

Article

Spline Function Simulation Data Generation for Walking Motion Using Foot-Mounted Inertial Sensors

Thanh Tuan Pham  and Young Soo Suh * 

Electrical Engineering Department, University of Ulsan, Ulsan 44610, Korea; josephpham1194@gmail.com

* Correspondence: yssuh@ulsan.ac.kr; Tel.: +82-52-259-2196

Received: 22 October 2018; Accepted: 20 December 2018; Published: 23 December 2018



Abstract: This paper investigates the generation of simulation data for motion estimation using inertial sensors. The smoothing algorithm with waypoint-based map matching is proposed using foot-mounted inertial sensors to estimate position and attitude. The simulation data are generated using spline functions, where the estimated position and attitude are used as control points. The attitude is represented using B-spline quaternion and the position is represented by eighth-order algebraic splines. The simulation data can be generated using inertial sensors (accelerometer and gyroscope) without using any additional sensors. Through indoor experiments, two scenarios were examined include 2D walking path (rectangular) and 3D walking path (corridor and stairs) for simulation data generation. The proposed simulation data is used to evaluate the estimation performance with different parameters such as different noise levels and sampling periods.

Keywords: motion estimation; inertial sensors; simulation; spline function; Kalman filter

1. Introduction

Motion estimation using inertial sensors is one of the most important research topics that is increasingly applied in many application areas such as medical applications, sports, and entertainment [1]. Inertial measurement unit (IMU) sensors are commonly used to estimate human motion [2–5]. Inertial sensors can be used alone or combined with other sensors such as cameras [6,7] or magnetic sensors [8]. In personal navigation and healthcare applications [9–11], foot-mounted inertial sensors without using any additional sensors is a key enabling technology since it does not require any additional infrastructure. If additional sensors (other than inertial sensors) are also used, sensor fusion algorithms [12,13] can be used to obtain more accurate motion estimation.

To evaluate the accuracy of any motion estimation algorithm, it is necessary to compare the estimated value with the true value. The estimated value is usually verified through experiments with both IMU and optical motion tracker [14,15]. In [2], the Vicon optical motion capture system is used to evaluate the effectiveness of human pose estimation method and its associated sensor calibration procedure. An optical motion tracking is used as a reference to compare with motion estimation using inertial sensors in [15].

Although the experimental validation gives the ultimate proof, it is not only more convenient but also more important to work with simulation data, which is statistically much more relevant for testing and the development of algorithms. For example, the effect of gyroscope bias on the performance of an algorithm cannot be easily identified in experiments. In this case, simulation using synthesized data is more convenient since gyroscope bias can be changed arbitrarily.

In [16–18], IMU simulation data are generated using optical motion capture data for human motion estimation. In [19], IMU simulation data is generated for walking motion by approximating walking trajectory as simple sinusoidal functions. In [20–22], IMU simulation data are also generated for flight motion estimation using an artificially generated trajectory. There are many different methods

to represent attitude and position simulation data. Among them, the spline function is the most popular approach [23]. The B-spline quaternion algorithm provides a general curve construction scheme that extends the spline curves [24]. Reference [25] presents an approach for attitude estimation using cumulative B-splines unit quaternion curves. Position can be represented by any spline function [23]. Usually, position data are represented using cubic spline functions. The cubic spline function gives a continuous position, velocity, and acceleration. However, the acceleration is not sufficiently smooth. This is a problem for the inertial sensor simulation data generation since the smooth accelerometer data cannot be generated. Thus, we use the eighth-order spline function for position data since it gives position data with smooth acceleration, whose jerk (third derivative) can be controlled in the optimization problem.

In this paper, we use cumulative B-spline for attitude representation and eighth-order algebraic spline for position representation. The eighth-order algebraic spline is a slightly modified version of the spline function from a previous study [26].

The aim of this paper is to obtain simulation data combining inertial sensors (accelerometer and gyroscope) and waypoint data without using any additional sensors. A computationally efficient waypoint-based map matching smoothing algorithm is proposed for position and attitude estimation from foot-mounted inertial sensors. The simulation data are generated using spline functions, where the estimated position and attitude are used as control points. This paper is an extended version of a work published in the conference paper [27], where a basic algorithm with a simple experimental result is given.

The remainder of the paper is organized as follows. Section 2 describes the smoothing algorithm with waypoint-based map matching to estimate motion for foot-mounted inertial sensors. Section 3 describes the computation of the spline function to generate simulation data. Section 4 provides the experiment results to verify the proposed method. Conclusions are given in Section 5.

2. Smoothing Algorithm with Waypoint-Based Map Matching

In this section, a smoothing algorithm is proposed to estimate the attitude, position, velocity and acceleration, which will be used to generate simulation data in Section 3.

Two coordinate frames are used in this paper: the body coordinate frame and the navigation coordinate frame. The three axes of the body coordinate frame coincide with the three axes of the IMU. The z axis of the navigation coordinate frame coincides with the local gravitational direction. The choice of the x and y axes of the navigation coordinate frame can be arbitrarily chosen. The notation $[p]_n$ ($[p]_b$) is used to denote that a vector p is represented in the navigation (body) coordinate frame.

The position is defined by $[r]_n \in R^3$, which is the origin of the body coordinate frame expressed in the navigation coordinate frame. Similarly, the velocity and the acceleration are denoted by $[v]_n \in R^3$ and $[a]_n \in R^3$, respectively. The attitude is represented using a quaternion $q \in R^4$, which represents the rotation relationship between the navigation coordinate frame and the body coordinate frame. The directional cosine matrix corresponding to quaternion q is denoted by $C(q) \in SO(3)$.

The accelerometer output $y_a \in R^3$ and the gyroscope output $y_g \in R^3$ are given by

$$\begin{aligned} y_a &= C(q)\tilde{g} + a_b + n_a, \\ y_g &= \omega + n_g, \end{aligned} \quad (1)$$

where $\omega \in R^3$ is the angular velocity, $a_b \in R^3$ is the external acceleration (acceleration related to the movement, excluding the gravitational acceleration) expressed in the body coordinate frame, and $n_a \in R^3$ and $n_g \in R^3$ are sensor noises. The vector \tilde{g} is the local gravitational acceleration vector. It is assumed that \tilde{g} is known, which can be computed using the formula in [28]. The sensor biases are assumed to be estimated separately using calibration algorithms [29,30].

Let T denote the sampling period of a sensor. For a continuous time signal $y(t)$, the discrete value is denoted by $y_k = y(kT)$. The discrete sensor noise $n_{a,k}$ and $n_{g,k}$ are assumed to be white Gaussian sensor noises, whose covariances are given by

$$\begin{aligned} R_a &= E \left\{ n_{a,k} n_{a,k}^T \right\} = r_a I_3 \in R^{3 \times 3}, \\ R_g &= E \left\{ n_{g,k} n_{g,k}^T \right\} = r_g I_3 \in R^{3 \times 3}. \end{aligned} \quad (2)$$

where $r_a > 0$ and $r_g > 0$ are scalar constants.

We assume that a walking trajectory consists of straight line paths, where the angle between two adjacent paths can only take one of the following angle $\{90^\circ, -90^\circ, 180^\circ\}$. An example of a walking trajectory is given in Figure 1, where there is 90° turn at P_2 and -90° turn at P_3 . Waypoints are denoted by $P_m \in R^3$ ($1 \leq m \leq M$), which include positions with turn events (P_2 and P_3 in Figure 1), the initial position (P_1) and the final position (P_4).

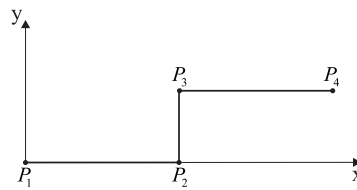


Figure 1. Example of waypoints ($P_1 \sim P_4$).

2.1. Standard Inertial Navigation Using an Indirect Kalman Filter

In this subsection, q , r , and v are estimated using a standard inertial navigation algorithm with an indirect Kalman filter [31].

The basic equations for inertial navigation are given as follows:

$$\begin{aligned} \dot{q} &= \frac{1}{2} \Omega(\omega) q, \\ \dot{v}_n &= a_n = C^T(q) a_b, \\ \dot{r}_n &= v_n, \end{aligned} \quad (3)$$

where symbol Ω is defined by

$$\Omega(\omega) = \begin{bmatrix} 0 & \omega^T \\ \omega & -[\omega \times] \end{bmatrix},$$

and $[\omega \times] \in R^{3 \times 3}$ denotes the skew symmetric matrix of ω :

$$[\omega \times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}.$$

Let \hat{q}_k , \hat{r}_k and \hat{v}_k be the estimated values of q , r and v using (3), where ω and a_b are replaced by y_g and $y_a - C(\hat{q})\tilde{g}$:

$$\begin{aligned} \dot{\hat{q}} &= \frac{1}{2} \Omega(y_g) \hat{q}, \\ \dot{\hat{v}} &= C^T(\hat{q}) y_a - \tilde{g}, \\ \dot{\hat{r}} &= \hat{v}. \end{aligned} \quad (4)$$

Coriolis and Earth curvature effects are ignored since we use a consumer grade IMU for limited walking distances where these effects are below the IMU noise level [32].

When the initial attitude (\hat{q}_0) is computed, the pitch and roll angles can be computed from $y_{a,0}$. However, the yaw angle is not determined since there is no heading reference sensor such as a magnetic

sensor. In the proposed algorithm, the initial yaw angle can be arbitrarily chosen since the yaw angle is automatically adjusted later (see Section 2.3).

\bar{q}_e , r_e and v_e denote the estimation errors in \hat{q} , \hat{r} , and \hat{v} , which are defined by

$$\begin{aligned}\bar{q}_e &\triangleq q \otimes \hat{q}_k^*, \\ r_e &\triangleq r_n - \hat{r}_k, \\ v_e &\triangleq v_n - \hat{v}_k,\end{aligned}\quad (5)$$

where \otimes is the quaternion multiplication and q^* denotes the quaternion conjugate of a quaternion q . Assuming that \bar{q}_e is small, we can approximate \bar{q}_e as follows:

$$\bar{q}_e \approx \begin{bmatrix} 1 \\ q_e \end{bmatrix}. \quad (6)$$

The state of an indirect Kalman filter is defined by

$$x \triangleq \begin{bmatrix} q_e \\ r_e \\ v_e \end{bmatrix} \in R^{9 \times 1}. \quad (7)$$

The state equation for the Kalman filter is given by [31]:

$$\dot{x} = \begin{bmatrix} [-y_g \times] & 0 & 0 \\ 0 & 0 & I_3 \\ -2C(\hat{q})^T [y_a \times] & 0 & 0 \end{bmatrix} x + w \quad (8)$$

where the covariance of the process noise w is given by

$$E\{ww^T\} = \begin{bmatrix} 0.25R_g & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & C(\hat{q})^T R_a C(\hat{q}) \end{bmatrix}.$$

The discretized version of (8) is used in the Kalman filter as in [31].

Two measurement equations are used in the Kalman filter: the zero velocity updating (ZUPT) equation and the map matching equation.

The zero velocity updating uses the fact that there is almost periodic zero velocity intervals (when a foot is on the ground) during walking. If the following conditions are satisfied, the discrete time index k is assumed to belong to zero velocity intervals [31]:

$$\begin{aligned}\|y_{g,i}\| &\leq B_g, & k - \frac{N_g}{2} \leq i \leq k + \frac{N_g}{2} \\ \|y_{a,i} - y_{a,i-1}\| &\leq B_a, & k - \frac{N_a}{2} \leq i \leq k + \frac{N_a}{2}\end{aligned}\quad (9)$$

where B_g , B_a , N_g and N_a are parameters for zero velocity interval detection. This zero velocity updating algorithm is not valid when a person is on a moving transportation such as an elevator. Thus, the proposed algorithm cannot be used when a person is on a moving transportation.

During the zero velocity intervals, the following measurement equation is used:

$$0_{3 \times 1} - \hat{v}_k = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & I_3 \end{bmatrix} x_k + n_v, \quad (10)$$

where n_v is a fictitious measurement noise representing a Gaussian white noise with the noise covariance R_v .

The second measurement equation is from the map matching, which is used during the zero velocity intervals. From the assumption, a straight line path is parallel to either x axis or y axis. If a path is parallel to $x(y)$ axis, $y(x)$ position is constant and this can be used in the measurement equation.

Let $e_i \in R^{3 \times 1}$ ($1 \leq i \leq 3$) be the unit vector whose i -th element is 1 and the remaining elements are 0. Suppose a person is on the path $[P_m, P_{m+1}]$ and

$$e_j^T (P_{m+1} - P_m) = 0, \quad j = 1 \text{ or } 2. \quad (11)$$

For example, (11) is satisfied with $j = 1$ if the path is parallel to the y axis.

When (11) is satisfied with j ($j = 1$ or 2), then the map matching measurement equation is given by

$$e_j^T (P_m - \hat{r}_k) = \begin{bmatrix} 0_{1 \times 3} & e_j^T & 0_{1 \times 3} \end{bmatrix} x_k + n_{r,12}, \quad (12)$$

where $n_{r,12}$ is the horizontal position measurement noise whose noise covariance is $R_{r,12}$.

If the path is level (that is, $e_3^T (P_{m+1} - P_m) = 0$), the z axis value of r_k is almost the same in the zero velocity intervals when the foot is on the ground. In this case, the following z axis measurement equation is also used:

$$e_3^T (P_m - \hat{r}_k) = \begin{bmatrix} 0_{1 \times 3} & e_3^T & 0_{1 \times 3} \end{bmatrix} x_k + n_{r,3}, \quad (13)$$

where $n_{r,3}$ is the vertical position measurement noise whose noise covariance is $R_{r,3}$.

The proposed filtering algorithm is illustrated in Figure 2. To further reduce the estimation errors of the Kalman filter, the smoothing algorithm in [31] is applied to the Kalman filter estimated values.

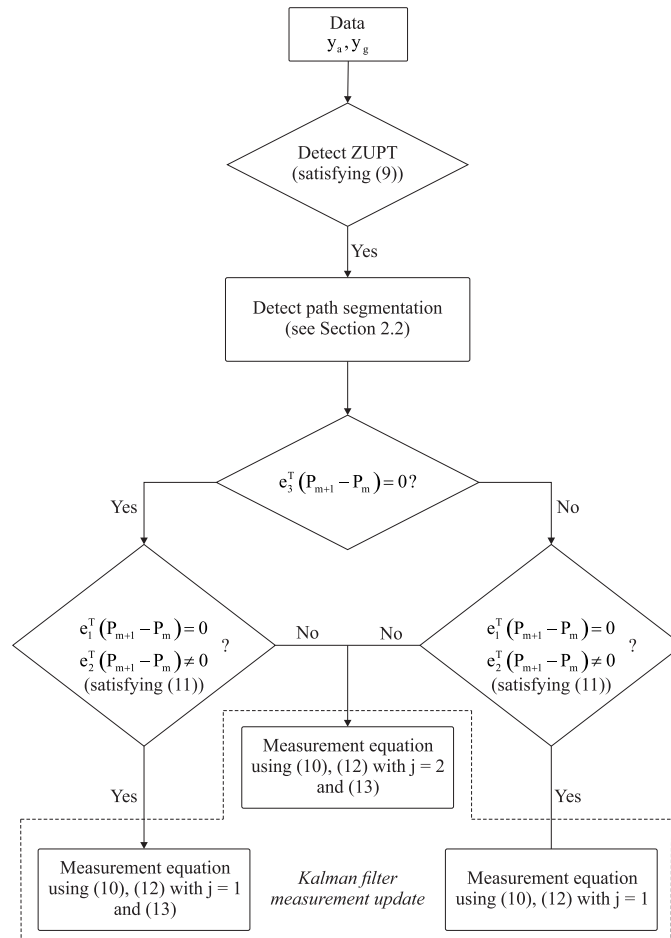


Figure 2. Proposed measurement update flowchart.

2.2. Path Identification

To apply map matching measurement Equation (12), a current path must be identified. Using the zero velocity intervals, each walking step can be easily determined. Let S_l be the discrete time index of the end of l -th zero velocity interval (see Figure 3).

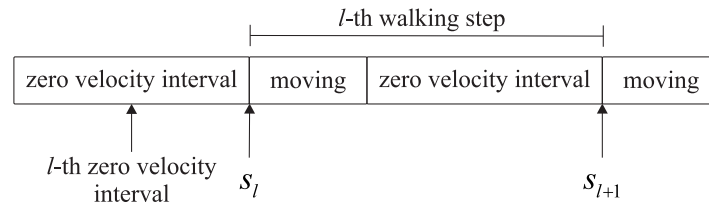


Figure 3. Walking step segmentation.

Let ψ_{S_l} be the yaw angle at the discrete time S_l and $\Delta\psi_l$ be defined by

$$\Delta\psi_l \triangleq \psi_{S_{l+1}} - \psi_{S_l}. \quad (14)$$

Please note that $\Delta\psi_l$ denotes the yaw angle change during l -th walking step.

The turning (that is, the change of a path) is detected using $\Delta\psi_l$. Suppose a current path is $[P_m, P_{m+1}]$. Then the next turning angle is determined by two vectors $P_{m+1} - P_m$ and $P_{m+2} - P_{m+1}$. For example in Figure 1, if a current path is $[P_1, P_2]$, the next turning angle is 90° . Let δ be the next turning angle. Turning is determined to occur at l -th walking step if the following condition is satisfied:

$$\left| \sum_{j=l-M_l}^{l+M_l} \Delta\psi_j - \delta \right| < \psi_{th}, \quad (15)$$

where ψ_{th} is a threshold parameter and M_l is a positive integer parameter. Equation (15) detects the turning if the summation of the yaw angle changes during $2M_l + 1$ walking steps is similar to the next expected turning angle δ . The use of parameter M_l is due to the fact that turning does not occur during a single step. Once turning is detected in (15), a current path is modified to $[P_{m+1}, P_{m+2}]$ from $[P_m, P_{m+1}]$.

2.3. Initial Yaw Angle Adjustment

The initial yaw angle is arbitrarily chosen and adjusted as follows. Suppose the first N_{first} walking steps belong to the first path $[P_1, P_2]$. Let x_l and y_l ($1 \leq l \leq N_l$) be defined by

$$\begin{bmatrix} x_l \\ y_l \end{bmatrix} = \begin{bmatrix} e_1^T \\ e_2^T \end{bmatrix} \hat{r}_{S_l}, \quad 1 \leq l \leq N_l. \quad (16)$$

The line equation representing (x_l, y_l) is computed using the following least squares optimization:

$$\min_{a,b,c} \sum_{l=1}^{N_l} \|ax_l + by_l + c\|_2^2. \quad (17)$$

The angle between the line (defined by the line equation a, b, c) and $[e_1 \ e_2]^T (P_2 - P_1)$ is computed. Using this angle, the initial yaw angle is adjusted so that the walking direction coincides with the direction dictated by the waypoints.

3. Spline Function Computation

In this section, spline function $\bar{q}(t)$ (quaternion spline) and $\bar{r}(t)$ (position spline) are computed using \hat{q}_k , \hat{r}_k and \hat{v}_k as control points.

3.1. Cumulative B-Splines Quaternion Curve

Since the quaternion spline $\bar{q}(t)$ is computed for each interval $[kT, (k+1)T)$, it is convenient to introduce $\bar{q}_k(u)$ notation, which is defined by

$$\bar{q}(t) = \bar{q}_k(u) \triangleq \bar{q}(kT + u), \quad t \in [kT, (k+1)T), \quad 0 \leq u < T. \quad (18)$$

To define $\bar{q}_k(u)$, $\hat{\omega}_k$ is defined as follows:

$$\hat{\omega}_k \triangleq \frac{1}{T} \log(C(\hat{q}_{k+1})C^T(\hat{q}_k)), \quad (19)$$

where the logarithm of an orthogonal matrix is defined in [33]. Please note that $\hat{\omega}_k$ is a constant angular velocity vector, which transform from \hat{q}_k to \hat{q}_{k+1} .

Cumulative B-spline basis function is used to represent the quaternion spline, which is first proposed in [24] and also used in [34].

$$\bar{q}_k(u) = \left(\prod_{i=1}^3 S_i(u) \right) \hat{q}_{k-1} \quad (20)$$

where

$$S_i(u) \triangleq \exp\left(\frac{1}{2} \tilde{B}_i(u) \Omega(\hat{\omega}_{k-2+i})\right).$$

The cumulative basis function $\tilde{B}_i(u) \in R^3$ ($1 \leq i \leq 3$) is defined by

$$\begin{bmatrix} \tilde{B}_1(u) \\ \tilde{B}_2(u) \\ \tilde{B}_3(u) \end{bmatrix} = D \begin{bmatrix} 1 \\ \frac{u}{T} \\ \frac{u^2}{T^2} \\ \frac{u^3}{T^3} \end{bmatrix} \quad (21)$$

where D is computed using the matrix representation of the De Boor–Cox formula [34]:

$$D = \frac{1}{6} \begin{bmatrix} 5 & 3 & -3 & 1 \\ 1 & 3 & 3 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

To generate the gyroscope simulation data, the angular velocity spline $\bar{\omega}(t)$ is required. From the relationship between $\bar{q}(t)$ and $\bar{\omega}(t)$ [35], $\bar{\omega}(t)$ is given by

$$\bar{\omega}(t) = 2\Xi(\bar{q}(t))\dot{\bar{q}}(t) \quad (22)$$

where

$$\Xi(q) \triangleq \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}.$$

The derivative of $\bar{q}(t)$ ($kT \leq t < (k+1)T$) is given by

$$\dot{\bar{q}}(t) = \frac{d\bar{q}_k(u)}{du} = (\dot{S}_3 S_2 S_1 + S_3 \dot{S}_2 S_1 + S_3 S_2 \dot{S}_1) \hat{q}_{k-1} \quad (23)$$

where

$$\dot{S}_i = \frac{dS_i(u)}{du} = \frac{1}{2}\dot{\tilde{B}}_i(u)\Omega(\hat{\omega}_{k-2+i})S_i$$

$$\dot{\tilde{B}}(u) = \begin{bmatrix} \dot{\tilde{B}}_1(u) \\ \dot{\tilde{B}}_2(u) \\ \dot{\tilde{B}}_3(u) \end{bmatrix} = D \begin{bmatrix} 0 \\ \frac{1}{T} \\ \frac{2u}{T^2} \\ \frac{3u^2}{T^3} \end{bmatrix}.$$

3.2. Eighth-Order Algebraic Splines

The position $\bar{r}(t)$ is represented by eighth-order algebraic spline [26], where k -th spline segment is given by

$$\bar{r}(t) = \bar{r}_{k,m}(u) = \sum_{j=0}^7 p_{kj,m} u^j, \quad t \in [kT, (k+1)T). \quad (24)$$

where $p_{kj,m}$ ($0 \leq k \leq N, 0 \leq j \leq 7, 1 \leq m \leq 3$) are the coefficients of the k -th spline segment of the m -th element of \bar{r}_k (for example, $\bar{r}_{k,1}$ is the x position spline). Continuity constraints up to third derivative of $\bar{r}_{k,m}(u)$ are imposed on the coefficients.

Let \hat{a}_k be the estimated value of the external acceleration in the navigation coordinate frame, which can be computed as follows (see (3))

$$\hat{a}_k = C^T(\hat{q}_k)y_{a,k} - \tilde{g}. \quad (25)$$

The coefficients $p_{kj,m}$ are chosen so that $\bar{r}(kT)$, $\dot{\bar{r}}(kT)$ and $\ddot{\bar{r}}(kT)$ are close to \hat{r}_k , \hat{v}_k and \hat{a}_k . An additional constraint is also imposed to reduce the jerk term (the third derivative of $\bar{r}(t)$), which makes $\bar{r}(t)$ smooth rather than jerky. Thus, the performance index for m -th element ($1 \leq m \leq 3$) of $\bar{r}(t)$ is defined as follows:

$$J_m = \alpha_1 \sum_{k=0}^N (\bar{r}_{k,m}(T) - \hat{r}_{k,m})^2 + \alpha_2 \sum_{k=0}^N (\dot{\bar{r}}_{k,m}(T) - \hat{v}_{k,m})^2 + \alpha_3 \sum_{k=0}^N (\ddot{\bar{r}}_{k,m}(T) - \hat{a}_{k,m})^2 + \beta \sum_{k=0}^N \int_0^T (\ddot{\bar{r}}_{k,m}(t))^2 dt. \quad (26)$$

This minimization of J_m can be formulated as a quadratic minimization problem of the coefficients $p_{kj,m}$ as follows:

$$\min_X J_m = X^T M_1 X + 2M_2 X + M_3 \quad (27)$$

where $M_1 \in R^{8(N-1) \times 8(N-1)}$, $M_2 \in R^{1 \times 8(N-1)}$ and $M_3 \in R$ can be computed from (26).

Although the matrix size of M_1 could be large, it is a banded matrix with a bandwidth of 7. An efficient Cholesky decomposition algorithm is available for matrices with a small bandwidth [36] and (27) can be solved efficiently.

Let $\bar{p}_{kj,m}$ be the minimization solution of (26). Then \bar{r} can be computed by inserting $\bar{p}_{kj,m}$ into (24) and \bar{v} , \bar{a} can be computed by taking the derivatives.

The weighting factors α_i ($1 \leq i \leq 3$) and β determine the smoothness of \bar{r} , \bar{v} and \bar{a} . If β is large, we obtain smoother \bar{r} , \bar{v} and \bar{a} curves at the sacrifice of nearness to the control points \hat{r} , \hat{v} and \hat{a} .

Let \bar{y}_a and \bar{y}_g be accelerometer and gyroscope output generated using the spline functions. These values can be generated using (1), where q , a_b , ω are replaced by \bar{q} , $C(\bar{q})\bar{a}$, $\bar{\omega}$ with appropriate sensor noises. An overview of the simulation data generation system is shown in Figure 4.

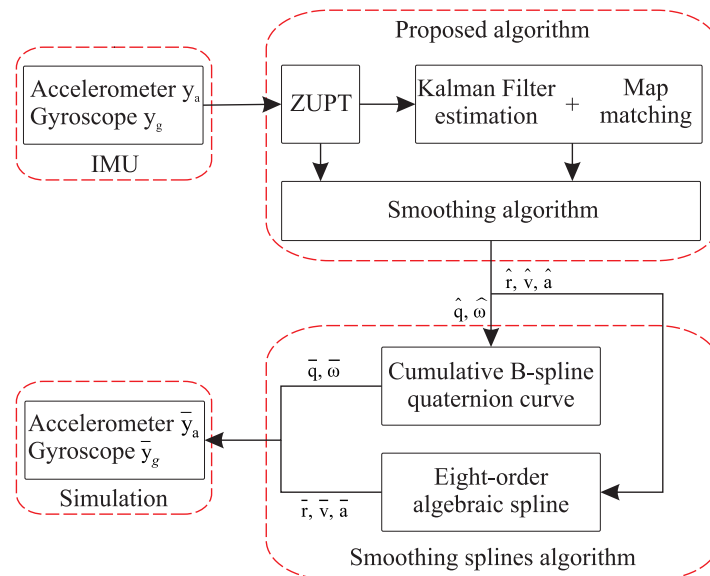


Figure 4. System overview for simulation data generation.

4. Experiment and Results

In this section, the proposed algorithm is implemented through indoor experiments. An inertial measurement unit MTi-1 of XSens was attached on the foot with the sampling frequency of 100 Hz. The parameters used in the proposed algorithm are given in Table 1.

Table 1. Parameters used in the proposed algorithm

Parameters	Values	Related Equations
r_a	0.0015	(2)
r_g	0.00001	
B_g	0.8	(9)
B_a	1.5	
N_g, N_a	30	
n_v	0.01	(10)
$n_{r,12}$	0.0004	(12)
$n_{r,3}$	0.0004	(13)
ψ_{th}	30°	(15)
$\alpha_1, \alpha_2, \alpha_3$	1	(26)
β	0.00001	

Two experiment sets of scenarios are conducted: (1) walking along a rectangular path for 2D walking data generation; and (2) walking on corridors and stairs for 3D walking data generation.

4.1. Walking along a Rectangular Path

In this experimental scenario, the subjects walked five laps at normal speed along a rectangular path with dimensions of 13.05 m by 6.15 m. The total distance for five laps on the path was 192 m.

The standard Kalman filter results without map matching are given in Figure 5. The errors in Figure 5 were primarily driven by the errors in the orientation estimation as well as run bias instability of the inertial sensors leading to the position error of 1 m. As can be seen that the maximum error was almost 2 m at the end of the walking.

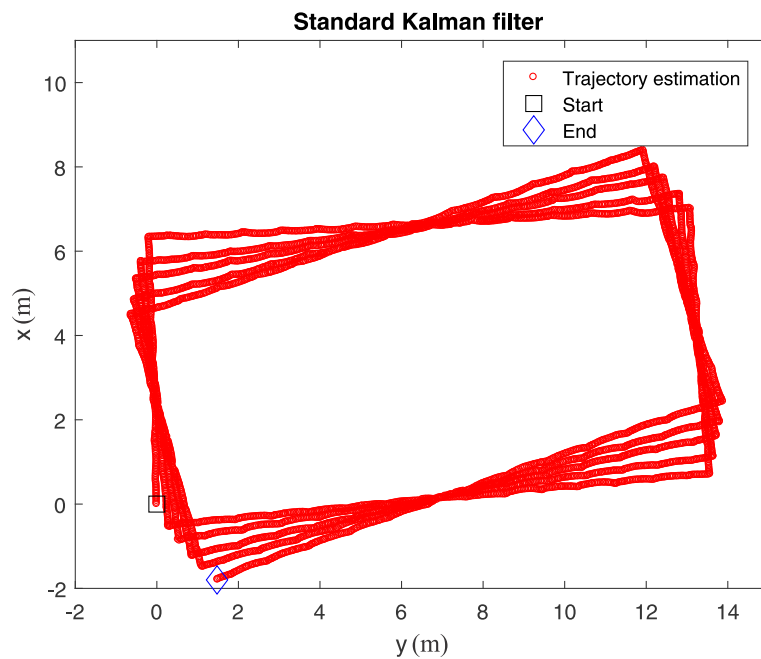


Figure 5. Rectangle walking estimation for five laps using a standard Kalman filter.

The navigation results in Figure 6 were obtained from the proposed smoothing algorithm with map matching. It can be seen that the results are improved due to the map matching.

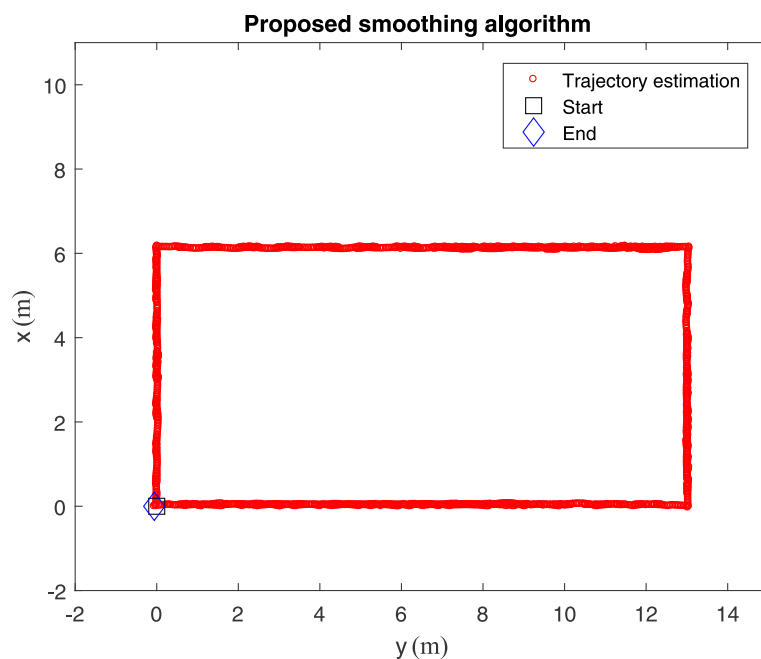


Figure 6. Rectangle walking estimation for five laps using the proposed smoothing algorithm.

The z axis spline data (for the three walking steps out of the rectangle walking steps) is given in Figure 7.

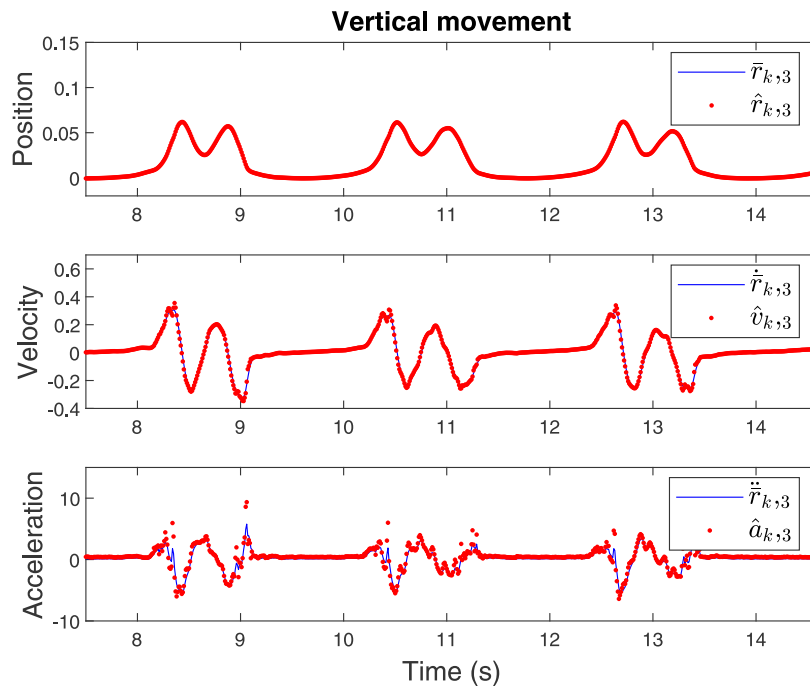


Figure 7. z axis spline functions of the rectangle walking.

4.2. Walking along a 3D Indoor Environment

In the second experimental scenario, a person walked along a 3D indoor environment corridors and stairs.

Figure 8 shows the walking along a 3D trajectory results with a waypoint P_1 is the starting point whose coordinate is (0,0,0). The coordinates of waypoints P_1 to P_{14} are known, and the starting waypoint P_1 coincides with the end waypoint P_{14} on the projection plane. The routes are as follow: starting from the third floor waypoint P_1 , walking along the floor to collect data with normal speed then turning 180° in P_2 and turning on the right in P_3 , then going up the stairs through the floor by the same way as the other floors, finally reaching the fifth floor waypoint P_{14} , after that turning 180° and walking along the floor to coming back the starting waypoint P_1 . The total length of the path line is approximately 562.98 m.

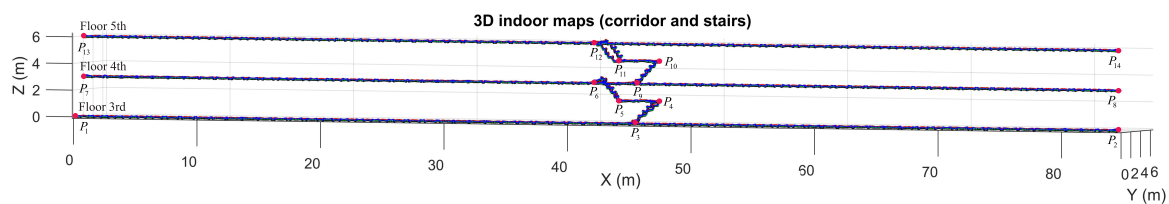


Figure 8. 3D path indoor estimation with the proposed smoothing algorithm.

4.3. Evaluation of Simulation Data Usefulness

The simulation data used to test specific performance of a certain algorithm. In this subsection, the standard Kalman filter without map matching is evaluated using the simulation data generated in Section 4.1. The algorithm is tested 1000 times with different noise values (walking motion data is the same).

Let $\bar{r}_{xy,final,i}$ be the estimation error of the final position value (x and y components) of i -th simulation, where $1 \leq i \leq 1000$. Since the mean value of $\bar{r}_{xy,final,i}$ is not exactly zeros, its mean value is subtracted.

The accuracy of an algorithm can be evaluated from the distribution of $\bar{r}_{xy,final,i}$. Consider an ellipse whose center point is the mean of $\bar{r}_{xy,final,i}$ and which include 50% of $\bar{r}_{xy,final,i}$ (see Figure 9). Let a be the length of the major axis and b be the length of the minor axis. Smaller values of a and b indicate that the error of a certain algorithm is smaller.

4.3.1. Affects of Sampling Rate on the Estimation Performance

The simulation results for the final position estimated errors with different sampling rates are given in Table 2. As an example, Figure 9 shows the estimation error of the final position value with a sampling rate of 100 Hz, where an ellipse represents the boundary of 50% of probability. As can be seen, the final position errors of the different sampling rates are almost the same except for the result with a sampling rate of 50 Hz. This simulation result suggests that the sampling rate is smaller than 100 Hz is not desirable. Also, there is no apparent benefit using the sampling rate higher than 100 Hz.

Table 2. Final position errors with different sampling rates.

Sampling Rate	Mean of $\bar{r}_{xy,final}$ (m)		Radius (m)	
	\bar{x}_{final}	\bar{y}_{final}	a	b
50 Hz	−1.5217	−0.5428	0.7605	0.0215
100 Hz	0.6639	0.4745	0.0554	0.0034
150 Hz	0.6955	0.4558	0.0594	0.0032
200 Hz	0.7156	0.4520	0.0582	0.0042

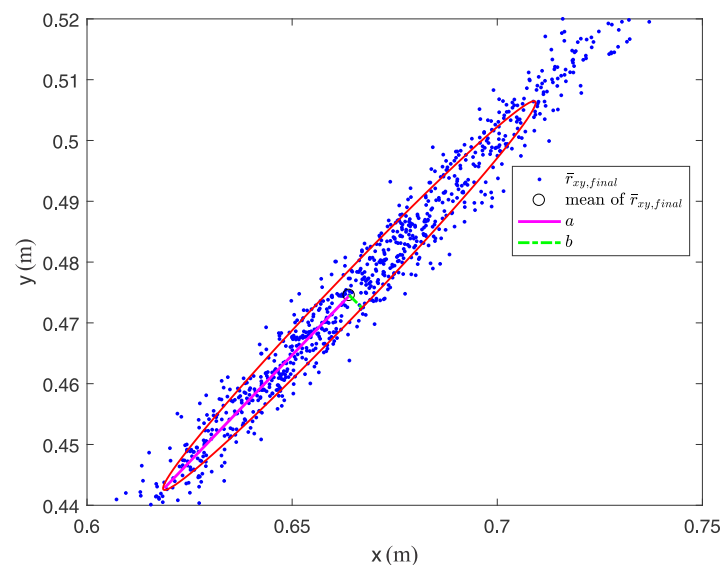


Figure 9. Estimation error of the final position value with a sampling rate 100 Hz [ellipse with 50% of probability radius].

4.3.2. Gyroscope Bias Effect

The simulation results for the final position estimated errors with different gyroscope bias are given in Table 3. As can be seen, the final position errors change significantly when the gyroscope bias becomes to change larger.

Table 3. Final position errors with different gyroscope bias.

r_g	Mean of $\bar{r}_{xy,final}$ (m)		Radius (m)	
	\bar{x}_{final}	\bar{y}_{final}	a	b
0	0.7023	0.6122	0.133	0.0226
0.00001	0.6568	0.4696	0.0595	0.0035
0.0001	−0.2038	−0.0280	0.5952	0.0309
0.001	−6.3768	0.5568	3.5123	1.0306

5. Conclusions

In this paper, simulation data is generated for walking motion estimation using inertial sensors. The first step to generate simulation data is to perform a specific motion, which you want to estimate. The attitude and position are computed from inertial sensors data using a smoothing algorithm. The spline functions are generated using the smoother estimated values as control points. B-spline function is used for the attitude quaternion, and eighth-order algebraic spline function is used for the position.

The spline simulation data generated from the proposed algorithm can be used to test a new motion estimation algorithm by easily changing the parameters such as noise covariance, sampling rate and bias terms.

The shape of generated spline function depends on the weighting factors, which control the constraint on the jerk term. It is one of future research topic to investigate how to choose the optimal weighting factors, which give the most realistic simulation data.

Author Contributions: T.T.P. and Y.S.S. conceived and designed this study. T.T.P. performed the experiments and wrote the paper. Y.S.S. reviewed and edited the manuscript.

Funding: This work was supported by the 2018 Research Fund of University of Ulsan.

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

References

1. Zhou, H.; Hu, H. Human motion tracking for rehabilitation—A survey. *Biomed. Signal Process. Control* **2008**, *3*, 1–18.10.1016/j.bspc.2007.09.001. [\[CrossRef\]](#)
2. Tao, Y.; Hu, H.; Zhou, H. Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation. *Int. J. Robot. Res.* **2007**, *26*, 607–624.10.1177/0278364907079278. [\[CrossRef\]](#)
3. Raiff, B.R.; Karataş, Ç.; McClure, E.A.; Pompili, D.; Walls, T.A. Laboratory validation of inertial body sensors to detect cigarette smoking arm movements. *Electronics* **2014**, *3*, 87–110.10.3390/electronics3010087. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Fang, T.H.; Park, S.H.; Seo, K.; Park, S.G. Attitude determination algorithm using state estimation including lever arms between center of gravity and IMU. *Int. J. Control Autom. Syst.* **2016**, *14*, 1511–1519.10.1007/s12555-015-0251-4. [\[CrossRef\]](#)
5. Ahmed, H.; Tahir, M. Improving the accuracy of human body orientation estimation with wearable IMU sensors. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 535–542.10.1109/TIM.2016.2642658. [\[CrossRef\]](#)
6. Suh, Y.S.; Phuong, N.H.Q.; Kang, H.J. Distance estimation using inertial sensor and vision. *Int. J. Control Autom. Syst.* **2013**, *11*, 211–215.10.1007/s12555-011-9205-7. [\[CrossRef\]](#)
7. Erdem, A.T.; Ercan, A.Ö. Fusing inertial sensor data in an extended Kalman filter for 3D camera tracking. *IEEE Trans. Image Process.* **2015**, *24*, 538–548.10.1109/TIP.2014.2380176. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Zhang, Z.; Meng, X. Use of an inertial/magnetic sensor module for pedestrian tracking during normal walking. *IEEE Trans. Instrum. Meas.* **2015**, *64*, 776–783.10.1109/TIM.2014.2349211. [\[CrossRef\]](#)
9. Ascher, C.; Kessler, C.; Maier, A.; Crocoll, P.; Trommer, G. New pedestrian trajectory simulator to study innovative yaw angle constraints. In Proceedings of the 23rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2010), Portland, OR, USA, 21–24 September 2010; pp. 504–510.

10. Jiménez, A.R.; Seco, F.; Prieto, J.C.; Guevara, J. Indoor pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU. In Proceedings of the 7th Workshop on Positioning, Navigation and Communication, Dresden, Germany, 11–12 March 2010; pp. 135–143.10.1109/WPNC.2010.5649300. [\[CrossRef\]](#)
11. Fourati, H. Heterogeneous data fusion algorithm for pedestrian navigation via foot-mounted inertial measurement unit and complementary filter. *IEEE Trans. Instrum. Meas.* **2015**, *64*, 221–229.10.1109/TIM.2014.2335912. [\[CrossRef\]](#)
12. Rodger, J.A. Toward reducing failure risk in an integrated vehicle health maintenance system: A fuzzy multi-sensor data fusion Kalman filter approach for IVHMS. *Expert Syst. Appl.* **2012**, *39*, 9821–9836. [\[CrossRef\]](#)
13. He, C.; Kazanzides, P.; Sen, H.T.; Kim, S.; Liu, Y. An Inertial and Optical Sensor Fusion Approach for Six Degree-of-Freedom Pose Estimation. *Sensors* **2015**, *15*, 16448–16465.10.3390/s150716448. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Kim, A.; Golnaraghi, M.F. Initial calibration of an inertial measurement unit using an optical position tracking system. In Proceedings of the PLANS 2004. Position Location and Navigation Symposium (IEEE Cat. No.04CH37556), Monterey, CA, USA, 26–29 April 2004; pp. 96–101.10.1109/PLANS.2004.1308980. [\[CrossRef\]](#)
15. Enayati, N.; Momi, E.D.; Ferrigno, G. A quaternion-based unscented Kalman filter for robust optical/inertial motion tracking in computer-assisted surgery. *IEEE Trans. Instrum. Meas.* **2015**, *64*, 2291–2301.10.1109/TIM.2015.2390832. [\[CrossRef\]](#)
16. Karlsson, P.; Lo, B.; Yang, G.Z. Inertial sensing simulations using modified motion capture data. In Proceedings of the 11th International Conference on Wearable and Implantable Body Sensor Networks (BSN 2014), ETH Zurich, Switzerland, 16–19 June 2014.
17. Young, A.D.; Ling, M.J.; Arvind, D.K. IMUSim: A simulation environment for inertial sensing algorithm design and evaluation. In Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks, Chicago, IL, USA, 12–14 April 2011; pp. 199–210.
18. Ligorio, G.; Sabatini, A.M. A simulation environment for benchmarking sensor fusion-based pose estimators. *Sensors* **2015**, *15*, 32031–32044.10.3390/s151229903. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Zampella, F.J.; Jiménez, A.R.; Seco, F.; Prieto, J.C.; Guevara, J.I. Simulation of foot-mounted IMU signals for the evaluation of PDR algorithms. In Proceedings of the 2011 International Conference on Indoor Positioning and Indoor Navigation, Guimarães, Portugal, 21–23 September 2011; pp. 1–7.10.1109/IPIN.2011.6071930. [\[CrossRef\]](#)
20. Parés, M.; Rosales, J.; Colomina, I. *Yet Another IMU Simulator: Validation and Applications*; EuroCow: Castelldefels, Spain, 2008; Volume 30.
21. Zhang, W.; Ghogho, M.; Yuan, B. Mathematical model and matlab simulation of strapdown inertial navigation system. *Model. Simul. Eng.* **2012**, *2012*, 264537.10.1155/2012/264537. [\[CrossRef\]](#)
22. Parés, M.E.; Navarro, J.A.; Colomina, I. On the generation of realistic simulated inertial measurements. In Proceedings of the 2015 DGON Inertial Sensors and Systems Symposium (ISS), Karlsruhe, Germany, 22–23 September 2015; pp. 1–15.10.1109/InertialSensors.2015.7314268. [\[CrossRef\]](#)
23. Schumaker, L. *Spline Functions: Basic Theory*, 3 ed.; Cambridge Mathematical Library, Cambridge University Press: Cambridge, UK, 2007; doi:10.1017/CBO9780511618994.
24. Kim, M.J.; Kim, M.S.; Shin, S.Y. A general construction scheme for unit quaternion curves with simple high order derivatives. In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 6–11 August 1995; ACM: New York, NY, USA, 1995; pp. 369–376.10.1145/218380.218486. [\[CrossRef\]](#)
25. Sommer, H.; Forbes, J.R.; Siegwart, R.; Furgale, P. Continuous-time estimation of attitude using B-Splines on Lie groups. *J. Guid. Control Dyn.* **2016**, *39*, 242–261.10.2514/1.G001149. [\[CrossRef\]](#)
26. Simon, D. Data smoothing and interpolation using eighth-order algebraic splines. *IEEE Trans. Signal Process.* **2004**, *52*, 1136–1144.10.1109/TSP.2004.823489. [\[CrossRef\]](#)
27. Pham, T.T.; Suh, Y.S. Spline function simulation data generation for inertial sensor-based motion estimation. In Proceedings of the 2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Nara, Japan, 11–14 September 2018; pp. 1231–1234.
28. Pavlis, N.K.; Holmes, S.A.; Kenyon, S.C.; Factor, J.K. The development and evaluation of the Earth Gravitational Model 2008 (EGM2008). *J. Geophys. Res. Solid Earth* **2012**, *117*. [\[CrossRef\]](#)

29. Metni, N.; Pflimlin, J.M.; Hamel, T.; Souères, P. Attitude and gyro bias estimation for a VTOL UAV. *Control Eng. Pract.* **2006**, *14*, 1511–1520.10.1016/j.conengprac.2006.02.015. [[CrossRef](#)]
30. Hwangbo, M.; Kanade, T. Factorization-based calibration method for MEMS inertial measurement unit. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 1306–1311.10.1109/ROBOT.2008.4543384. [[CrossRef](#)]
31. Suh, Y.S. Inertial sensor-based smoother for gait analysis. *Sensors* **2014**, *14*, 24338–24357.10.3390/s141224338. [[CrossRef](#)]
32. Placer, M.; Kovačič, S. Enhancing indoor inertial pedestrian navigation using a shoe-worn marker. *Sensors* **2013**, *13*, 9836–9859. [[CrossRef](#)]
33. Gallier, J.; Xu, D. Computing exponential of skew-symmetric matrices and logarithms of orthogonal matrices. *Int. J. Robot. Autom.* **2002**, *17*, 1–11.
34. Patron-Perez, A.; Lovegrove, S.; Sibley, G. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *Int. J. Comput. Vis.* **2015**, *113*, 208–219. [[CrossRef](#)]
35. Markley, F.L.; Crassidis, J.L. *Fundamentals of Spacecraft Attitude Determination and Control*; Springer: New York, NY, USA, 2014.
36. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; Johns Hopkins University Press: Baltimore, MD, USA, 1983.



© 2018 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/>).