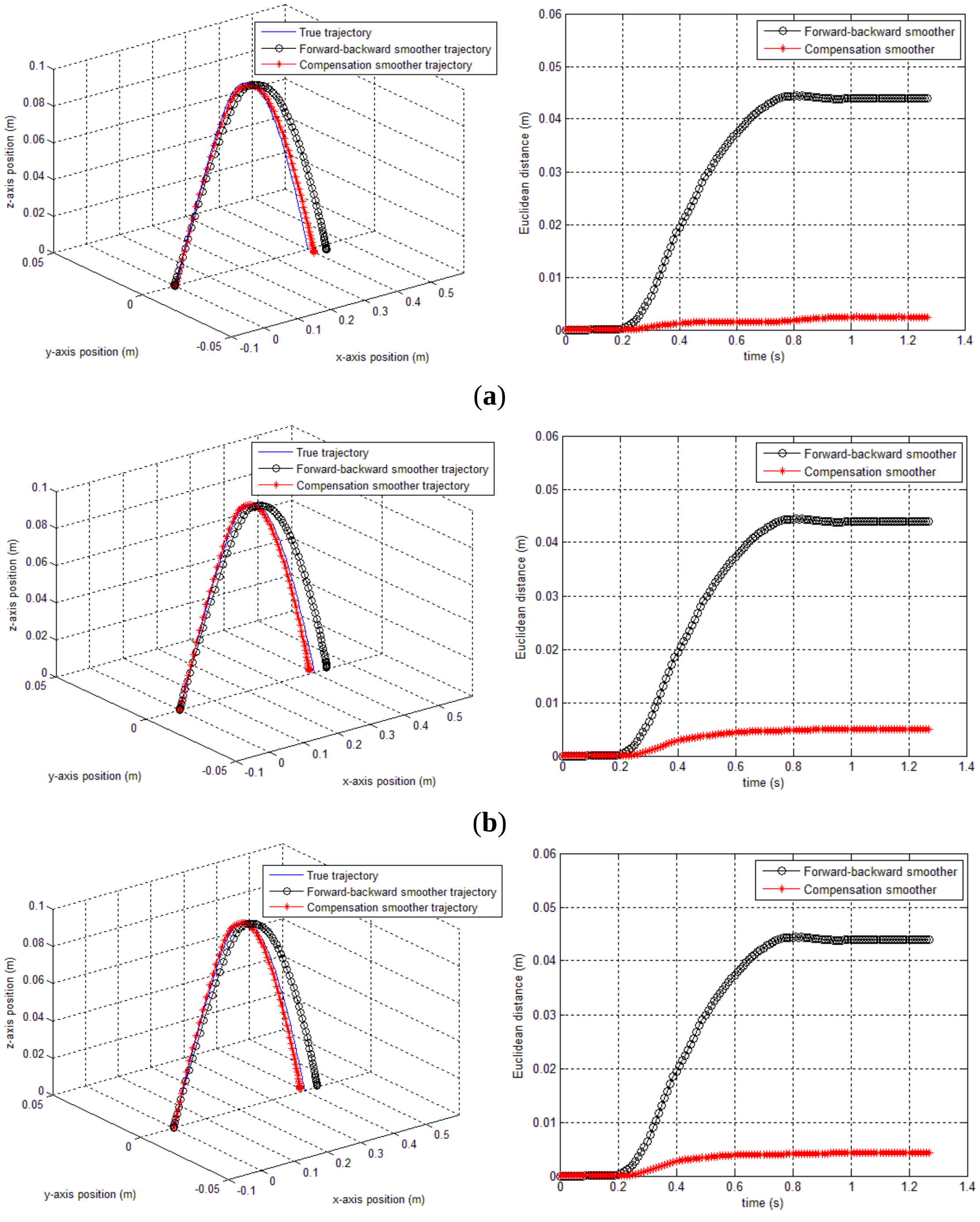


**Figure 7.** True and estimated trajectories. (**a**) method 1 result: 3D trajectory and Euclidean distance over time; (**b**) method 2 (triangle approximation): 3D trajectory and Euclidean distance over time; (**c**) method 2 (quadratic approximation): 3D trajectory and Euclidean distance over time.



(**a**)

(**b**)

(**c**)

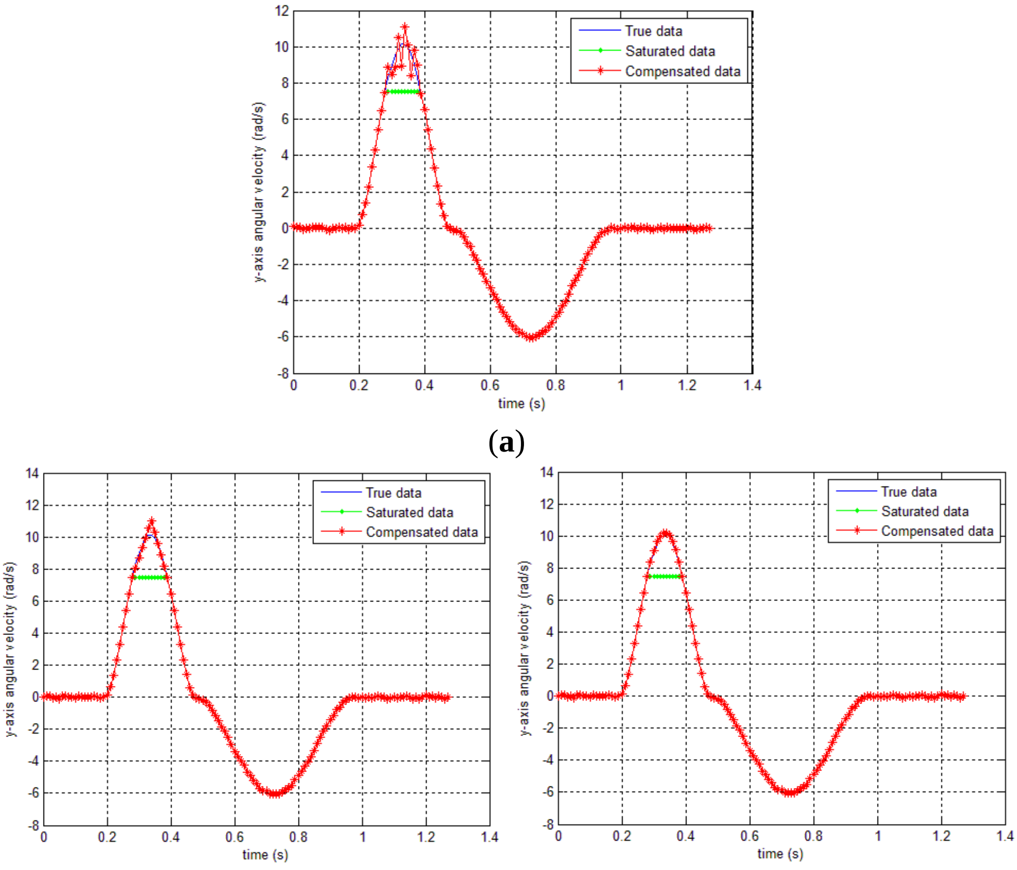
In this simulation, $\delta$ only contains $y$ axis compensation ($\delta_{g,y,k}$). The smoothing results (forward-backward smoother [20] and the proposed methods) are given in Figure 7 and Table 1. In case of method 1, $\delta_b$ is chosen as 5, so that the performance index $J$ is minimized subject to $0 \leq \delta \leq 5$. Similarly, method 2 (both triangle and quadratic approximation) is applied. In Table 1, we can see that both method 1 and 2 produce significant improvements over the conventional smoothing (forward-backward smoother) result. Among the proposed methods, method 1 gives the best result and the second best is method 2 (quadratic approximation). We note that the method 1 requires more computation than method 2 since it needs more optimization variables. As for method 2, it is not conclusive whether quadratic approximation is better than triangle approximation since triangle approximation sometimes gives better results as seen in our later experiment (see Table 2).

**Table 1.** Last position accuracy of different smoothers.

| Estimation Method | Last Position Error | Position Error Norm |
|---|---|---|
| Forward-backward smoother | 0.0458 | 0.1544 |
| Proposed smoother (method 1) | 0.0032 | 0.0009 |
| Proposed smoother (method 2: triangle approximation) | 0.0045 | 0.0019 |
| Proposed smoother (method 2: quadratic approximation) | 0.0036 | 0.0014 |

In Figure 8, the gyroscope sensor saturation compensation results by the proposed methods are given. In all three cases, it can be seen that the sensor saturation is well estimated.

**Figure 8.** $y_{g,y}$ data compensation result by the proposed methods. (**a**) sensor compensation result by method 1; (**b**) sensor compensation result by method 2 (triangle and quadratic approximation).



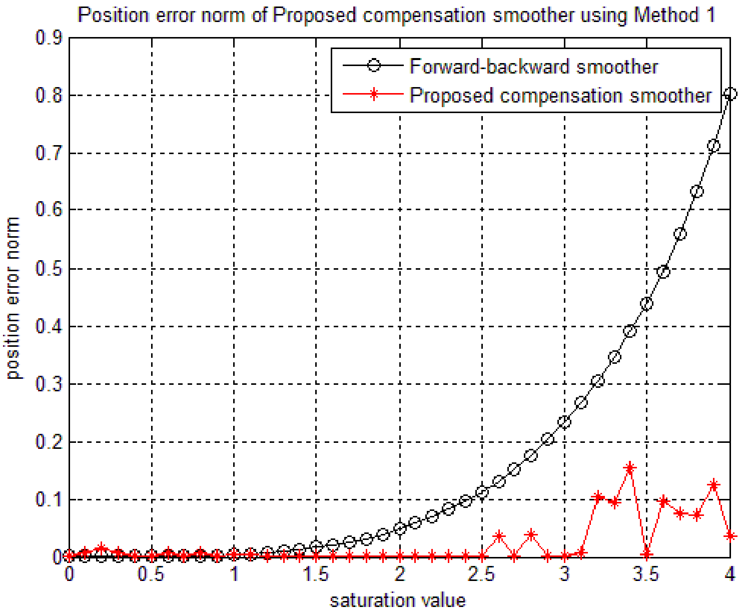
(**a**)

(**b**)

In Figure 7, the gyroscope sensor saturation compensation results by the proposed methods are given. In all three cases, it can be seen that the sensor saturation is well estimated.

In order to verify the robustness of the proposed algorithms to the saturation, a simulation is done by checking the position norm errors while $y_{g,y,sat}$ value is changed. For example, if we set $y_{g,y,sat}$=12 in Figure 6, there is no saturation in the sensor. On other hand, if we decrease $y_{g,y,sat}$ value, the sensor saturation increases. With the method 1, when the $\delta_b$ is fixed and the saturation value is changed, the accuracy of the proposed smoother may be affected (if the $\delta_b$ is chosen not large enough). As can be seen from Figure 9 where the $\delta_b$ is chosen as 2.5, the accuracy will decrease when the compensation value that is used to compensate the saturated data is larger than $\delta_b$. However, this effect can be avoided since the saturation value is known for each sensor. Therefore, we can choose $\delta_b$ as a large number compared with the sensor's saturation value.

**Figure 9.** The effect of saturation value (in gyroscope data) on the method 1 smoother accuracy.



Sections 4.1 and 4.2 showed that the accuracy of the method 2 only depends on the saturation value while the accuracy of the method 1 is affected by the saturation value and the $\delta_b$. Figure 10 illustrates that the saturation is well compensated by method 2, even when the saturation value is changing. In general, compared with a forward-backward smoother, our proposed compensation smoother has better performance.

As mentioned above, how to choose $\delta_b$ may affect the accuracy. In some applications, the $\delta_b$ is known from doing statistical experiments. In the case where $\delta_b$ is unknown, we can choose an arbitrarily large value. This does not affect the result as shown in Figure 11. In Figure 11, when $\delta_b$ is chosen smaller than the compensation value that is used to compensate the saturated data, the error could be large. If we increase $\delta_b$, the error decreases since a larger saturation can be compensated. However, it could lead to more computations for solving the optimization Equation (19).

**Figure 10.** The effect of saturation value (in gyroscope data) on the method 2 smoother accuracy.

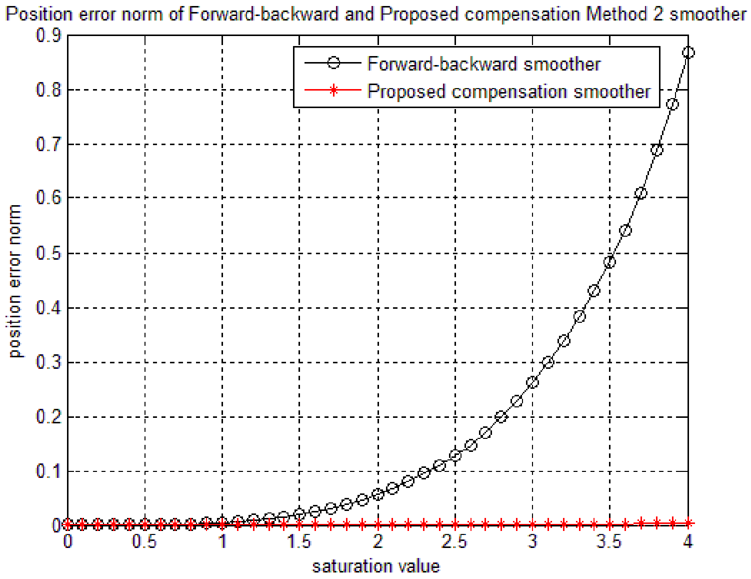
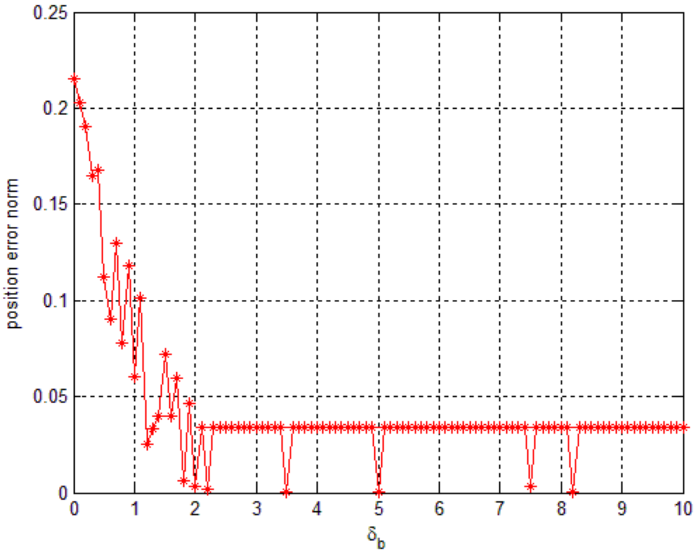


**Figure 11.** The effect of $\delta_b$ on the method 1 smoother accuracy.



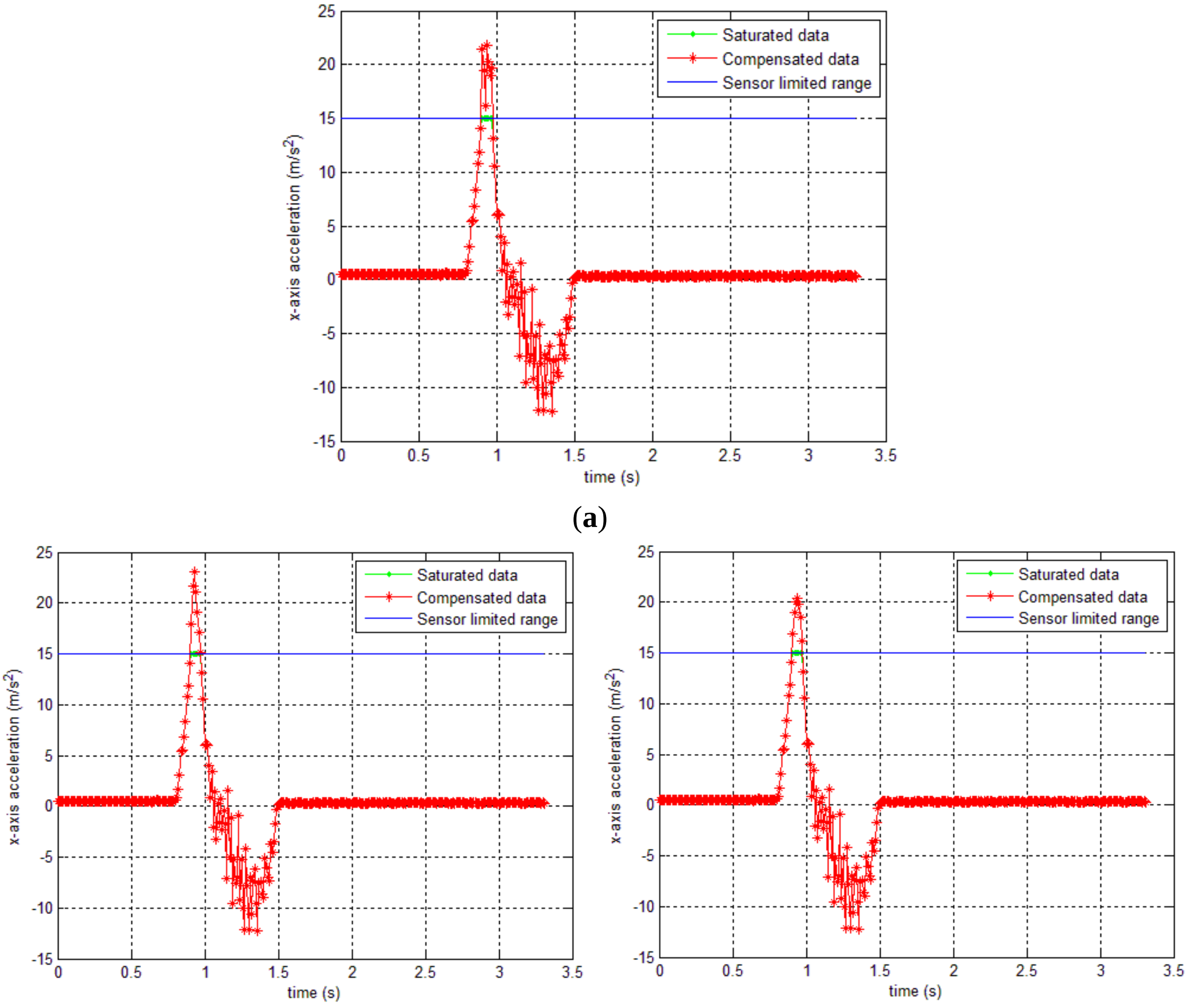
In the first experiment, an object, which is attached with an IMU on the top, moved in a straight line of 0.95 m so that the $x$ axis of the IMU coincides with direction of movement. In this case, there exists saturation in the sensor's accelerometer in the *x* axis ( $y_{a,x}$ ). The trajectory of the object is estimated by the forward-backward smoother and our proposed compensation smoother using two methods. Figure 12 shows that the saturated $y_{a,x,k}$ data was compensated using the proposed compensation smoothers.

The estimated position errors are given in Figure 13 (method 1 only, since the results of method 2 are similar) and Table 2. As can be seen from Figure 13, the proposed compensation smoother using method 1 gives a closer result to the true position (0.9121 m, equivalent to 4% error) than the forward-backward smoother (0.8404 m, equivalent to 11.54% error). The proposed compensation smoother using method 2 also gives a good result of 0.9084 m (4.39% error) and 0.8976 m (5.52% error) for triangle and quadratic approximations, respectively.

**Table 2.** The error of 0.95 m straight movement experiment.

| Estimation Method | Last Position Error |
|---|---|
| Forward-backward smoother | 0.1096 |
| Proposed smoother (method 1) | 0.0379 |
| Proposed smoother (method 2: triangle approximation) | 0.0416 |
| Proposed smoother (method 2: quadratic approximation) | 0.0524 |

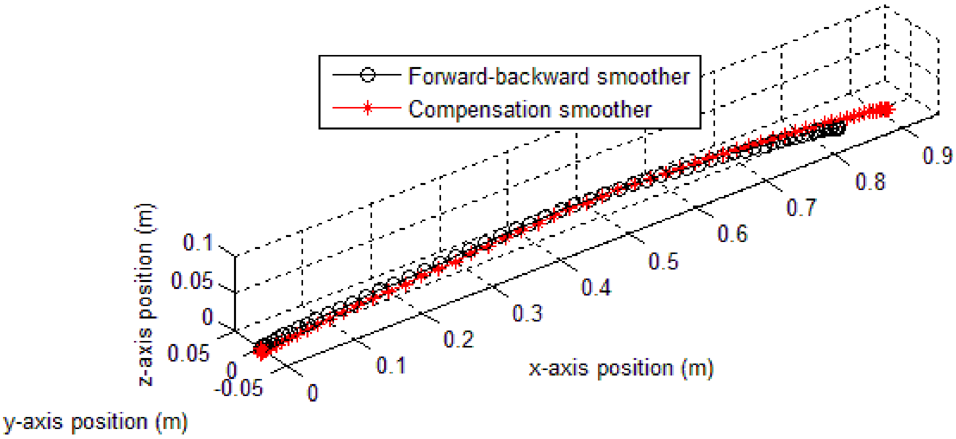
**Figure 12.** $y_{a,x,k}$ sensor's output and compensated data of 0.95 m straight movement experiment; (**a**) method 1 result; (**b**) method 2 results (triangle and quadratic approximation).



(**a**)

(**b**)

**Figure 13.** Trajectories of 0.95 m straight movement experiment (method 1 result).



Another experiment has been done to verify the compensation feasibility of the proposed algorithm. In this experiment, an IMU is attached at the tip of a digitizer (see Figure 14). The tip was moved in a curved line. The position data of the tip was recorded by the digitizer while the trajectory of the IMU is estimated by the proposed compensation and the forward-backward smoothers, respectively. The movement was made so that there is saturation in the $y_{g,z,k}$ data. The estimated trajectories are compared with the true trajectory of the digitizer in Figure 15 (in millimeters). The distance between the start and stop point is used as an evaluation criterion. Based on this criterion, a table of distance errors of the estimated positions and the digitizer's data is formed, as shown in Table 3. Table 3 shows that method 1 gives a best accuracy (7.9 mm error) compared with other estimations. The forward-backward smoother has a worst estimation with a 50.4 mm error result. Moreover, the two approximation methods in method 2 provide similar results (11.8 and 10.3 mm errors).

**Figure 14.** Curve movement experiment setup.

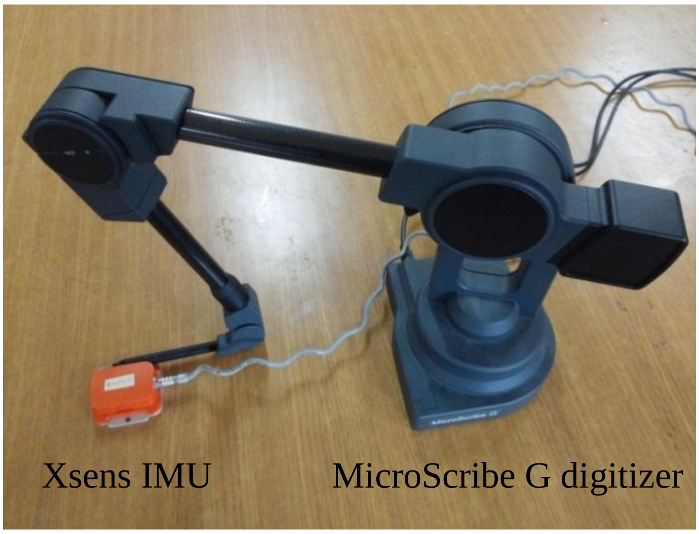


**Figure 15.** Trajectories of a curve movement experiment (method 1).

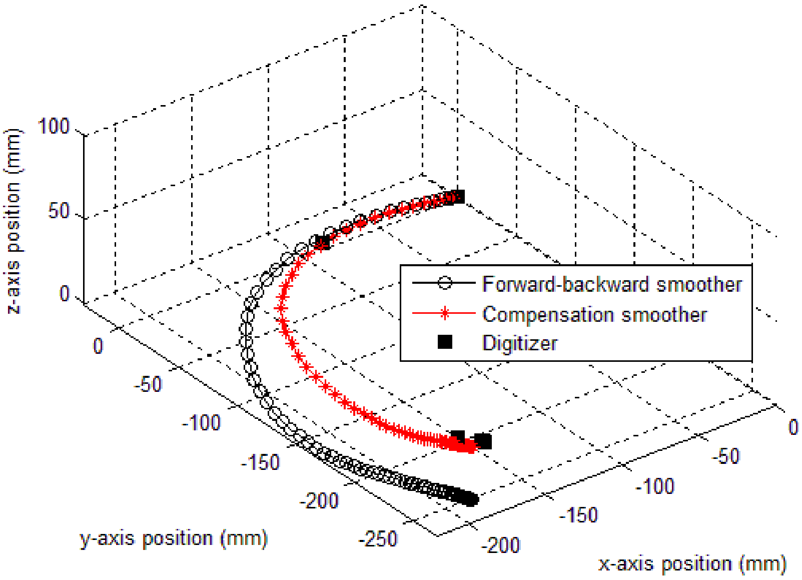
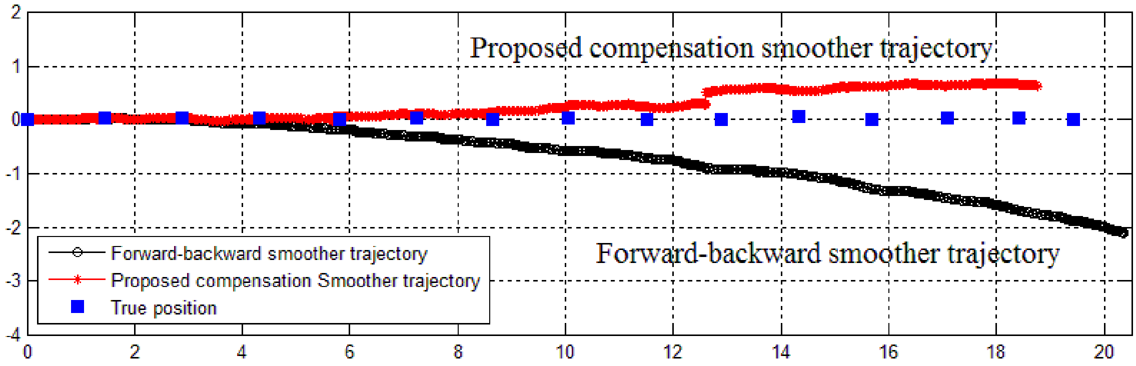


**Table 3.** The start-stop point distance error of curve movement experiment (in mm).

| Estimation Method | Distance Error |
|---|---|
| Forward-backward smoother | 50.4 |
| Proposed smoother (method 1) | 7.9 |
| Proposed smoother (method 2: triangle approximation) | 11.8 |
| Proposed smoother (method 2: quadratic approximation) | 10.3 |

In the last experiment, we verify the compensation smoother application in multiple zero velocity intervals movement (in Section 5). In this experiment, an IMU is attached on a human foot. The volunteer was asked to walk along a straight corridor. A pen is also attached on the volunteer's shoe to mark the steps' positions on the floor. The obtained data from IMU is used to estimate the trajectory of the foot. A comparison of proposed and forward-backward smoothers trajectories is given in Figure 16. The result shows that the last position error of proposed smoother trajectory is 0.9042 m, while it is 2.3128 m for the forward-backward smoother trajectory.

**Figure 16.** True and estimated trajectories of a walking person.



## 7. Conclusions

This paper has proposed some approaches to compensate for sensor saturation. Saturation is a common problem with IMU sensors in tracking a moving object. The lost data in the saturated parts could be important due to the accumulated errors. In the paper, the authors used a standard smoothing algorithm with zero velocity intervals to compensate sensor saturation. The considered motion includes a moving interval between two zero velocity intervals. Two methods were proposed. The first method directly estimates the saturation compensation while the second one uses a geometric form to estimate the saturation. The proposed smoothing algorithm can be applied in some motions which contain many moving intervals separated by zero velocity intervals. In this case, the motion is divided into segments based on zero-velocity intervals so that between two zero velocity intervals there is one moving interval. The saturation estimation algorithm is applied in each segment from the first to the last one. To verify the feasibility of the two methods, some experiments have been done. The experiments showed that the proposed smoothing algorithm can compensate the sensor saturation and provides a smaller error than a conventional smoother (forward-backward filter). In practical applications, the sensor saturation compensation methods proposed in this paper can be used to improve the accuracy of small dynamic range sensors instead of using a large dynamic range one which usually tends to be more expensive.

## Acknowledgment

This work was supported by 2013 Research Funds of Hyundai Heavy Industries for University of Ulsan.

## Author Contributions

The paper is a collaborative work of two authors. The problem and the saturation estimation, which consists of direct and triangle estimations, are proposed by Young Soo Suh. Quoc Khanh Dang contributed the quadratic approximation, designed experiments, collected and analyzed data. Dang wrote the paper, and Suh reviewed and edited the manuscript.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Hong, J. Gait Analysis and Identification. In Proceedings of the 18th International Conference on Automation & Computing, Loughborough University, Leicestershire, UK, 8 September 2012; pp. 1–6.
2. Lee, L.; Grimson, W.E.L. Gait Analysis for Recognition and Classification. In Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, USA, 20–21 May 2002; pp. 148–155.

3. Tadano, S.; Takeda, R.; Miyagawa, H. Three Dimensional Gait Analysis Using Wearable Acceleration and Gyro Sensors Based on Quaternion Calculations. *Sensors* **2013**, *13*, 9321–9343.
4. Do, T.N.; Suh, Y.S. Gait Analysis Using Floor Marker and Inertial Sensors. *Sensors* **2012**, *12*, 1594–1611.
5. Tran, N.H.; Suh, Y.S. Inertial Sensor-Based Two Feet Motion Tracking for Gait Analysis. *Sensors* **2013**, *13*, 5614–5629.
6. Tao, W.; Liu, T.; Zheng, R.; Feng, H. Gait Analysis Using Wearable Sensors. *Sensors* **2012**, *12*, 2255–2283.
7. Hoffmann, J.; Bruggemann, B.; Kruger, B. Auto Calibration of a Motion Capture System Based on Inertial Sensors for Tele-manipulation. In Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Funchal, Madeira, Portugal, 15–18 June 2010.
8. Helten, T.; Muller, M.; Tautges, J.; Weber, A.; Seidel, H.P. Towards Cross-modal Comparison of Human Motion Data. In Proceedings of the 33rd Annual Symposium of the German Association for Pattern Recognition (DAGM), Frankfurt/Main, Germany, 31 August–2 September 2011; pp. 61–70.
9. Tao, G.; Huang, Z.; Sun, Y.; Yao, S.; Wu, J. Biomechanical Model-based Multi-sensor Motion Estimation. *Sens. Appl. Symp. (SAS)* **2013**, *23*, 19–21.
10. Psiaki, M.L. Backward-Smoothing Extended Kalman Filter. *J. Guid. Control Dyn.* **2005**, *28*, 885–894.
11. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation*; John Wiley & Sons: New York, NY, USA, 2001.
12. Simon, D. Kalman Filtering with State Constraints: A Survey of Linear and Nonlinear Algorithms. *IET Control Theory Appl.* **2010**, *4*, 1303–1318.
13. Kuipers, J.B. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*; Princeton University Press: Princeton, NJ, USA, 1999.
14. Collinson, R.P. *Introduction to Avionics Systems*, 2nd ed.; Springer: New York, NY, USA, 2003.
15. Luinge, H.J.; Veltink, P.H. Measuring Orientation of Human Body Segments Using Miniature Gyroscopes and Accelerometers. *Med. Biol. Eng. Comput.* **2005**, *43*, 273–282.
16. Aggarwal, P.; Syed, Z.; Niu, X.; El-Sheimy, N. A Standard Testing and Calibration Procedure for Low Cost MEMS Inertial Sensors and Units. *J. Navig.* **2008**, *61*, 323–336.
17. Nam, C.N.K.; Kang, H.J.; Suh, Y.S. Golf Swing Motion Tracking Using Inertial Sensors and a Stereo Camera. *IEEE Trans. Instrum. Meas.* **2013**, *63*, 943–952.
18. Creamer, G. Spacecraft Attitude Determination Using Gyros and Quaternions Measurements. *J. Austronautical Sci.* **1996**, *44*, 357–371.
19. Suh, Y.S. Orientation Estimation Using a Quaternion-Based Indirect Kalman Filter with Adaptive Estimation of External Acceleration. *IEEE Trans. Instrum. Meas.* **2010**, *59*, 3296–3305.
20. Brown, R.G.; Hwang, P.Y.C. *Introduction to Random Signals and Applied Kalman Filtering*; John Wiley & Sons: New York, NY, USA, 1997.
21. Boyd, S; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004.
22. Bober, T.; Putnam, C.A.; Woodworth, G.G. Factors Influencing the Angular Velocity of a Human Limb Segment. *J. Biomech.* **1987**, *20*, 511–521.

23. More Acceleration Perturbations of Daily Living. Available online: http://hypertextbook.com/facts/2006/accelerometer.shtml (accessed on 17 April 2014).