

Article

Inertial Sensor-Based Smoother for Gait Analysis

Young Soo Suh

Department of Electrical Engineering, University of Ulsan, Mugeo, Namgu, Ulsan 680-749, Korea;
E-Mail: yssuh@ulsan.ac.kr; Tel.: +82-52-259-2196

External Editor: Gert F. Trommer

Received: 17 October 2014; in revised form: 5 December 2014 / Accepted: 9 December 2014 /

Published: 17 December 2014

Abstract: An off-line smoother algorithm is proposed to estimate foot motion using an inertial sensor unit (three-axis gyroscopes and accelerometers) attached to a shoe. The smoother gives more accurate foot motion estimation than filter-based algorithms by using all of the sensor data instead of using the current sensor data. The algorithm consists of two parts. In the first part, a Kalman filter is used to obtain initial foot motion estimation. In the second part, the error in the initial estimation is compensated using a smoother, where the problem is formulated in the quadratic optimization problem. An efficient solution of the quadratic optimization problem is given using the sparse structure. Through experiments, it is shown that the proposed algorithm can estimate foot motion more accurately than a filter-based algorithm with reasonable computation time. In particular, there is significant improvement in the foot motion estimation when the foot is moving off the floor: the z -axis position error squared sum (total time: 3.47 s) when the foot is in the air is 0.0807 m² (Kalman filter) and 0.0020 m² (the proposed smoother).

Keywords: gait analysis; smoother; inertial sensors; inertial navigation algorithm

1. Introduction

Gait analysis is the study of human walking [1–3]. It is helpful in the medical diagnosis and treatment of those diseases that affect the ability to walk.

There are many devices that can be used to perform gait analysis [4]. Mechanical goniometers are used to measure hip, knee and ankle angles. Force plates on the floor or on a shoe (such as the wearable force plate from Tec Gihan Co., Kyoto, Japan) are used to measure the vertical force beneath the foot [5].

Optical motion trackers (such as Vicon) are used to measure the movement of a leg and a foot. These sensor systems can be used independently or together to give necessary gait parameters, such as stride length, walking speed and ground reaction forces.

Probably, an optical motion tracker is the most popular gait analysis tool [6], since it provides a very accurate description of foot and leg motion. The disadvantages of an optical motion tracker are its high cost and limited walking range. You can only increased walking range by using additional cameras, which can be quite expensive for a long walking range.

An alternative technology for the optical motion tracker is the use of inertial sensors [7,8]. Although the accuracy of inertial sensor-based estimation is generally worse than that of the optical motion tracker, the inertial sensor approach does not have a walking range limitation, and the cost is generally cheaper. If inertial sensors (gyroscopes and accelerometers) are attached to a leg, the angles of leg joints can be estimated (see [9–11] and commercial products from INSENCO and Xsens). These angle-based systems are extended to estimate whole human body motion [12–14]. Theoretically, the position of a leg also can be estimated using the inertial navigation algorithm [15]. However, the position estimation error quickly diverges when low cost inertial sensors are used. If inertial sensors are attached to a foot, the position (in addition to angle estimation) can be estimated using an inertial navigation algorithm. Unlike in the leg case, we know that the foot touches the ground almost periodically during walking. When the foot touches the ground, the velocity of the foot becomes temporarily zero, and during this zero velocity interval, the velocity errors of the inertial navigation algorithm can be reset. This zero velocity updating significantly reduces the position error [16].

Estimation of foot movement by attaching inertial sensors to the foot has been extensively studied in the context of a pedestrian navigation system [17,18]. In the filter-based foot motion estimation, the position error increases as soon as the foot is off the ground, since there are no measurements when the foot is in the air. The position error decreases once the foot touches the ground from the zero velocity updating. Thus, in the filter-based foot motion estimation, the motion estimation is not accurate when the foot is moving, but the final position (after the zero velocity updating) becomes accurate. In the pedestrian navigation system, the final position of the foot (that is, the final position of a person) is the main concern and, thus, the filter-based estimation can be used.

On the other hand, foot motion trajectory is the main concern in gait analysis. The filter estimated position can be improved using the smoother. The filtering algorithm uses the measurement data up to the current time to compute the current estimation. On the other hand, the smoother uses the whole measurement data (including the future data) to compute the current estimation. In [19], a smoother algorithm is proposed based on the forward-backward filter, where a smoothing algorithm is applied for each walking step. The position after the zero velocity updating is used as the initial value of a backward filter. The smoothed result for each step is combined rather heuristically to produce total walking estimation.

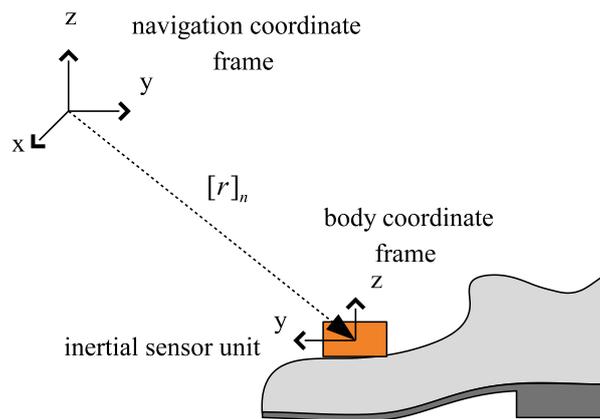
In this paper, the smoothing problem is formulated as a global optimization problem, where the whole walking motion is estimated in a single optimization problem. The smoother algorithm is formulated as a quadratic optimization problem (motivated by [20]), where its problem size depends on the total walking time. Since the problem size becomes very large even for a few minutes of walking data, the efficient solution is given utilizing the sparse structure of the problem.

The remainder of the paper is organized as follows. In Section 2, a basic material is introduced defining the coordinate systems and the main variables. In Section 3, a Kalman filter to estimate the foot motion is given. In Section 4, a smoother algorithm to estimate the foot motion is formulated in the quadratic optimization problem. In Section 5, a solution to the quadratic optimization problem is given using the sparse structure of the problem. In Section 6, experiment results are given to verify the proposed algorithm. Concluding remarks are given in Section 7.

2. Problem Formulation

Two-coordinate frames (the body coordinate frame and the navigation coordinate frame) are defined as in Figure 1. The three axes of the body coordinate frame coincide with the three axes of the inertial sensor unit. The z -axis of the navigation coordinate frame coincides with the local gravitational direction. The choice of the x -axis of the navigation coordinate frame can be chosen arbitrarily, since the walking direction is not important in the gait analysis. The notation $[p]_n$ ($[p]_b$) is used to denote that a vector p is represented in the navigation (body) coordinate frame.

Figure 1. Overview of the foot inertial sensor unit.



The position of the foot 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 of the foot is denoted by $[v]_n \in R^3$. The attitude of the foot 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 basic dynamic equations [21,22] of q , r and v are given by:

$$\begin{aligned} \dot{q} &= \frac{1}{2}\Omega(\omega)q \\ \dot{v} &= C'(q)a_b \\ \dot{r} &= v \end{aligned} \quad (1)$$

where $\omega = [\omega_x \ \omega_y \ \omega_z]'$ is the angular velocity of the body coordinate frame with respect to the navigation coordinate frame. The symbol $\Omega(\omega)$ is defined by:

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

The external acceleration (acceleration related to the movement, excluding the gravitational acceleration) expressed in the body coordinate frame is denoted by $a_b \in R^3$.

The goal of this paper is to estimate q , r and v . The estimated values are denoted by \hat{q} , \hat{r} and \hat{v} , respectively. The inertial navigation algorithm [19,23] is used to estimate q , r and v , where the equations in Equation (1) are integrated to compute \hat{q} , \hat{r} and \hat{v} . Accelerometer output $y_a \in R^3$ and gyroscope output $y_g \in R^3$ are related to ω and a_b as follows:

$$\begin{aligned} y_a &= C(q)\tilde{g} + a_b + b_a + n_a \\ y_g &= \omega + b_g + n_g \end{aligned} \quad (2)$$

where $n_a \in R^3$ and $n_g \in R^3$ are sensor noises. The vector \tilde{g} is the local gravitational acceleration, where $\tilde{g} = [0 \ 0 \ 9.8]'$ is used. The sensor biases are denoted by $b_a \in R^3$ and $b_g \in R^3$. It is assumed that the sensor biases are constant and unknown.

Let T be the sampling period of the sensor output. The sampled outputs of y_a and y_g are denoted by $y_{a,k} = y_a(kT)$ and $y_{g,k} = y_g(kT)$. The same notation is used for q_k , r_k and v_k . Let N be the final discrete time index of the sensor output, where the starting index is one. The discrete sensor noise $n_{a,k}$ and $n_{g,k}$ are assumed to be zero mean white Gaussian noises, whose covariances are given by:

$$R_a = E\{n_{a,k}n'_{a,k}\} \in R^{3 \times 3}, \quad R_g = E\{n_{g,k}n'_{g,k}\} \in R^{3 \times 3}. \quad (3)$$

Due to high sensor noise level of low cost inertial sensors and unknown biases, an inertial navigation algorithm without external reference measurement can estimate foot motion only for a few seconds. To overcome this problem, a zero velocity updating method is used in an inertial sensor-based foot motion estimation. During walking, the foot touches the ground almost periodically. When the foot is on the ground, we know that the velocity of the foot is zero. Thus, whenever the foot touches the ground, the velocity error in the inertial navigation algorithm can be reset, which significantly reduces foot motion estimation errors. To use this zero velocity updating, zero velocity intervals (that is, when the foot is on the ground) must be detected. Many zero velocity detection algorithms have been proposed [16,24].

In this paper, we use a simple zero velocity detection algorithm. Let Z_m be a set of all discrete time indices belonging to the zero velocity intervals. The discrete time k belongs to Z_m if the following conditions are satisfied:

$$\begin{aligned} \|y_{g,i}\| &\leq B_g, & k - \frac{N_g}{2} \leq i \leq k + \frac{N_g}{2} \\ \|y_{a,i} - y_{a,i-1}\| &\leq B_a, & k - \frac{N_a}{2} \leq i \leq k + \frac{N_a}{2} \end{aligned} \quad (4)$$

where N_g and N_a are even number integers.

3. Kalman Filter

In this section, q , r , v , b_g and b_a are estimated using a Kalman filter. The estimated values are denoted by \hat{q}_{KF} , \hat{r}_{KF} , \hat{v}_{KF} , $\hat{b}_{g,KF}$ and $\hat{b}_{a,KF}$. As in [19,23], $q_k \sim b_{a,k}$ are not directly estimated in the Kalman filter. They are first estimated from the integration equation of (1), and then, the errors in the integrated values are estimated in the Kalman filter. These estimated values will be used in Section 4 to derive a smoothed estimation.

3.1. Initialization

Since only the relative position is important in the gait analysis, the initial position is assumed to be zero: that is, $\hat{r}_{KF,1} = 0_{3 \times 1}$, where “1” denotes the initial discrete time index. Assuming that a person is not moving at the initial time, the initial attitude is computed using the triad algorithm [25,26]. The following two vector relations are used in the triad algorithm:

$$\begin{aligned} [y_a]_b &= C(\hat{q}_{KF,1}) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} &= C(\hat{q}_{KF,1}) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \quad (5)$$

The pitch and roll angles are determined from the first vector relation, and the yaw angle is determined from the second vector relation. Since the heading of walking is not important in the gait analysis, the yaw angle can be chosen arbitrarily.

The initial velocity is chosen to be zero: that is, $\hat{v}_{KF,1} = 0_{3 \times 1}$. If there is no particular information on b_g and b_a , the initial estimates are chosen to be zero ($\hat{b}_{g,KF,1} = \hat{b}_{a,KF,1} = 0_{3 \times 1}$).

3.2. Integration of Estimated Values

Suppose we have computed $\hat{q}_{KF,k}$, $\hat{r}_{KF,k}$, \dots , $\hat{b}_{a,KF,k}$ at the discrete time k . The estimated values at the discrete time $k + 1$ are firstly computed using Equation (1), where ω and a_b are replaced by $y_g - \hat{b}_g$ and $y_a - C(\hat{q})\tilde{g} - \hat{b}_a$:

$$\begin{aligned} \dot{\hat{q}} &= \frac{1}{2}\Omega(y_g - \hat{b}_g)\hat{q} = \frac{1}{2}\Omega(\omega + n_g + b_g - \hat{b}_g)\hat{q} \\ \dot{\hat{v}} &= C(\hat{q})'(y_a - \hat{b}_a) - \tilde{g} = C(\hat{q})'(a_b + n_a + b_a - \hat{b}_a) + (C(\hat{q})'C(q) - I)\tilde{g} \\ \dot{\hat{r}} &= \hat{v}. \end{aligned} \quad (6)$$

For later use, we define a function f_k , which represents a numerical integration of Equation (6) from kT to $(k + 1)T$:

$$\begin{bmatrix} \hat{q}_{KF,k+1} \\ \hat{r}_{KF,k+1} \\ \hat{v}_{KF,k+1} \end{bmatrix} = f_k \left(\begin{bmatrix} \hat{q}_{KF,k} \\ \hat{r}_{KF,k} \\ \hat{v}_{KF,k} \end{bmatrix}, \begin{bmatrix} n_{g,k} \\ n_{a,k} \end{bmatrix}, \begin{bmatrix} b_g - \hat{b}_{g,k} \\ b_a - \hat{b}_{a,k} \end{bmatrix} \right). \quad (7)$$

We assume that the variables not used as arguments of f_k are embedded in f_k : for example, the ω , $C(q)$ and \tilde{g} terms are assumed to be embedded in the function f_k .

3.3. Kalman Filter

Due to noise terms ($n_{g,k}$ and $n_{a,k}$) and bias estimation error ($b_g - \hat{b}_{g,k}$ and $b_a - \hat{b}_{a,k}$), there are errors in the numerically-integrated values of Equation (7). These errors are estimated using a Kalman filter.

Let $X_{KF,k}$ be defined by:

$$X_{KF,k} = \begin{bmatrix} \bar{q}_{KF,k} \\ \bar{r}_{KF,k} \\ \bar{v}_{KF,k} \\ \bar{b}_{g,KF,k} \\ \bar{b}_{a,KF,k} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0_{3 \times 1} & I_3 \end{bmatrix} (\hat{q}_{KF,k}^* \otimes q_k) \\ r_k - \hat{r}_{KF,k} \\ v_k - \hat{v}_{KF,k} \\ b_g - \hat{b}_{g,KF,k} \\ b_a - \hat{b}_{a,KF,k} \end{bmatrix} \in \begin{bmatrix} R^3 \\ R^3 \\ R^3 \\ R^3 \\ R^3 \end{bmatrix} \quad (8)$$

where \otimes is the quaternion multiplication and q^* denotes the quaternion conjugate of a quaternion q [21]. The elements of $X_{KF,k}$ represent the estimation errors in $\hat{q}_{KF,k} \sim \hat{b}_{a,KF,k}$. In defining $X_{KF,k}$, it is assumed that the quaternion errors in $\hat{q}_{KF,k}$ are small. Thus, $(\hat{q}_{KF,k})^* \otimes q_k$ is approximated by:

$$(\hat{q}_{KF,k})^* \otimes q_k \approx \begin{bmatrix} 1 \\ \bar{q}_{KF,k} \end{bmatrix} \in \begin{bmatrix} R \\ R^3 \end{bmatrix}.$$

With this assumption, the error in $\hat{q}_{KF,k} \in R^4$ is represented by three-dimensional vector $\bar{q}_{KF,k}$ instead of a four-dimensional vector. See [27] for the details about the multiplicative error notation of a quaternion. The initial estimate is set to $\hat{X}_{KF,1} = 0_{15 \times 1}$, and the initial error covariance P_1 is given by:

$$P_1 = E\{(X_{KF,1} - \hat{X}_{KF,1})(X_{KF,1} - \hat{X}_{KF,1})'\} \\ = \begin{bmatrix} P_{q,init} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & P_{r,init} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & P_{v,init} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & P_{b_g,init} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & P_{b_a,init} \end{bmatrix}. \quad (9)$$

The initial attitude error covariance $P_{q,init}$ is given by the algorithm in [25], where R_a and $10^{-11}I_3$ are used as measurement noise covariances of the first and second equations in Equation (5). The small covariance value for the second equation is to make $P_{q,init}$ nonsingular, since $P_{q,init}^{-1}$ is used in the smoothing algorithm. The remaining covariances $P_{r,init} \sim P_{b_a,init}$ can be considered as design parameters, whose values are given in Table 1.

The dynamic equation of $X_{KF,k}$ is given by:

$$X_{KF,k+1} = \begin{bmatrix} \Phi_{k+1,k} & \Psi_{k+1,k} \\ 0_{9 \times 6} & I_6 \end{bmatrix} X_{KF,k} + \begin{bmatrix} w_k \\ 0_{6 \times 1} \end{bmatrix} \quad (10)$$

where:

$$e^{A_k T} = \begin{bmatrix} \Phi_{k+1,k} & \Psi_{k+1,k} \\ 0_{9 \times 6} & I_6 \end{bmatrix} \in \begin{bmatrix} R^{9 \times 9} & R^{9 \times 6} \\ R^{6 \times 9} & R^{6 \times 6} \end{bmatrix} \quad (11)$$

$$A_k = \left[\begin{array}{ccc|cc} -[y_{g,k} \times] & 0_{3 \times 3} & 0_{3 \times 3} & -\frac{1}{2}I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} & 0_{3 \times 3} \\ \hline -2C(\hat{q}_{KF,k})'[y_{a,k} \times] & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & -C(\hat{q}_{KF,k})' \\ 0_{6 \times 3} & 0_{6 \times 3} & 0_{6 \times 3} & 0_{6 \times 3} & 0_{6 \times 3} \end{array} \right] = \left[\begin{array}{c|c} A_{k,1} & A_{k,2} \\ \hline 0_{6 \times 9} & 0_{6 \times 6} \end{array} \right].$$

For a vector $p = [p_1 \ p_2 \ p_3] \in R^3$, $[p \times]$ is defined by:

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

Note that A_k is a lower triangular matrix, and thus, $e^{A_k T}$ is also a lower triangular matrix having the structure in Equation (11) [28].

Table 1. Parameters of the proposed smoother and Kalman filter.

Parameter	Symbol	Values	Related Equation
sampling period	T	0.01	
zero velocity detection parameters	B_g, N_g, B_a, N_a	0.2, 10, 0.4, 10	(4)
sensor noise error covariance	R_g, R_a	$0.001I_3, 0.01I_3$	
measurement noise covariance	R_v, R_r	$0.0001I_3, 0.0001$	(15)
filter and smoother parameter	$P_{r,init}, P_{v,init}$	$0.0001I_3, 0.0001I_3$	(9), (27)
filter and smoother parameter	$P_{b_g,init}, P_{b_a,init}$	$0.000001I_3, 0.000001I_3$	(9), (27)

The process noise w_k is the zero mean white Gaussian noise with:

$$Q_k = E\{w_k w_k'\} = \int_0^T \exp(A_{k,1} r) \begin{bmatrix} \frac{1}{4}R_g & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 10^{-6}I_3 & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & R_a \end{bmatrix} \exp(A'_{k,1} r) dr$$

where the 10^{-6} value in Q_k is added to make Q_k nonsingular, since Q_k^{-1} is used in the smoother. Derivation of A_k and Q_k can be obtained by slightly modifying the results in [19,23]. The computational load of $e^{A_k T}$ can be reduced by using the following approximation:

$$e^{A_k T} \approx I_{15} + A_k T + \frac{1}{2!} A_k^2 T^2. \tag{12}$$

Remark 1. How the estimation error in k -th discrete time evolves after the integration Equation (7) is represented in Equation (10). For example, (10) states that the error in $\hat{r}_{KF,k+1}$ comes from two sources. One is from the estimation error in k -th time (before the integration), and the other is from the process of numerical integration. The latter is not from the numerical integration error (it is ignored), but from the noise terms and bias estimation errors.

In an inertial sensor-only foot motion estimation, there is no external measurement. The fact that the velocity is zero during the zero velocity intervals is used as a fictitious measurement.

During the zero velocity interval, the following equation can be used as a measurement equation:

$$0_{3 \times 1} - \hat{v}_{KF,k} = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & I_3 & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} X_{KF,k} + n_{v,k}, \quad k \in Z_m \quad (13)$$

where $n_{v,k}$ is a fictitious measurement noise representing the confidence of zero velocity interval detection in Equation (4). The z -axis value of $\hat{r}_{KF,k}$ is almost the same when the foot is on the ground, assuming that a person is walking on a flat floor. Thus, the following measurement equation can also be used during the zero velocity interval.

$$0 - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \hat{r}_{KF,k} = \begin{bmatrix} 0_{1 \times 3} & 0 & 0 & 1 & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} \end{bmatrix} X_{KF,k} + n_{r,k}, \quad k \in Z_m \quad (14)$$

where $n_{r,k}$ is a fictitious measurement noise.

We assume that $n_{v,k}$ and $n_{r,k}$ are zero mean white Gaussian noises, where covariances are given by:

$$R_v = E\{n_{v,k}n'_{v,k}\} \in R^{3 \times 3}, \quad R_r = E\{n_{r,k}n'_{r,k}\} \in R. \quad (15)$$

A standard discrete time Kalman filter is applied to Equations (10), (13) and (14). If $k \in Z_m$, we can compute $\hat{X}_{KF,k}$ from the Kalman filter, and these values are used to update $\hat{q}_{KF,k} \sim \hat{b}_{a,KF,k}$.

The Kalman filter algorithm of position estimation ($\hat{r}_{KF,k}$) is summarized in the following for $k_1 \leq k \leq k_2$ assuming that $k_1 \in Z_m$, $k_2 \in Z_m$ and $k \notin Z_m$ for $k_1 < k < k_2$. The algorithm for other values ($\hat{q}_{KF,k}$, $\hat{v}_{KF,k}$, $\hat{b}_{g,KF,k}$ and $\hat{b}_{a,KF,k}$) are similar.

- discrete time $k = k_1 \in Z_m$
 - $\hat{r}_{KF,k}$ is given from the previous step
 - $\hat{r}_{KF,k} = 0$
- discrete time $k_1 < k < k_2$
 - compute $\hat{r}_{KF,k}$ using Equation (7)
 - $\hat{r}_{KF,k} = 0$ from Kalman filter time update Equation (10) with $\hat{r}_{KF,k-1} = 0$
- discrete time $k = k_2 \in Z_m$
 - compute $\hat{r}_{KF,k}$ using Equation (7)
 - compute $\hat{r}_{KF,k}$ from the Kalman filter measurement update Equations (13) and (14)
 - $\hat{r}_{KF,k} \leftarrow \hat{r}_{KF,k} + \hat{r}_{KF,k}$
 - $\hat{r}_{KF,k} = 0$

The result in this section is not a new result, and similar results can be found in [19].

4. Smoothing Algorithm

In this section, the Kalman filter-estimated values ($\hat{q}_{KF,k} \sim \hat{b}_{a,KF,k}$) in Section 3 are improved using the smoother, where the errors in $\hat{q}_{KF,k}$, $\hat{r}_{KF,k}$, $\hat{v}_{KF,k}$, $\hat{b}_{g,KF}$ and $\hat{b}_{a,KF}$ are compensated for. To do that, the estimation error is defined as follows.

$$\begin{aligned}
 \bar{q}_{SM,k} &= \begin{bmatrix} 0_{3 \times 1} & I_3 \end{bmatrix} \hat{q}_{KF,k}^* \otimes q_k \\
 \bar{r}_{SM,k} &= r_k - \hat{r}_{KF,k} \\
 \bar{v}_{SM,k} &= v_k - \hat{v}_{KF,k} \\
 \bar{b}_{g,SM} &= b_g - \hat{b}_{g,KF} \\
 \bar{b}_{a,SM} &= b_a - \hat{b}_{a,KF}
 \end{aligned} \tag{16}$$

where $\hat{b}_{g,KF}$ and $\hat{b}_{a,KF}$ are $\hat{b}_{g,KF,N}$ and $\hat{b}_{a,KF,N}$ (the final values) in the Kalman filter, which can be considered as the most accurate estimation of b_g and b_a . Since b_g and b_a are constant, $\bar{b}_{g,SM}$ and $\bar{b}_{a,SM}$ are not time dependent, while $\bar{q}_{SM,k}$, $\bar{r}_{SM,k}$ and $\bar{v}_{SM,k}$ are time dependent. We note that the estimation errors in Equation (8) represent the errors before the measurement update, while the errors in Equation (16) represent the errors after the measurement update.

Let $X_{SM,k}$ be defined by:

$$X_{SM,k} = \begin{bmatrix} \bar{q}_{SM,k} \\ \bar{r}_{SM,k} \\ \bar{v}_{SM,k} \end{bmatrix}. \tag{17}$$

Equations for time-dependent variables ($\bar{q}_{SM,k}$, $\bar{r}_{SM,k}$ and $\bar{v}_{SM,k}$) are given by:

$$\zeta_k + X_{SM,k+1} = \Phi_{k+1,k} X_{SM,k} + \Psi_{k+1,k} \begin{bmatrix} \bar{b}_{g,SM} \\ \bar{b}_{a,SM} \end{bmatrix} + w_k \tag{18}$$

where Φ and Ψ are defined in Equation (11) and:

$$\zeta_k = \begin{bmatrix} \tilde{q}_{SM,k} \\ \hat{r}_{KF,k+1} - f_{2,k} \\ \hat{v}_{KF,k+1} - f_{3,k} \end{bmatrix}, \tag{19}$$

$$\begin{bmatrix} f_{k,1} \\ f_{k,2} \\ f_{k,3} \end{bmatrix} = f_k \left(\begin{bmatrix} \hat{q}_{KF,k} \\ \hat{r}_{KF,k} \\ \hat{v}_{KF,k} \end{bmatrix}, \begin{bmatrix} n_{g,k} \\ n_{a,k} \end{bmatrix}, \begin{bmatrix} \bar{b}_{g,SM} \\ \bar{b}_{a,SM} \end{bmatrix} \right), \tag{20}$$

$$\tilde{q}_{SM,k} = \begin{bmatrix} 0_{3 \times 1} & I_3 \end{bmatrix} f_{1,k}^* \otimes \hat{q}_{KF,k+1}.$$

Derivation of Equation (18) is now explained. From Equations (16), (19) and (20), we have:

$$\zeta_k + X_{SM,k+1} = \begin{bmatrix} \begin{bmatrix} 0_{3 \times 1} & I_3 \end{bmatrix} f_{1,k}^* \otimes q_{k+1} \\ r_{k+1} - f_{2,k} \\ v_{k+1} - f_{3,k} \end{bmatrix} \in R^9. \tag{21}$$

The second and the third elements of Equation (21) are straightforward. The first element of Equation (21) is derived from the following:

$$\begin{aligned}
 q_{k+1} &= \hat{q}_{KF,k+1} \otimes \begin{bmatrix} 1 \\ \bar{q}_{SM,k+1} \end{bmatrix} \\
 &= f_{1,k} \otimes f_{1,k}^* \otimes \hat{q}_{KF,k+1} \otimes \begin{bmatrix} 1 \\ \bar{q}_{SM,k+1} \end{bmatrix} \\
 &\approx f_{1,k} \otimes \begin{bmatrix} 1 \\ \tilde{q}_{SM,k} + \bar{q}_{SM,k+1} \end{bmatrix}
 \end{aligned}
 \tag{22}$$

where the following approximation is used:

$$f_{1,k}^* \otimes \hat{q}_{KF,k+1} \approx \begin{bmatrix} 1 \\ \tilde{q}_{SM,k+1} \end{bmatrix}, \tag{23}$$

$$\begin{bmatrix} 1 \\ \tilde{q}_{SM,k} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \bar{q}_{SM,k+1} \end{bmatrix} \approx \begin{bmatrix} 1 \\ \tilde{q}_{SM,k} + \bar{q}_{SM,k+1} \end{bmatrix}. \tag{24}$$

From Equation (21), $\zeta_k + X_{SM,k+1}$ denotes the estimation errors of f_k in Equation (20). Thus, Equation (18) represents how the estimation error evolves after the integration Equation (20). Thus, Equation (18) is essentially the same as Equation (10). In the exact same methods in the case of Equation (10), Equation (21) can be obtained by slightly modifying the results in [19,23].

The smoothing problem is formulated as a quadratic optimization problem using the method in [20]. Let an optimization variable \tilde{X} be defined by:

$$\tilde{X} = \begin{bmatrix} X_{SM,1} \\ X_{SM,2} \\ \vdots \\ X_{SM,N} \\ \bar{b}_{g,SM} \\ \bar{b}_{a,SM} \end{bmatrix} \in \begin{bmatrix} R^{9 \times N} \\ R^6 \end{bmatrix}. \tag{25}$$

The smoothing problem can be formulated as the following quadratic optimization problem:

$$\min_{\tilde{X}} J(\tilde{X}) \tag{26}$$

where:

$$\begin{aligned}
 J(\tilde{X}) &= \frac{1}{2} \sum_{k=1}^{N-1} w_k' Q_k^{-1} w_k + \frac{1}{2} \sum_{k \in Z_m} (z_k - \tilde{H}_k X_{SM,k})' \tilde{R}^{-1} (z_k - \tilde{H}_k X_{SM,k}) \\
 &\quad + \frac{1}{2} (X_{SM,1} - X_{init})' P_{X_{init}}^{-1} (X_{SM,1} - X_{init}) \\
 &\quad + \frac{1}{2} (\hat{b}_{g,SM} + \bar{b}_{g,SM} - b_{g,init})' P_{b_{g,init}}^{-1} (\hat{b}_{g,SM} + \bar{b}_{g,SM} - b_{g,init}) \\
 &\quad + \frac{1}{2} (\hat{b}_{a,SM} + \bar{b}_{a,SM} - b_{a,init})' P_{b_{a,init}}^{-1} (\hat{b}_{a,SM} + \bar{b}_{a,SM} - b_{a,init})
 \end{aligned}
 \tag{27}$$

$$X_{init} = \begin{bmatrix} \hat{q}_{KF,1} \\ 0_{3 \times 1} \\ 0_{3 \times 1} \end{bmatrix}, \quad P_{X_{init}} = \begin{bmatrix} P_{q,init} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & P_{r,init} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & P_{v,init} \end{bmatrix}$$

$$z_k = \begin{bmatrix} 0_{3 \times 1} - \hat{v}_{KF,k} \\ 0 - \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \hat{r}_{KF,k} \end{bmatrix}, \quad \tilde{H}_k = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & I_3 \\ 0_{1 \times 3} & 0 & 0 & 1 & 0_{1 \times 3} \end{bmatrix}, \quad \tilde{R} = \begin{bmatrix} R_v & 0 \\ 0 & R_r \end{bmatrix}.$$

Initial information on b_g and b_a are reflected in $b_{g,init} \in R^3$ and $b_{a,init} \in R^3$. For example, $b_{g,init}$ can be computed if a sensor unit is put on the ground without moving for 1~2 min. If there is no information on b_g and b_a , we set $b_{g,init}$ and $b_{a,init}$ to zero.

Inserting Equation (18) into Equation (27), we have:

$$\begin{aligned} J(\tilde{X}) = & \frac{1}{2} \sum_{k=1}^{N-1} (\zeta_k + X_{SM,k+1} - \Phi_{k+1,k} X_{SM,k} - \Psi_{k+1,k} \begin{bmatrix} \bar{b}_{g,SM} \\ \bar{b}_{a,SM} \end{bmatrix})' Q^{-1} \\ & (\zeta_k + X_{SM,k+1} - \Phi_{k+1,k} X_{SM,k} - \Psi_{k+1,k} \begin{bmatrix} \bar{b}_{g,SM} \\ \bar{b}_{a,SM} \end{bmatrix}) \\ & + \frac{1}{2} \sum_{k \in Z_m} (z_k - \tilde{H}_k X_{SM,k})' R^{-1} (z_k - \tilde{H}_k X_{SM,k}) + \frac{1}{2} (X_{SM,1} - X_{init})' P_{X_{init}}^{-1} (X_{SM,1} - X_{init}) \\ & + \frac{1}{2} (\hat{b}_{g,SM} + \bar{b}_{g,SM} - b_{g,init})' P_{b_{g,init}}^{-1} (\hat{b}_{g,SM} + \bar{b}_{g,SM} - b_{g,init}) \\ & + \frac{1}{2} (\hat{b}_{a,SM} + \bar{b}_{a,SM} - b_{a,init})' P_{b_{a,init}}^{-1} (\hat{b}_{a,SM} + \bar{b}_{a,SM} - b_{a,init}). \end{aligned} \quad (28)$$

It is not difficult to see that Equation (28) is a quadratic function of $\tilde{X} \in R^{9 \times N + 6}$:

$$J(\tilde{X}) = \frac{1}{2} \tilde{X}' M_1 \tilde{X} + M_2' \tilde{X} + M_3 \quad (29)$$

where M_1 , M_2 and M_3 can be computed from Equation (28). M_1 and M_2 are given in the Appendix (M_3 is omitted, since it is irrelevant in the optimization). The minimizing solution of Equation (29) can be computed by solving the following linear equation:

$$M_1 \tilde{X}^* + M_2 = 0 \quad (30)$$

where \tilde{X}^* is the minimizing solution.

Once the minimizing solution \tilde{X}^* is computed, the smoother estimated values are computed using Equation (16). For example, the position estimate is computed as follows:

$$\hat{r}_{SM,k} = \hat{r}_{KF,k} + X_{SM,k}^*$$

where $\hat{r}_{SM,k}$ is the smoother estimate of r_k and $X_{SM,k}^*$ is the $9(k-1) + 1 \sim 9k$ rows of \tilde{X}^* .

The result in this section is a new result. The main contribution of this section is that the foot motion Kalman filter error is compensated for using the quadratic optimization problem.

5. Smoothing Optimization Solution

Although the smoothing problem boils down to a simple linear Equation (30), $M_1 \in R^{(9N+6) \times (9N+6)}$ becomes very large when N is large. Suppose we solve Equation (30) using the Cholesky decomposition. Let L be the Cholesky triangle (lower triangular matrix) satisfying:

$$M_1 = LL'. \quad (31)$$

where $L_{11} \in R^{9N \times 9N}$ is a lower triangular matrix. Find $U_1 \in R^{9N \times 1}$ and $U_2 \in R^{9N \times 1}$ from the following linear equation:

$$\begin{aligned} L_{11}U_1 &= M_{12} \\ L_{11}U_2 &= M_{21}. \end{aligned} \quad (39)$$

Inserting Equation (39) into Equation (37), we have:

$$(M_{13} - U_1'U_1)\tilde{X}_2 = M_{22} - U_1'U_2. \quad (40)$$

After computing \tilde{X}_2 in Equation (40), we compute \tilde{X}_1 from the following equations:

$$\begin{aligned} L_{11}U_3 &= M_{21} - M_{12}\tilde{X}_2 \\ L_{11}'\tilde{X}_1 &= U_3. \end{aligned} \quad (41)$$

When solving Equations (38)–(41), the algorithms for banded matrices in [29] can be used for efficient computation and memory usage. The computational load of the Cholesky decomposition of the banded matrix Equation (38) is $(9N(p^2 + 3p))/2$ flops ($p = 17$ is the bandwidth of M_{11} , and $9N \gg p$ is assumed) and $9N$ square root computation. The computational load of each equation in Equations (39) and (41) is $9N(p + 1)$ flops. Equation (40) is a small dimensional problem ($\tilde{X}_2 \in R^6$).

Compared with the standard method, the solution using the banded structure of M_{11} requires much smaller computation and memory. For example, the Cholesky decomposition requires approximately 3.6×10^7 flops and $9N$ square root computation in the case of $N = 12,000$. Compared with 2×10^{14} flops in the standard computation case, its computational requirement is significantly small.

The main contribution of this section is to derive an efficient solution to the quadratic optimization problem and to investigate the quantitative computational requirement.

6. Experiment

Two experiments are performed to verify the proposed algorithm. In the experiments, an inertial sensor unit (Xsens MTi sensor unit) is attached on the foot, as shown in Figure 2. The parameters used in both experiments are given in Table 1.

In the first experiment, a person walked three steps, while the movement of the foot is tracked using an optical motion tracking system (Optitrack six Flex 13 camera system). The experimental setup is given in Figure 2. Six optical markers are used in the optical tracking system, and the center position of six markers is used as the true foot position. Since the coordinate systems of the optical tracker and the inertial sensor unit are different, the optical tracker output is rotated and translated.

The estimated position (Kalman filter and the proposed smoother) with the flat floor assumption (see Equation (14)) is given in Figure 3: the left figure is the z -axis movement, and the right figure is the xy -axis position. In the left figure, the zero velocity intervals are represented by the “*” symbol.

Assuming that the motion tracker value is the true position value, we can see that the Kalman filter gives a good final position estimation (around 5 s). However, the position estimated during walking is not accurate; for example, see the values between 2 and 3 s. Thus, the Kalman filter result may be good enough for the pedestrian navigation system, where the final position is important. However, the Kalman result is not good enough for gait analysis, where it is important to know how the foot has

moved in addition to the final position. The final estimated position of the smoother is almost similar to that of the Kalman filter. However, we can see that there is significant improvement in the middle foot position estimation.

Figure 2. Experimental setup for the first experiment.

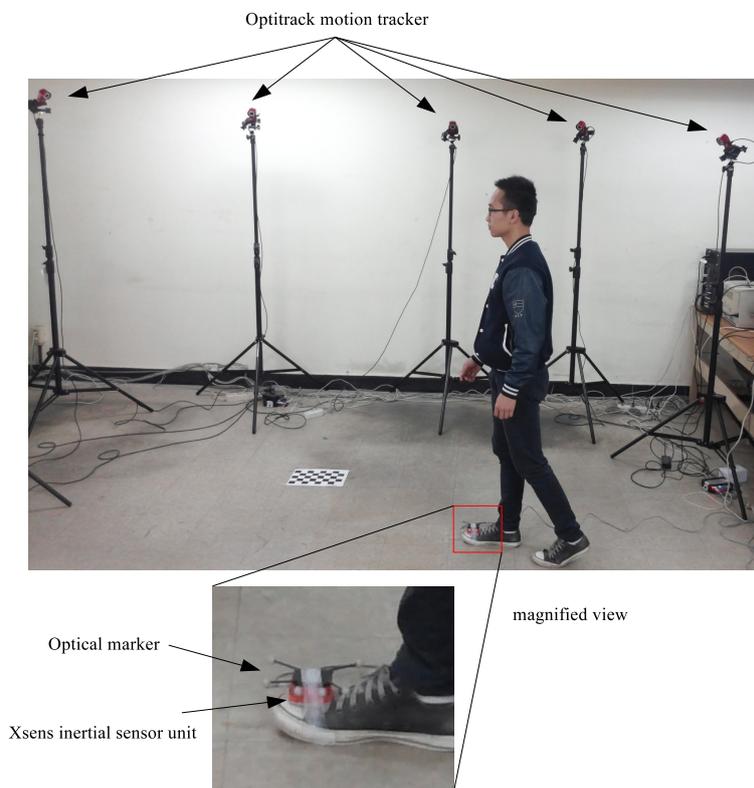
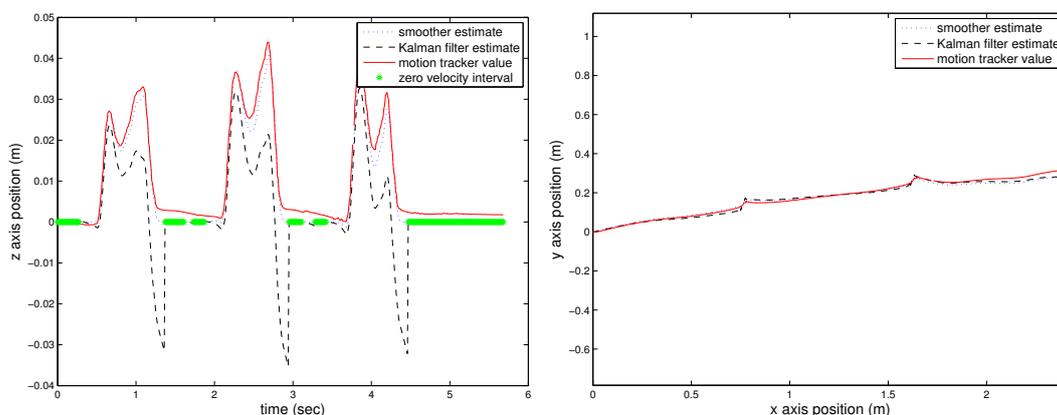


Figure 3. Estimated position with the flat floor assumption: z -axis position (Left) and xy -axis position (Right).



The z -axis estimation error is compared quantitatively in Table 2. The z -axis estimation error $e_{z,k}$ is defined by:

$$e_{z,k} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} (r_k - \hat{r}_k)$$

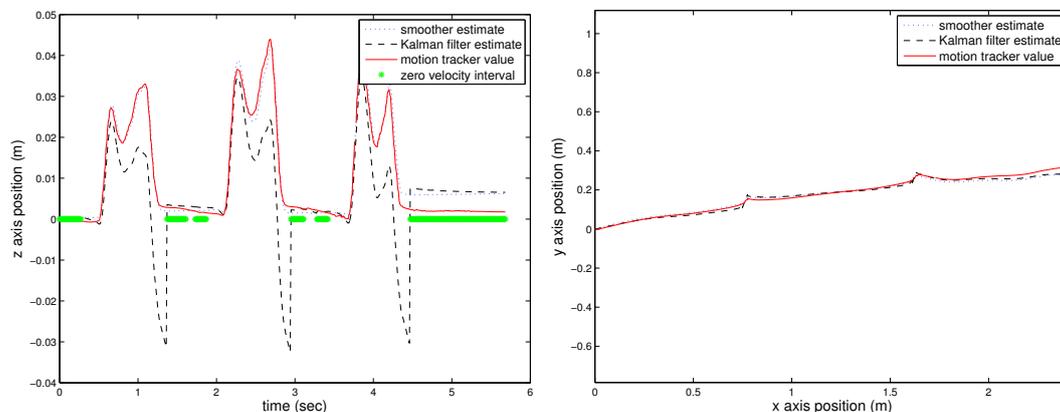
where \hat{r}_k is either $\hat{r}_k = \hat{r}_{KF,k}$ or $\hat{r}_k = \hat{r}_{SM,k}$.

Table 2. Estimated z position error squared sum with the flat floor assumption.

Algorithms	$E_1 = \sum_{k \in Z_m} e_{z,k}^2$ (error squared sum when the foot is on the ground)	$E_2 = \sum_{k \notin Z_m} e_{z,k}^2$ (error squared sum when the foot is in the air)	$E_1 + E_2$
Kalman filter	0.0008	0.0807	0.0815
Proposed smoother	0.0009	0.0020	0.0029

The z -axis estimation error squared sum is given in Table 2. The estimation error of both algorithms during the zero velocity interval ($k \in Z_m$) are almost the same, since the estimated values becomes zero with the flat floor assumption. The estimation error when the foot is in the air are different; the proposed smoother gives significantly smaller estimation error: 0.0020 in the case of the proposed smoother and 0.0807 in the case of the Kalman filter. Furthermore, the maximum z -axis position errors for the Kalman filter and the proposed smoother are 4 cm and 0.5 cm, respectively. This improved estimation is due to the fact the smoother uses all of the available information.

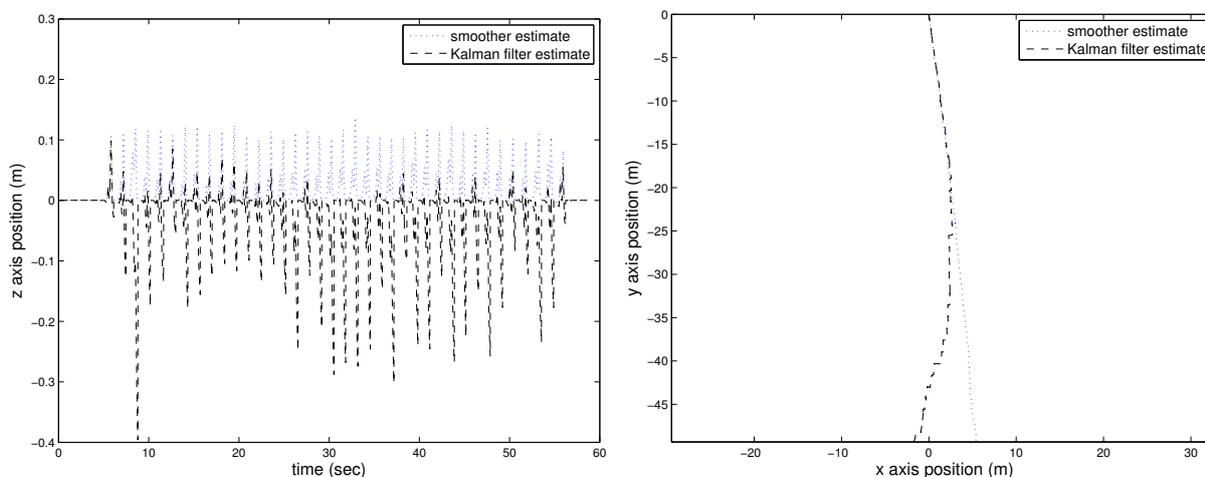
The position estimation by the Kalman filter and the proposed smoother without the flat floor assumption are given in Figure 4. Without the flat floor assumption, we can see that the z -axis error increases since the z -axis position error is not corrected using Equation (14).

Figure 4. Estimated position without the flat floor assumption: z -axis position (**Left**) and xy -axis position (**Right**).

Due to the viewpoint constraint of the optical tracker, the walking distance is rather limited in the first experiment. In the second experiment, a person walked 50 m along a straight line on the flat floor. In this experiment, there is no ground truth value, except the z -axis value, which should be almost constant during the zero velocity intervals, since a person walked on a flat floor.

The Kalman filter estimated and smoother estimated positions are given in Figure 5, both with the flat floor assumption. In the Kalman filter result, the estimated xy trajectory is slightly curved. The smoother estimated xy trajectory is a little bit more straight. The true trajectory should be almost a straight line, since the person walked along a straight line.

Figure 5. Estimated position (50-m walk): z -axis position (Left) and xy -axis position (Right).



Similar to the first experiment, there is no significant improvement in the smoother position estimation for the position where the foot is on the floor (that is, after the zero velocity updating). The smoother, however, gives more accurate foot motion estimation while the foot is moving in the air, which is important in the gait analysis. This can be seen in the z -axis graphs: the Kalman filter-estimated z -axis trajectory is not accurate, since the z -axis value cannot be negative (which means the foot is beneath the floor).

The 50-m walk is repeated by four subjects: five times for each person, thus a total of 20 experiments. The total walk length estimation is given in Table 3. The average estimated walk length is 48.99 m with the standard deviation being 0.26 m. From the small standard deviation, the estimation result does not seem to depend on the subjects.

Table 3. Four persons' 50-m walk experiment results.

Subject Number	Final xy Position Estimated Value (m)	Average (m)
1	48.97, 48.97, 49.16, 49.06, 49.02	49.04
2	49.36, 49.29, 49.21, 49.37, 49.14	49.27
3	48.52, 48.98, 48.46, 48.69, 48.45	48.62
4	48.98, 49.00, 48.92, 49.08, 49.06	49.01

We note that the xy -axis position error increases as the walk length increases. This is unavoidable, since the motion is estimated using the inertial navigation algorithm without any external reference. The z -axis position error increment is limited with the flat floor assumption.

The computation time of the proposed smoother was 8.7 s (MATLAB running on Intel Core i5-2310 CPU Windows PC, 2.9 GHz), where the walk time is 58.92 s ($N = 5892$).

7. Conclusions

In this paper, a smoother algorithm is proposed for gait analysis to obtain more accurate foot motion estimation than the filter-based estimation. In the smoother algorithm, the motion estimation problem is formulated as a quadratic optimization problem, and an efficient computational algorithm is given.

In the first experiment, we have compared the results of the Kalman filter and the proposed smoother. The Kalman filter-based estimation is quite accurate after the zero velocity update. However, foot motion estimation is not accurate when the foot is in the air. We have shown that the motion estimation is significantly improved for the foot motion in the air if the smoother is used: the z -axis position error squared sum when the foot is in the air is 0.0807 (Kalman filter) and 0.0020 (the proposed smoother). Thus, the proposed smoother is a useful algorithm for gait analysis, where the foot motion estimation is important.

In the second experiment, we investigated a long walk foot motion estimation (50 m). There is not much difference in the final position estimation of the Kalman filter and the proposed smoother. The 50-m walk was tested using four subjects, and we found that the estimation does not depend on the subject. This is not surprising, since there are no person-dependent parameters in the algorithm.

Since the proposed smoother algorithm is more complicated than the Kalman filter, the computational time is slow. In the second experiment, the computation time of the proposed smoother is about 8.7 s to process about one minute of walk data (the sampling period is 0.01 s), while it took 0.8 s for the Kalman filter. Thus, the estimation performance is improved with the additional computation time.

Acknowledgments

This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (No. 2011-0002770).

A. Appendix

In this Appendix, M_{11} , M_{12} , M_{13} , M_{21} , M_{22} (partition of M_1 and M_2 in Equation (33)) are defined. We use the notation $M_{11}[i, j]$, which represents a $R^{9 \times 9}$ matrix extracting $9(i - 1) + 1 \sim 9i$ rows and $9(j - 1) + 1 \sim 9j$ columns of M_{11} . Similarly, we use $M_{12}[i] \in R^{9 \times 6}$ and $M_{21}[i] \in R^{9 \times 1}$ notations.

- M_{11}

$$M_{11}[k, k] = \begin{cases} P_{X_{init}}^{-1} + \Phi'_{21} Q_k^{-1} \Phi_{21}, & k = 1 \\ \Phi'_{k+1,k} Q_k^{-1} \Phi_{k+1,k} + Q_k^{-1}, & 2 \leq k \leq N - 1 \\ Q_k^{-1}, & k = N. \end{cases}$$

$$M_{11}[k, k] = M_{11}[k, k] + \tilde{H}'_k \tilde{R}^{-1} \tilde{H}_k, \quad k \in Z_m$$

$$M_{11}[k, k + 1] = -\Phi'_{k+1,k} Q_k^{-1}, \quad 1 \leq k \leq N - 1$$

$$M_{11}[k + 1, k] = -Q_k^{-1} \Phi_{k+1,k}, \quad 1 \leq k \leq N - 1$$

- M_{12}

$$M_{12}[k] = \begin{cases} \Phi'_{k+1,k} Q_k^{-1} \Psi_{k+1,k}, & k = 1 \\ \Phi'_{k+1,k} Q_k^{-1} \Psi_{k+1,k} - \Psi_{k,k-1}, & 2 \leq k \leq N - 1 \\ -\Psi_{k,k-1}, & k = N. \end{cases}$$

- M_{13}

$$M_{13} = \left(\sum_{k=1}^{N-1} \Psi'_{k+1,k} Q_k^{-1} \Psi_{k+1,k} \right) + \begin{bmatrix} P_{b_g,init}^{-1} & 0_{3 \times 3} \\ 0_{3 \times 3} & P_{b_a,init}^{-1} \end{bmatrix}$$

- M_{21}

$$M_{21}[k] = \begin{cases} -\zeta'_k Q_k^{-1} \Phi_{k+1,k}, & k = 1 \\ -\zeta'_k Q_k^{-1} \Phi_{k+1,k} + \zeta'_{k-1} Q_k^{-1}, & 2 \leq k \leq N - 1 \end{cases}$$

- M_{22}

$$M_{22} = \left(\sum_{k=1}^{N-1} -\zeta'_k \Psi_{k+1,k} \right) + \begin{bmatrix} (\hat{b}_{g,SM} - b_{g,init})' P_{b_g,init}^{-1} & 0 \\ 0 & (\hat{b}_{a,SM} - b_{a,init})' P_{b_a,init}^{-1} \end{bmatrix}$$

Conflicts of Interest

The author declares no conflict of interest.

References

1. Perry, J. *Gait Analysis: Normal and Pathological Function*; SLACK Incorporated: Thorofare, NJ, USA, 1992.
2. Lee, H.; Lee, H.; Kim, E. A New Gait Recognition System based on Hierarchical Fair Competition-based Parallel Genetic Algorithm and Selective Neural Network Ensemble. *Int. J. Control Autom. Syst.* **2014**, *12*, 202–207.
3. Calliess, T.; Bocklage, R.; Karkosch, R.; Marschollek, M.; Windhagen, H.; Schulze, M. Clinical Evaluation of a Mobile Sensor-Based Gait Analysis Method for Outcome Measurement after Knee Arthroplasty. *Sensors* **2014**, *14*, 15953–15964.
4. Whittle, M.W. *Gait Analysis: An Introduction*, 4th ed.; Elsevier Ltd.: Philadelphia, PA, USA, 2007.
5. Liu, T.; Inoue, Y.; Shibata, K. A wearable ground reaction force sensor system and its application to the measurement of extrinsic gait variability. *Sensors* **2010**, *10*, 10240–10255.
6. Windolf, M.; Gotzenb, N.; Morlockb, M. Systematic accuracy and precision analysis of video motion capturing systems—Exemplified on the Vicon-460 system. *J. Biomech.* **2008**, *41*, 2776–2780.
7. Zheng, R.; Liu, T.; Inoue, Y.; Shibata, K.; Liu, K. Kinetics Analysis of Ankle, Knee and Hip Joints Using a Wearable Sensor Systems. *J. Biomech. Sci. Eng.* **2008**, *3*, 1031–1037.
8. Zhang, B.; Jiang, S.; Wei, D.; Marschollek, M.; Zhang, W. State of the Art in Gait Analysis Using Wearable Sensors for Healthcare Applications. In Proceedings of the 2012 IEEE/ACIS 11th International Conference on Computer and Information Science (ICIS), Shanghai, China, 30 May–1 June 2012; pp. 213–218.

9. Cooper, G.; Sheret, I.; McMillian, L.; Siliverdis, K.; Sha, N.; Hodgins, D.; Kenney, L.; Howard, D. Inertial sensor-based knee flexion/extension angle estimation. *J. Biomech.* **2009**, *42*, 2678–2865.
10. Schepers, H.M.; van Asseldonk, E.H.F.; Baten, C.T.; Veltink, P.H. Ambulatory estimation of foot placement during walking using inertial sensors. *J. Biomech.* **2010**, *43*, 3138–3143.
11. Liu, T.; Inoue, Y.; Shibata, K. Development of a wearable sensor system for quantitative gait analysis. *Measurement* **2009**, *42*, 978–988.
12. van Acht, V.; Bongers, E.; Lambert, N.; Verberne, R. Miniature Wireless Inertial Sensor for Measuring Human Motions. In Proceedings of the EMBS 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon, France, 23–26 August 2007; pp. 6278–6281.
13. Zheng, Y.; Chan, K.C.; Wang, C. Pedalvatar: An IMU-based real-time body motion capture system using foot rooted kinematic model. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago, IL, USA, 14–18 September 2014; pp. 4130–4135.
14. Bulling, A.; Blanke, U.; Schiele, B. A Tutorial on Human Activity Recognition Using Body-worn Inertial Sensors. *ACM Comput. Surv.* **2014**, *46*, 1–33.
15. Savage, P.G. Strapdown Inertial Navigation Integration Algorithm Design Part 1: Attitude Algorithms. *J. Guid. Control Dynamics* **1998**, *21*, 19–28.
16. Park, S.K.; Suh, Y.S. A Zero Velocity Detection Algorithm Using Inertial Sensors for Pedestrian Navigation Systems. *Sensors* **2010**, *10*, 9163–9178.
17. Foxlin, E. Pedestrian Tracking with Shoe-Mounted Inertial Sensors. *IEEE Comput. Graph. Appl.* **2005**, *25*, 38–46.
18. Karimi, H.A.; Jiang, M.; Zhu, R. Pedestrian Navigation Services: Challenges and Current Trends. *GEOMATICA* **2013**, *67*, 259–271.
19. Suh, Y.S. A Smoother for Attitude and Position Estimation Using Inertial Sensors with Zero Velocity Intervals. *IEEE Sens. J.* **2012**, *12*, 1255–1262.
20. Psiaki, M.L. Backward-Smoothing Extended Kalman Filter. *J. Guid. Control Dyn.* **2005**, *28*, 885–894.
21. Kuipers, J.B. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*; Princeton University Press: Princeton, NJ, USA, 1999.
22. Titterton, D.H.; Weston, J.L. *Strapdown Inertial Navigation Technology*; IPeter Peregrinus Ltd.: London, UK, 1997.
23. 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.* **2014**, *63*, 943–952.
24. Skog, I.; Nilsson, J.O.; Handel, P. Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems. In Proceedings of International Conference on Indoor Positioning and Indoor Navigation, Hoenggerberg, Switzerland, 15–17 September 2010; pp. 1–6.
25. Shuster, M.; Oh, S. Three-axis attitude determination from vector observations. *J. Guid. Control* **1981**, *4*, 70–77.
26. Cheng, Y.; Shuster, M.D. QUEST and the Anti-QUEST: Good and Evil Attitude Estimation. *J. Astronaut. Sci.* **2005**, *53*, 337–351.

27. Markley, F.L. Multiplicative vs. Additive Filtering for Spacecraft Attitude Determination. In Proceedings of the 6th Cranfield Conference on Dynamics and Control of Systems and Structures in Space, 8 September 2004; pp. 467–474.
28. Loan, C.F.V. Computing Integrals Involving the Matrix Exponential. *IEEE Trans. Automat. Control* **1978**, *AC-23*, 395–404.
29. Golub, G.H.; van Loan, C.F. *Matrix Computations*; The Johns Hopkins University Press: Baltimore, MD, USA, 1983.
30. Triggs, B.; McLauchlan, P.F.; Hartley, R.I.; Fitzgibbon, A.W. Bundle Adjustment—A Modern Synthesis. *Lect. Notes Comput. Sci.* **2000**, *1883*, 298–372.

© 2014 by the author; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).