

Article

A Zero Velocity Detection Algorithm Using Inertial Sensors for Pedestrian Navigation Systems

Sang Kyeong Park and Young Soo Suh *

Department of Electrical Engineering, University of Ulsan, Namgu, Ulsan 680-749, Korea;
E-Mail: damiro76@hotmail.com

* Author to whom correspondence should be addressed; E-Mail: yssuh@ulsan.ac.kr;
Tel.: +82-52-259-1524; Fax: +82-52-259-1686.

Received: 20 August 2010; in revised form: 20 September 2010 / Accepted: 28 September 2010 /
Published: 13 October 2010

Abstract: In pedestrian navigation systems, the position of a pedestrian is computed using an inertial navigation algorithm. In the algorithm, the zero velocity updating plays an important role, where zero velocity intervals are detected and the velocity error is reset. To use the zero velocity updating, it is necessary to detect zero velocity intervals reliably. A new zero detection algorithm is proposed in the paper, where only one gyroscope value is used. A Markov model is constructed using segmentation of gyroscope outputs instead of using gyroscope outputs directly, which makes the zero velocity detection more reliable.

Keywords: hidden Markov model; pedestrian navigation system; zero velocity update method; Kalman filter

1. Introduction

A pedestrian or personal navigation system provides position information about a person indoors or outdoors. For example, if a firefighter carrying a personal navigation system is injured in the line of duty, we can know her/his position from the information transmitted by a pedestrian navigation system. Pedestrian navigation systems using inertial sensors are proposed in [1–6]. The main advantage of inertial sensor-based systems is that they are environment-independent; pedestrian navigation systems using vision or wireless communication require that landmarks or wireless nodes be installed in the environments [7,8].

There are many different pedestrian navigation systems using inertial sensors. They use similar inertial navigation algorithms to compute the position [9,10]. Also they all use zero velocity updating algorithms. When an inertial navigation algorithm is used, the position and velocity errors diverge by a few seconds without error resetting. During normal walking cycles, a foot touches the ground almost periodically and stays on the ground for a short time (usually about 0.1~0.3 s), which is called the zero velocity interval. In the zero velocity updating algorithm, this zero velocity interval is detected and thus the velocity error is reset to zero. Accordingly the position and velocity errors diverge very slowly, and consequently being able to reliably detect this zero velocity interval is an important issue.

Slightly different zero detection algorithms are used in [1-3,5,6]. In [1], the zero velocity interval is determined based on gyroscope and accelerometer output norms. If the outputs are less than some threshold for a predetermined time, the zero velocity interval is decided. In [2,3], similar algorithms are used, where the main difference is which sensor values are used. In [2], the zero velocity interval is determined based on *z*-axis accelerometer and *y*-axis gyroscope outputs (see Figure 1 for the coordinate frame). In [3], the zero velocity interval is determined based on norms of gyroscope outputs. In [5], the zero velocity interval is detected based on the variance of accelerometer values. In [6], the zero velocity interval is detected based on norms of accelerometers and gyroscopes along with variance of accelerometers.

To use the zero velocity algorithms in [1-3,5,6], some threshold values (both for sensor values and time duration) must be assigned. These threshold values could differ significantly when a person is walking and running. The sensor values tend to become large and the interval duration tend to become shorter when a person is running. If we choose the threshold values too small, then we cannot detect zero velocity intervals when a person is running. On the other hand, if the threshold values are too large, then we could detect wrong zero velocity intervals. In general, the methods in [1-3,5,6] are relatively good for walking scenarios, but not for both walking and running.

In this paper, we propose a new zero velocity detection algorithm using a hidden Markov model [11,12]. The most relevant result is [4], where a zero velocity detection algorithm using a hidden Markov model is also used. The main differences are twofold. One is that only one gyroscope value is used in this paper while a foot sensor is also used in [4], which makes the zero detection problem relatively easy. The other difference is that a Markov model is constructed using segmentation of gyroscope outputs instead of using gyroscope outputs directly, which makes the zero velocity detection more reliable.

2. Inertial Sensor-Based Pedestrian Navigation Systems and Zero Velocity Intervals

In inertial sensor-based pedestrian navigation systems, inertial sensors are installed on a shoe, as can be seen in Figure 1. Note that the *y*-axis is nearly perpendicular to the sagittal plane. Thus, when a person walks or runs, the dominant rotation axis is the *y*-axis. We use *y*-axis gyroscope value to determine the zero velocity intervals. The gyroscope measures angular rate of foot rotations, and *y*-axis gyroscope output in our system has positive values when a foot is rotating clockwise.

Figure 2 shows the *y*-axis gyroscope output of two walking steps. From experiments, we found that a typical pattern is given as in Figure 3, where the pattern consists of four segments. The pattern starts with the zero value segment and it has two positive value segments and one negative value segment

between them. These four segments are related to the foot movement. When a foot is on the ground, the output value is near upon zero. As a foot takes off the ground, the gyroscope output has positive values. And then it has negative values when a foot is swinging. Lastly, gyroscope output has positive values once more when the heel of a foot (or shoe) contacts the ground. This pattern is repeated in walking and running. We assign four states (1,2,3, and 4) to each segment, as in Figure 3. The details are discussed in Section 3.

Figure 1. Inertial sensors for the pedestrian navigation system.

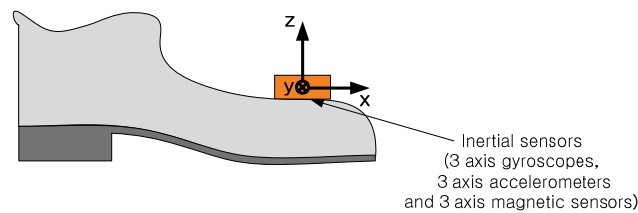


Figure 2. The y-axis gyroscope output in walking.

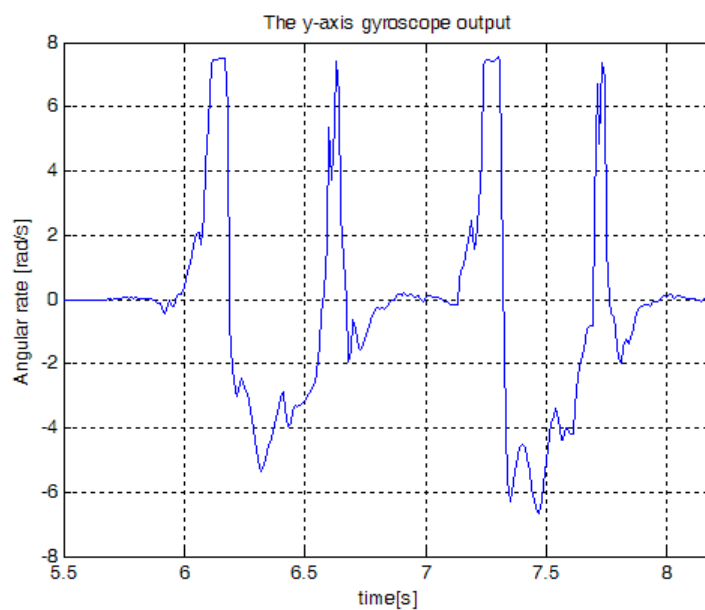
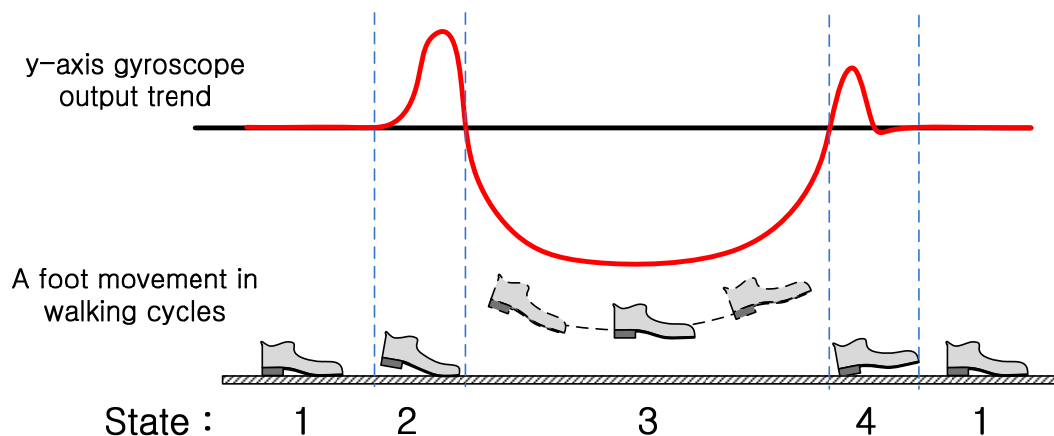
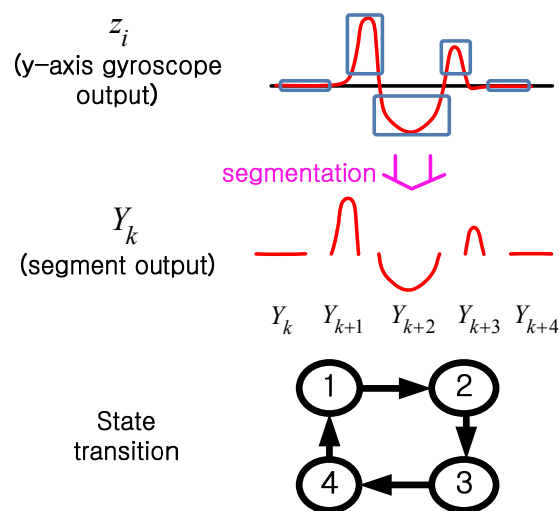


Figure 3 shows how the y-axis gyroscope value changes during the normal walking cycles. A foot touches the ground almost periodically for a short interval. During this short interval, a foot is fixed on the ground and not moving. These short intervals are called “zero velocity intervals”. Due to sensor noises in the real data, it is not always easy to determine the zero velocity interval. Similar patterns can be observed during running cycles. In the running cycles, the zero velocity interval becomes shorter and it is more difficult to detect the zero velocity interval. We divided the gait pattern into four states based on the features of the y-axis gyroscope output. At this time, the state 1 is zero velocity interval.

Figure 3. y-axis gyroscope value trend and a foot movement in normal walking cycles.

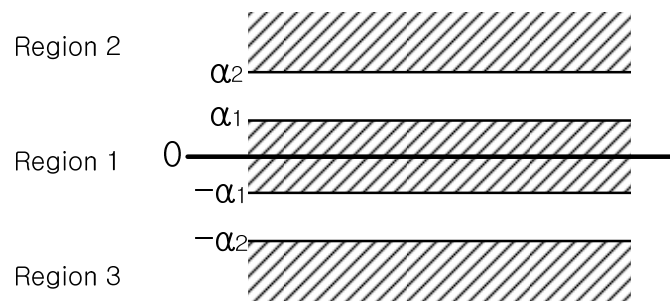
3. Hidden Markov Model

In this section, we introduce a hidden Markov model for the zero velocity detection. The walking states are modeled as a finite state machine (see Figure 4), whose states can be observed through y-axis gyroscope value z_i . Instead of using z_i directly as in [4], Y_k (a series of segments derived from z_i) is used as an output in the hidden Markov model.

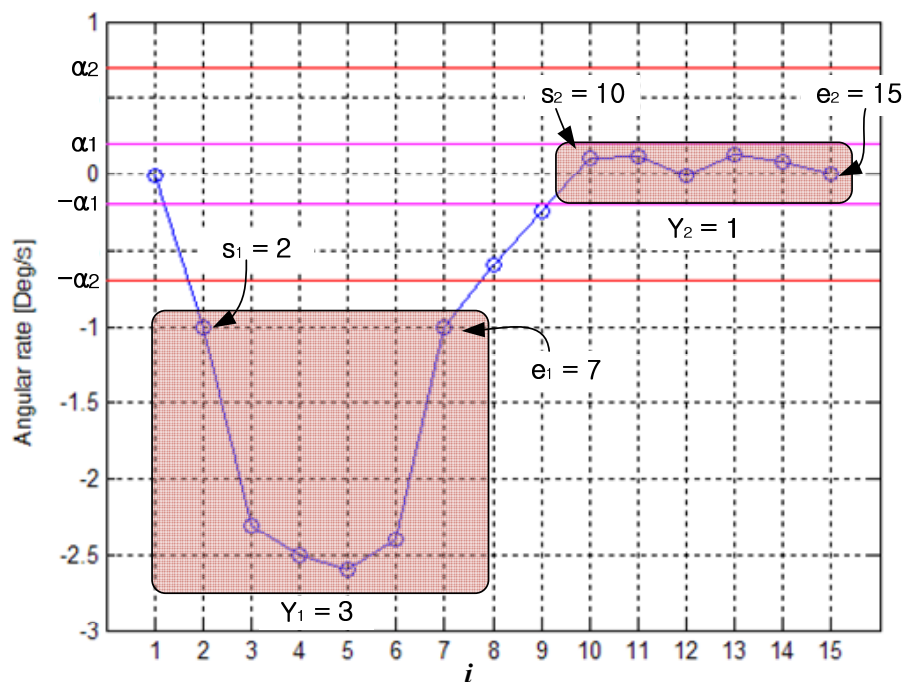
Figure 4. Hidden Markov model based on segmentation of z_i .

The segmentation of y-axis gyroscope value z_i is explained. First three regions (Region 1, 2, and 3) are defined depending on z_i values (see Figure 5). To formally state this, we define a function $f(z_i)$:

$$f(z_i) = \begin{cases} 1, & |z_i| \leq \alpha_1 \\ 2, & z_i > \alpha_2 \\ 3, & z_i < -\alpha_2 \\ 0, & \text{otherwise} \end{cases} \quad \text{where } 0 < \alpha_1 \leq \alpha_2. \quad (1)$$

Figure 5. The region classification.

If z_i values stay in the region j more than N_j ($j=1,2,3$) (that is, at least N_j consecutive z_i values have the same $f(z_i)$), those z_i values make a segment.

Figure 6. Example of a segmentation.

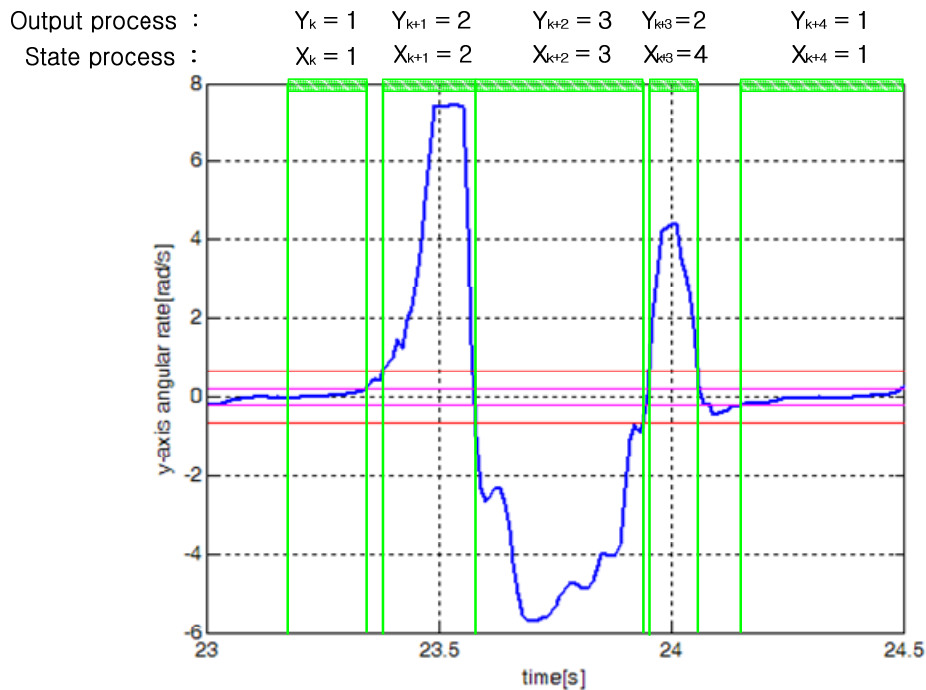
A segmentation example of is given in Figure 6 with $N_1 = N_2 = N_3 = 3$. Since six consecutive z_i ($2 \leq i \leq 7$) values have the same $f(z_i) = 3$ (that is $6 \geq N_3 = 3$), they form the first segment. We will use Y_k (k is the segment index) to denote the segment value, which is defined by $Y_k = f(z_i)$, where z_i belongs to the k -th segment. In Figure 6, we note that $Y_1 = 3$. Similarly, the second segment is $Y_2 = 1$. Also note that some z_i values do not belong to any segments. For example, z_8 and z_9 do not belong to any segments and these values are ignored.

In Figure 6, s_k and e_k denote the starting and ending indices of the k -th segment, respectively: note that $s_1 = 2$, $e_1 = 7$, $s_2 = 10$ and $e_2 = 15$.

We will identify each segment using the finite state hidden Markov model. We assume that there is a Markov process X_k , which represents four walking states in Figure 3. X_k assumes one of four states: 1, 2, 3, and 4. When a person is walking, a typical state transition is $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow \dots$. The Markov

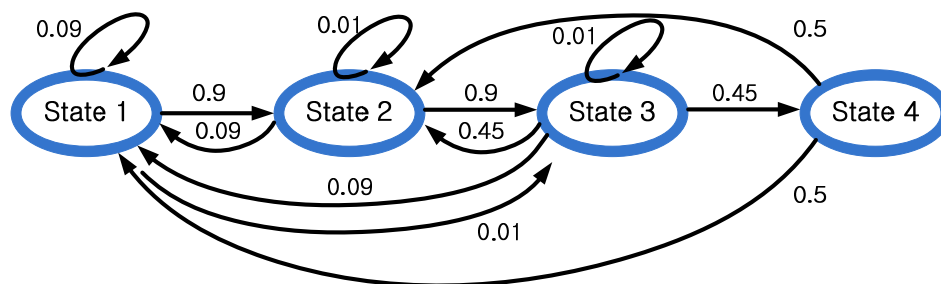
process X_k cannot be obtained directly but is estimated through the output Y_k (that is, segment output), which can be observed. An example of state transitions of X_k and output Y_k is given in Figure 7.

Figure 7. State transition example of normal walking cycles.



From experimental results, we determined the state transition probability as in Figure 8. A typical state transition is $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$. However, there are also other possible transition patterns. For example, $4 \rightarrow 3 \rightarrow 2$ transition (state 1 is missing) is possible. In this case, output data corresponding to state 1 is too short to be recognized as Y_k and state 1 is considered to be missing.

Figure 8. The state transition diagram.



From Figure 8, the state transition probability matrix A is given by:

$$A = \begin{bmatrix} 0.09 & 0.09 & 0.09 & 0.5 \\ 0.9 & 0.01 & 0.45 & 0.5 \\ 0.01 & 0.9 & 0.01 & 0 \\ 0 & 0 & 0.45 & 0 \end{bmatrix} \quad (2)$$

where the (m,n) -th element of A is given by:

$$A_{mn} = \Pr[X_{k+1} = m | X_k = n], \quad \sum_{m=1}^4 A_{mn} = 1.$$

The determination of A is a trial-and-error process, which is explained in the following. A normal state transition is $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$. If walking and running cycles consist of only normal transitions, A should be as follows:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

However, we found that some states are missing during walking and running. A person walked or ran 30 steps and we examined Y_k . If no state is missing, Y_k should be $1 \rightarrow 2 \rightarrow 3 \rightarrow 2$ for one walking or running cycle. If Y_k is $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$, then we know state 4 is missing. The results are summarized in Table 1.

Table 1. Missing states during walking and running (30 steps test for each trial).

	Number of missing states in walking			Number of missing states in running			Missing probability [%]
	1st	2nd	3rd	1st	2nd	3rd	
State 1	0	0	0	20	28	14	34.4
State 2	0	0	0	0	0	0	0.0
State 3	0	0	0	0	0	0	0.0
State 4	0	0	0	13	14	3	16.7

From Table 1, we can draw the following conclusions:

- State 1 is sometimes missing (in particular, when a person is running)
- State 2 and 3 are not missing
- State 4 is sometimes missing (in particular, when a person is running).

Thus we modified A_1 to allow the following state transitions: $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ (state 4 is missing), $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2$ (state 1 is missing), and $1 \rightarrow 2 \rightarrow 3 \rightarrow 2$ (state 1 and 4 are missing):

$$A_1 = \begin{bmatrix} 0 & 0 & 0 & p_3 \\ 1 & 0 & p_1 & p_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & p_2 & 0 \end{bmatrix}$$

If we choose large p_1 (for example $p_1 = 0.9$ and $p_2 = 0.1$), it tends to estimate the transition $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ (normal cycle) as $1 \rightarrow 2 \rightarrow 3 \rightarrow 2$ (state 1 and 4 are missing). On the other hand, if we choose small p_1 (for example, $p_1 = 0.1$ and $p_2 = 0.9$), it tends to estimate the transition $1 \rightarrow 2 \rightarrow 3 \rightarrow 2$ (state 1 and 4 are missing) as $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ (normal cycle). If difference between p_1 and p_2 is not large, the estimation results are similar.

Finally, we added small values for most transitions to allow unusual transitions. For example, when a person is standing and idly moving foot, unusual transitions (such as $2 \rightarrow 1$) could happen.

We have explained guidelines of how A are determined. Some values (such as 0.01 and 0.09) are determined by changing values and testing the results. We note that the proposed A is by no means an optimal A . As far as we know there is no analytic design procedure to derive the optimal A as stated in [13]. It is possible that extensive tuning of the parameters could produce better results and to derive a better tuning method is one of future topics.

If the output $Y_k = 1$, then we can assume $X_k = 1$ from Figure 7. Also if $Y_k = 2$, then X_k is either 2 or 4. If $Y_k = 3$, then X_k is 3. This observation is contained in a matrix C , which represents the link between the output process Y_k and the state process X_k . Matrix C is defined by:

$$C_{mn} = \Pr[Y_k = m | X_k = n], \quad \sum_{m=1}^3 C_{mn} = 1.$$

In this paper, we use the following C matrix:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3)$$

Once A and C are defined, we can estimate X_k from the output Y_k using HMM filters or smoothers [11-14]. Let $\Pi_{k|k+l} \in R^{4 \times 1}$ be a probability vector whose i -th element is given by $\Pr[X_k = i | Y_1, \dots, Y_{k+l}]$. If $l = 0$, then the problem becomes an HMM filter. If $l > 0$, the problem becomes an HMM (fixed-lag) smoother problem. It is known that an HMM smoother gives a more reliable result than an HMM filter [11]. We found an HMM smoother with $l = 1$ gives a significantly much more reliable results in our case (see the experimental results in Table 1). For example, suppose the state 4 and 1 are missing and the state transitions are given by $1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 3$ (X_k). In the case, the outputs are given by $1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 3$ (Y_k). If we use an HMM filter, the estimated first four states are likely to be $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ (X_k) since it only uses the current outputs $1 \rightarrow 2 \rightarrow 3 \rightarrow 2$ (Y_k). On the other hand, an HMM smoother gives the correct estimates using the outputs $1 \rightarrow 2 \rightarrow 3 \rightarrow 2 \rightarrow 3$ (Y_k). When an HMM smoother with $l = 1$, we can obtain an estimation result only after the current and the next outputs are available. Thus the estimation result is delayed by the combined duration of the current and next segments: typical delay is 1 second.

4. Zero Velocity Detection Algorithm

After X_k is estimated, the next step is to determine zero velocity intervals. We decide that the time i belongs to zero velocity intervals if $|z_i| \leq \alpha_3$ with anti-chattering algorithm and i satisfies the following conditions:

$$\begin{aligned} \text{case 1 : if } X_{k-1} = 1, X_k = 2 \\ s_{k-1} + \beta_1 \leq i \leq e_{k-1} - \gamma_1 \\ \text{case 2 : if } X_{k-1} = 4, X_k = 2 \\ s_{k-1} + \beta_2 \leq i \leq s_k - \gamma_2 \\ \text{case 3 : if } X_{k-1} = 3, X_k = 2 \\ s_{k-1} + \beta_3 \leq i \leq s_k - \gamma_3. \end{aligned}$$

The $|z_i| \leq \alpha_3$ condition is used as a safeguard to prevent false identification of the zero velocity interval.

Case 1 corresponds to the normal walking cycles, where the state transition is $3 \rightarrow 4 \rightarrow 1 \rightarrow 2$ (see Figure 7). Since the zero velocity interval belongs to the state 1, the interval is chosen inside state 1. The parameters β_1 and γ_1 are used to discard time intervals near edges of the interval whose state is 1:

$$\beta_1 = \gamma_1 = 0.1(e_{k-1} - s_{k-1}).$$

Since a foot is moving before and after state 1, the foot velocity cannot be zero in the edge regions. Thus we cut 10% (corresponding to 0.1) off the interval at both edges. If we choose larger values (for example, 0.2), the chance that a non-zero velocity interval is mistakenly identified as a zero velocity interval is reduced. However, we could obtain narrower zero velocity intervals, which is not desirable.

Case 2 and 3 correspond to walking or running cycles, where state 1 and/or state 4 are missing. Case 2 is when state 1 is missing. This could happen if the zero velocity interval is too short to form a segment. For example, state 1 is missing in Figure 9: state transition is $2 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 3$. However, we can identify an interval corresponding to state 1 in Figure 9 by the manual inspection. This interval is identified as state 1 in the manual inspection. Thus the state classifications by the manual inspection and by X_k are different: even if some states are missing in X_k transition, they can be identified in the manual inspection.

We manually inspected the data and found the average duration ratio of each state is given as follows:

$$(\text{state 1}, \text{state 2}, \text{state 3}, \text{state 4}) = (11.3\%, 27.4\%, 42.8\%, 18.5\%).$$

The ratio was computed from a weighted average of walking (weighting 0.1) and running (weighting 0.9) data. More weighting was given to running since missing states (like in case 2 and 3) are more likely to happen in running.

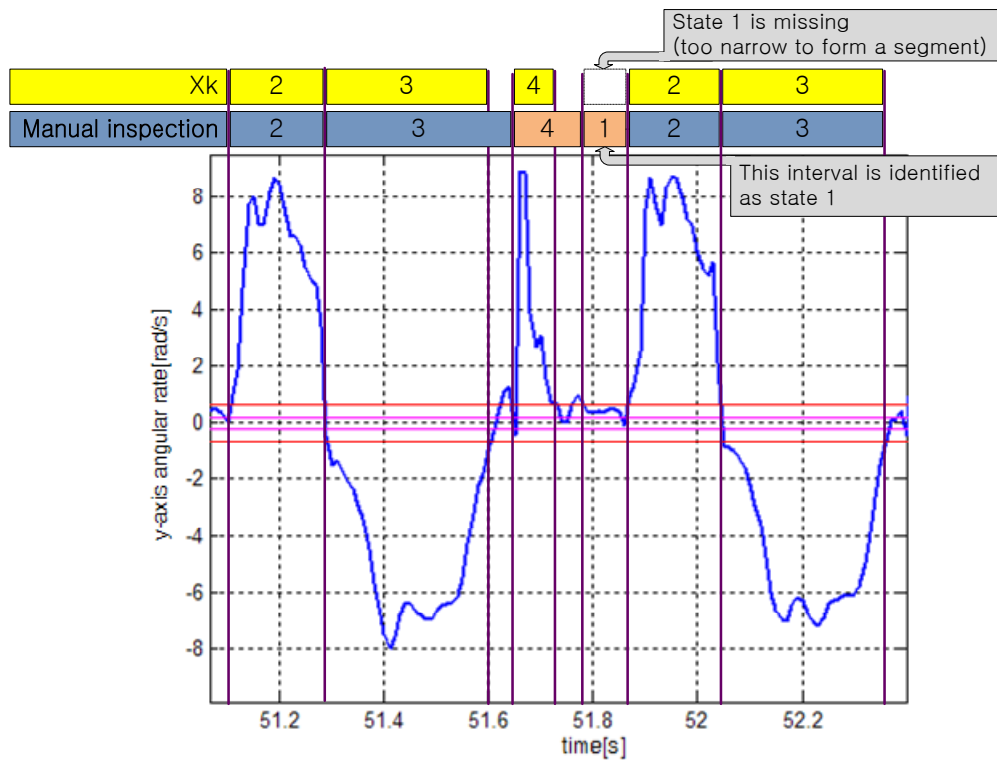
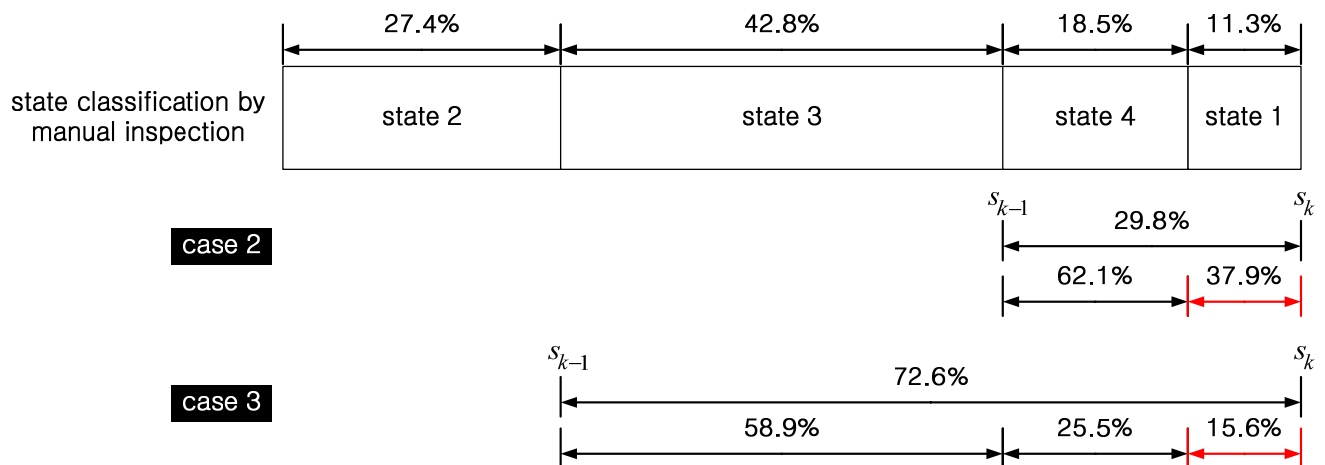
In case 2, the state 1 is missing. We assume the interval of missing state 1 is $(s_{k-1} + b_k, s_k)$. Note that the average duration percentage of state 4 and state 1 are 18.5% and 11.3%, respectively. Thus the duration ratio between s_{k-1} and s_k is 0.621:0.379. Thus we choose β_2 and γ_2 as follows (see Figure 10):

$$\beta_2 = 0.621(s_k - s_{k-1}), \gamma_2 = 0$$

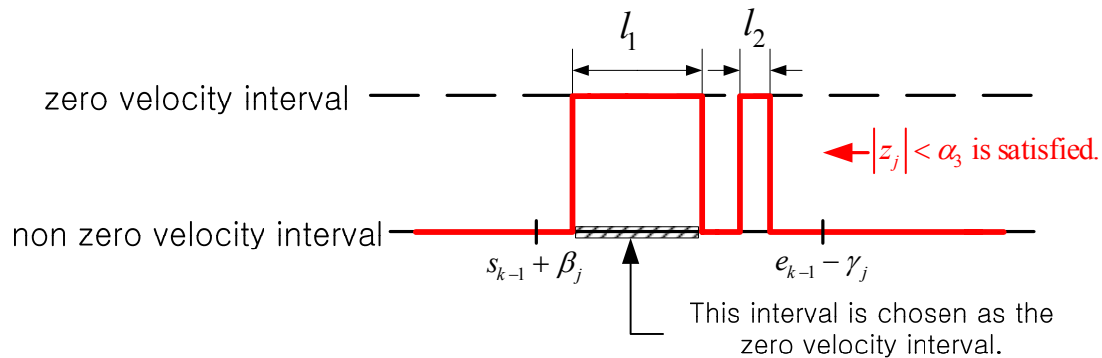
In other words, the interval $[s_{k-1}, s_k]$ is divided into $[s_{k-1}, s_{k-1} + \beta_2]$ and $[s_{k-1} + \beta_2, s_k]$, where the latter is considered as state 1 interval.

Case 3 is when state 4 and 1 are missing. In this case, interval $[s_{k-1}, s_k]$ includes the state 3 and missing states (4 and 1). Note that the average duration percentage of state 3, 4, and state 1 are 42.8%, 18.5%, and 11.3%, respectively. Thus the duration ratio among state 3:state 4:state 1 is 0.589 : 0.255 : 0.156. Thus we choose β_3 and γ_3 as follows:

$$\beta_3 = 0.844(s_k - s_{k-1}), \gamma_3 = 0$$

Figure 9. Case 2: state 1 is missing.**Figure 10.** Average duration ratio of each state by manual inspection and its use in Case 2 and 3.

When the interval $[s_{k-1} + \beta_j, e_{k-1} - \gamma_j]$ is determined, we checked whether z_i , $i \in [s_{k-1} + \beta_j, e_{k-1} - \gamma_j]$ satisfies $|z_i| < \alpha_3$ condition. Let $U = \{i | |z_i| < \alpha_3, i \in [s_{k-1} + \beta_j, e_{k-1} - \gamma_j]\}$. If the set U consists of more than one intervals as in Figure 11, the interval whose duration is the largest is chosen as the zero velocity interval.

Figure 11. Zero velocity interval selection.

5. Experiments

In this section, the proposed method is verified using experiment data, which are obtained from an inertial sensor unit (Xsens Mti) attached on a shoe. The sampling rate is 100 Hz. In Figure 12, sensor outputs are given when a person walked and then ran slowly. We note that only gyroscope output is used in the proposed algorithm. The z-axis accelerometer output is also given since it is used by other existing algorithms. The proposed algorithm along with other existing algorithms [1-3] are applied to the experiment data. The parameters of the proposed algorithm are given by:

$$\alpha_1 = \alpha_2 = \alpha_3 = 0.7, N_1 = N_2 = 10, N_3 = 20, l = 1$$

The parameters α_j and N_j are determined from the experimental data so that normal walking gives four different states. For example, if we use a smaller α_2 value, it becomes sensitive to noises so that states 2/3/4 could be falsely identified. On the other hand, if we use larger α_2 value, states 2/3/4 could be lost. The parameter N_j is closely related to α_j . For example, smaller N_2 has a similar effect of using smaller α_2 in the sense that state 2/3/4 could be falsely identified.

As a reference, zero velocity interval is determined by manually analyzing gyroscope data. When data are analyzed manually from the plot, the sensor noises and irregularities can be easily spotted and thus we can consider the detected interval as a true zero velocity interval.

The result is given in Figure 13. The algorithm in [1] failed to detect the zero velocity interval when a person is running. The algorithm in [3] is not reliable in the sense that there are false zero velocity detections.

In Figure 14, sensor outputs are given when a person ran slowly and then ran fast. The speeds of “walk”, “run slowly” and “run fast” are about 1.05 m/s, 2.14 m/s, and 3.50 m/s, respectively, which are computed by dividing the distance by the elapsed time. The zero detection results are given in Figure 15. It can be seen that the methods in [1] and [2] cannot detect the zero velocity intervals when a person ran fast. In all cases (Figures 13 and 15), the proposed algorithm reliably detected the zero velocity intervals.

Figure 12. Sensor outputs when a person is walking(23~27 s) and slowly running(27~32 s).

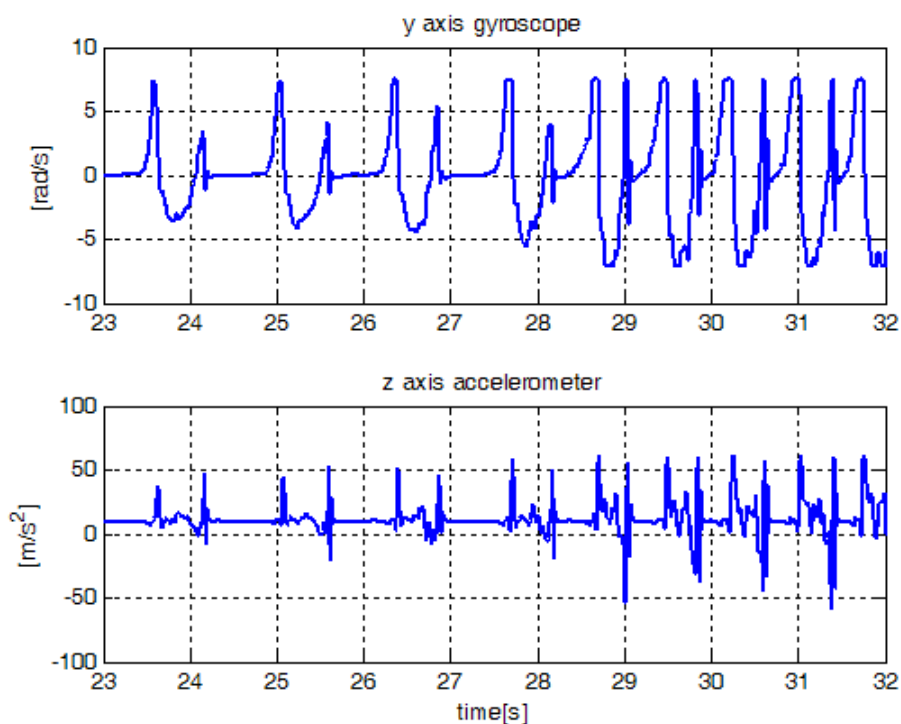


Figure 13. Zero velocity detection comparison when a person is walking and slowly running (Z. V. I. means zero velocity interval).

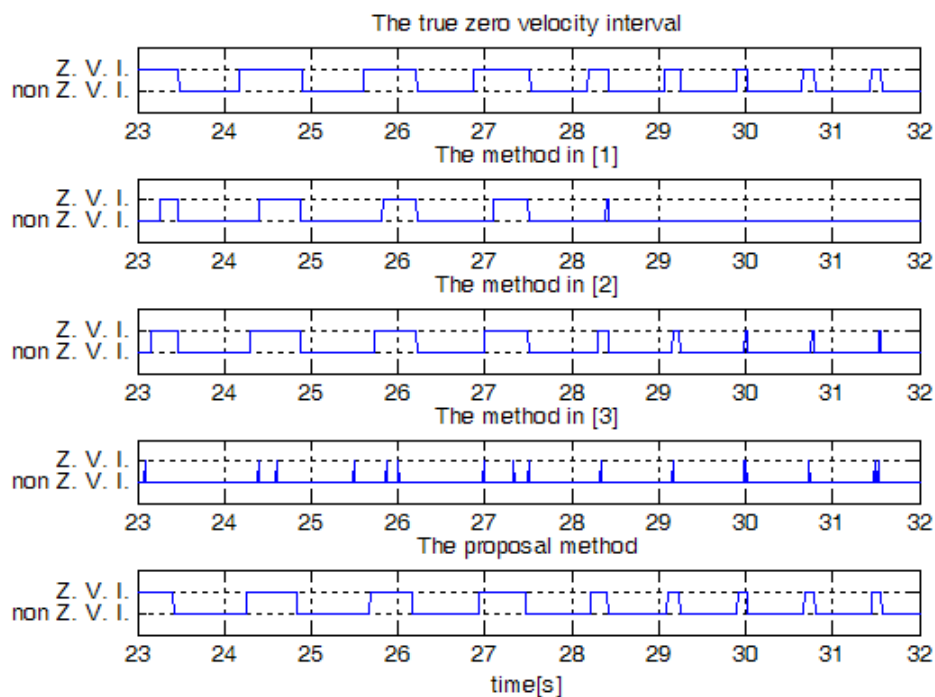


Figure 14. Sensor outputs when a person is running slowly(25~30s) and running fast(30~34 s).

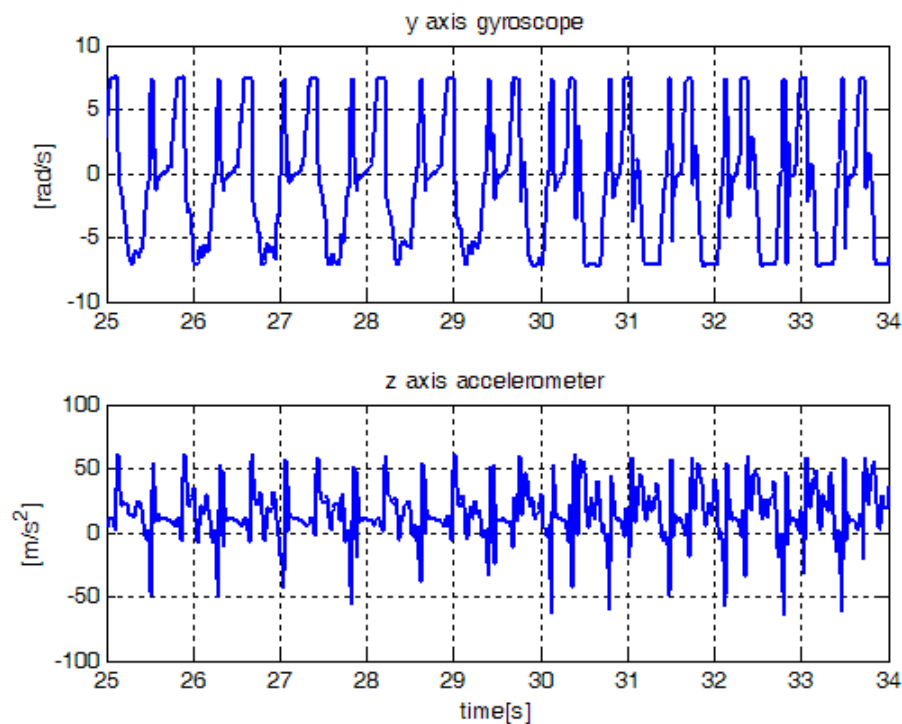
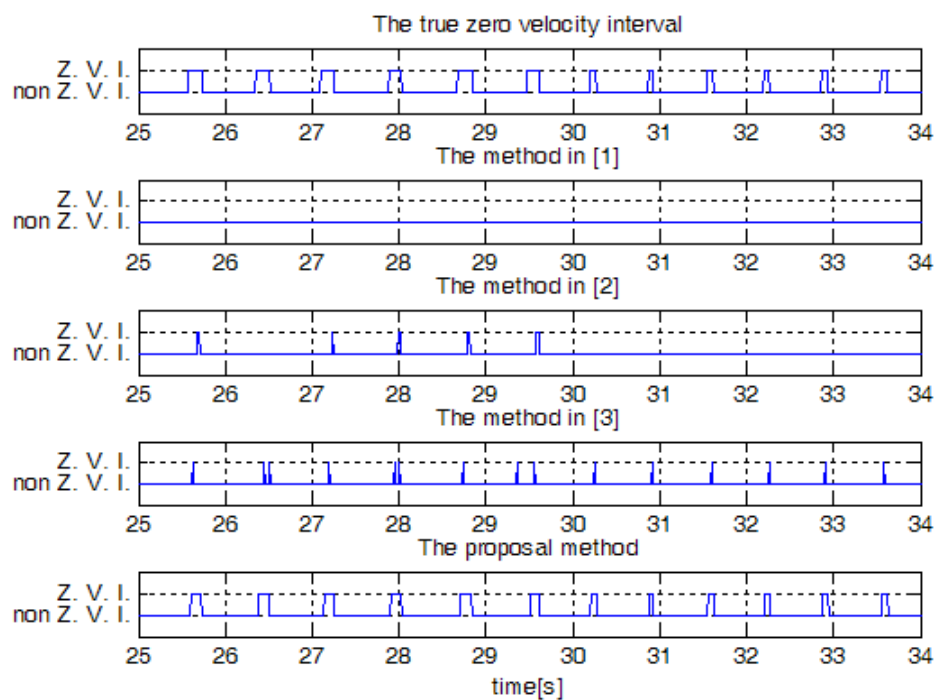


Figure 15. Zero velocity detection comparison when a person is running slowly and running fast(Z. V. I. means zero velocity interval).



To compare performance of each method quantitatively, a person walked or ran 30 steps at different speed. Since there is one zero velocity interval in each step, 30 zero velocity intervals should be detected. The number of detected zero velocity intervals by different methods is given in Table 2. We can see when a person is walking, all methods give a good result. When a person is running, method 1 and 2 failed to detect the zero velocity intervals most times. The method [3] and the proposed method with an HMM smoother are able to detect all the zero velocity intervals. When counting the number of detected zero velocity intervals, we counted one even if there are more than two detection during the same zero velocity interval. For example, see the two detected zero velocity intervals by the method [3] between 24 and 25 s. These two intervals are counted as one interval. We note that there is a zero velocity interval between the two detected zero velocity intervals, but it is not detected by the method [3]. On the other hand, the proposed method detects almost the whole range of a zero velocity interval. Thus the proposed method can detect the zero velocity intervals more reliably.

Table 2. Number of detected zero velocity intervals by different methods (actual number of zero velocity interval is 30).

	Experiments	Speed [m/s]	Number of detected zero velocity intervals by different methods				
			[1]	[2]	[3]	Proposed method with an HMM filter	Proposed method with an HMM smoother
Walking	1st	1.3	28	30	30	30	30
	2nd	1.4	22	29	30	30	30
	3rd	1.4	28	30	30	30	30
Running	1st	2.4	0	2	30	10	30
	2nd	2.8	0	1	30	1	30
	3rd	2.5	0	0	30	16	30

We also investigated whether there is false zero velocity detection (an interval is identified as a zero velocity interval while a foot is not in the zero velocity interval). The result is given in Table 3, where the same experiment data were used as in Table 2. It can be seen that the method in [3] sometimes falsely identifies the zero velocity intervals: we found the method in [3] sometimes falsely identifies zero crossing points of gyroscope norm as zero velocity intervals. There was no false zero velocity detection in other methods.

Table 3. Number of falsely detected zero velocity intervals by different methods.

	Experiments	Speed [m/s]	Number of detected non zero velocity intervals by different methods				
			[1]	[2]	[3]	Proposed method with an HMM filter	Proposed method with an HMM smoother
Walking	1st	1.3	0	0	5	0	0
	2nd	1.4	0	0	1	0	0
	3rd	1.4	0	0	1	0	0
Running	1st	2.4	0	0	0	0	0
	2nd	2.8	0	0	0	0	0
	3rd	2.5	0	0	3	0	0

6. Conclusions

In this paper, we have proposed a new zero velocity detection algorithm, which can be used in inertial sensor based pedestrian navigation systems.

Existing zero detection methods usually work well for normal walking cycles, but the detection is not reliable for running cycles, in particular when the running speed is high. Using a hidden Markov model, the proposed algorithm can analyze both walking and running cycles and the zero velocity interval is thus more reliably detected.

The actual state transition probabilities differ depending on speed of a person. In this paper, we used one averaged transition probability. It is a future work to derive a zero velocity detection algorithm that uses multiple state transition probability matrices.

Acknowledgments

This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (No. 2010-0016230).

References

1. Foxlin, E. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Comp. Graph. Appl.* **2005**, *25*, 38-46.
2. Yun, X.; Bachmann, E.R.; IV, H.M.; Calusidan, J. Self-contained position tracking of human movement using small inertial/magnetic sensor modules. In *Proceedings of IEEE International Conference on Robotics and Automation*, Rome, Italy, April 10–14, 2007; pp. 2526-2533.
3. Ojeda, L.; Borenstein, J. Non-GPS navigation for security personnel and first responders. *J. Navigation* **2007**, *60*, 391-407.
4. Suh, Y.S.; Park, S.K. Pedestrian inertial navigation with gait phase detection assisted zero velocity updating. In *Proceedings of the 4th International Conference on Autonomous Robots and Agents*, Wellington, New Zealand, February 10–12, 2009; pp. 336-341.
5. Stirling, R.G. Development of A Pedestrian Navigation System Using Shoe Mounted Sensors. Master of Science, University of Alberta, Edmonton, AB, Canada, 2003.
6. Jiménez, A.R.; Seco, F.; Prieto, C.; Guevara, J. Indoor Pedestrian Navigation using an INS/EKF framework for Yaw Drift Reduction and a Foot-mounted IMU. In *Proceedings of 7th Workshop on Positioning, Navigation and Communication, WPNC 2010*, Dresden, Germany; March 11–12, 2010.
7. Millonig, A.; Schechtner, K. Developing landmark-based pedestrian navigation systems. In *Proceedings of IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria, September 13–16, 2005; pp. 197-202.
8. Wendlandt, K.; Berbig, M.; Robertson, P. Indoor localization with probability density functions based on Bluetooth. In *Proceedings of the 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, Berlin, Germany, September 2005; pp. 2040-2044.
9. Chatfield, A.B. *Fundamentals of High Accuracy Inertial Navigation*; AIAA: Richmond, VA, USA, 1997.

10. Yang, C.K.; Shim, D.S. Best sensor configuration and accommodation rule based on navigation performance for INS with seven inertial sensors. *Sensors* **2009**, *9*, 8456-8472.
11. Shue, L.; Andeson, B.D.O.; Dey, S. Exponential stability of filters and smoothers for Hidden Markov models. *IEEE Trans. Acoustics, Speech, Signal Proc.* **1998**, *46*, 2180-2194.
12. Anderson, B.D.O. From Winer to hidden Markov models. *IEEE Control Syst. Mag.* **1999**, *19*, 41-51.
13. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *IEEE* **1989**, *77*, 257-286.
14. Seo, D.J.; Ko, N.Y.; Son, J.E. A method for combining odometry and distance sensor information for Effective obstacle avoidance of autonomous mobile robots. *Int. J. Cont. Auto. Syst.* **2010**, *8*, 597-603.

© 2010 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 license (<http://creativecommons.org/licenses/by/3.0/>).