Tightly Integrated Motion Classification and State Estimation in Foot-Mounted Navigation Systems

Skog, Isaac; Hendeby, Gustaf; Kok, Manon

DOI
10.1109/IPIN57070.2023.10332538

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 13th International Conference on Indoor Positioning and Indoor Navigation, IPIN 2023

Citation (APA)
Skog, I., Hendeby, G., & Kok, M. (2023). Tightly Integrated Motion Classification and State Estimation in Foot-Mounted Navigation Systems. In *Proceedings of the 2023 13th International Conference on Indoor Positioning and Indoor Navigation, IPIN 2023* IEEE. https://doi.org/10.1109/IPIN57070.2023.10332538

2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN)

Tightly Integrated Motion Classification and State Estimation in Foot-Mounted Navigation Systems

Isaac Skog
Dept. of Electrical Engineering
Uppsala University
Uppsala, Sweden
isaac.skog@angstrom.uu.se

Gustaf Hendeby
Dept. of Electrical Engineering
Linköping University
Linköping, Sweden
gustaf.hendeby@liu.se

Manon Kok
Delft Center for Systems and Control
Delft University of Technology
Delft, The Netherlands
m.kok-1@tudelft.nl

2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN) | 979-8-3503-2011-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/IPIN57070.2023.10332538

***Abstract*—A framework for tightly integrated motion mode classification and state estimation in motion-constrained inertial navigation systems is presented. The framework uses a jump Markov model to describe the navigation system's motion mode and navigation state dynamics with a single model. A bank of Kalman filters is then used for joint inference of the navigation state and the motion mode. A method for learning unknown parameters in the jump Markov model, such as the motion mode transition probabilities, is also presented. The application of the proposed framework is illustrated via two examples. The first example is a foot-mounted navigation system that adapts its behavior to different gait speeds. The second example is a foot-mounted navigation system that detects when the user walks on flat ground and locks the vertical position estimate accordingly. Both examples show that the proposed framework provides significantly better position accuracy than a standard zero-velocity aided inertial navigation system. More importantly, the examples show that the proposed framework provides a theoretically well-grounded approach for developing new motion-constrained inertial navigation systems that can learn different motion patterns.**

***Index Terms*—Inertial navigation, Zero-velocity detection, Constant height detection, Filter bank, Motion-constraints.**

I. Introduction

Current state-of-the-art technology for zero-velocity aided inertial navigation systems is based upon a strategy of loose integration between the zero-velocity detector and the inertial navigation filter [1]. That is, the zero-velocity detector and the inertial navigation filter are treated as separate functions, with a one-directional flow of information from the former to the latter. From an information theoretical perspective this is suboptimal because: (a) the estimated navigation states carry information about the system's motion mode that is not used in the zero-velocity detector, and (b) the test statistic used in the zero-velocity detector is quantized before it is used in the navigation filter. Thus, information that could be used both to improve the detection of zero-velocity events and to control how the zero-velocity updates are performed, is lost. The same argumentation also applies to other motion constraints commonly applied to foot-mounted inertial navigation systems and where an external motion classifier is used to determine when to apply these constraints. A few examples of commonly used motion constraints are that the system keeps a constant height [2], constant heading [3], or constant speed [4].

The fact that the estimated navigation state carries information that can be used to improve the zero-velocity detection process is utilized in [5], where the velocity estimates from the inertial navigation filter are used as a prior for a Bayesian zero-velocity detector. And in [6], the test statistics from the zero-velocity detector are used to control the magnitude of the measurement covariance in the inertial navigation filter. Thereby, quantization of the zero-velocity test statistics is avoided and the zero-velocity update process can adapt to different gait conditions. Still, both [5] and [6] employ a loose integration strategy where the zero-velocity detector and the inertial navigation filter are treated as separate functions.

This paper instead proposes a framework for tightly integrated motion mode classification and navigation state estimation, of which tightly integrated zero-velocity aided inertial navigation is a special case. The core of the framework is a jump Markov model that includes both the navigation state and the motion mode. That is, a single model is used to describe both the kinematics of the system under various motion modes and the probability of transitioning between the motion modes. A bank of Kalman filters is then used to jointly estimate the navigation state and the motion mode. A method to automatically learn unknown parameters in the jump Markov model is also presented. The proposed framework is evaluated on two data sets consisting of various gait conditions and motion modes.

Reproducible research: The data and code used to produce the presented results can be downloaded at:
`https://gitlab.liu.se/open-shoe/filterbanks`.

II. Signal Model

Let the system state x_k and input vector u_k at time instant k be defined as

$$x_k = \begin{bmatrix} r_k \\ v_k \\ q_k \\ \xi_k \end{bmatrix} \quad \text{and} \quad u_k = \begin{bmatrix} s_k \\ \omega_k \end{bmatrix}, \tag{1}$$

respectively. Here r_k, v_k, and q_k denote the position, velocity, and attitude quaternion, respectively. Further, ξ_k denotes potential auxiliary states, such as sensor biases, needed to describe the behavior of the system. Moreover, s_k and ω_k denote the specific force and angular velocity, respectively.

Next, define the discrete state

$$\delta_k \in \{1, \dots, L\} \tag{2}$$

that indicates the current motion mode of the navigation system, e.g., the system is stationary, keeping a constant height, or moving at constant velocity. A jump Markov nonlinear model that can be used to the describe the dynamics of the inertial navigation system is then given by [7]

$$\begin{aligned} x_{k+1} &= f(x_k, \delta_k, u_k, \eta_k) && \text{(3a)}\\ y_k &= h(x_k, \delta_k, u_k) + e_k && \text{(3b)}\\ \eta_k &\overset{\text{i.i.d.}}{\sim} \mathcal{N}\big(n_k; 0, Q_k\big) && \text{(3c)}\\ e_k &\overset{\text{i.i.d.}}{\sim} \mathcal{N}\big(e_k; 0, R(\delta_k)\big) && \text{(3d)}\\ p(\delta_{k+1}|\delta_k) &= \Pi_{\delta_{k+1},\delta_k}. && \text{(3e)} \end{aligned}$$

Here $y_k \triangleq 0$ is used as a pseudo-observation to impose a set of motion mode dependent "stochastic constraints", such as zero-velocity constraints, on the navigation state x_k. The exact form of these constraints, defined by $h(\cdot)$, depends on the assumed motion modes and will be discussed later. The "hardness" of the imposed constraints is controlled via the motion mode dependent covariance $R(\delta_k)$. Further, $\mathcal{N}\big(\,\cdot\,; \mu, \Sigma\big)$ denotes a multivariate normal distribution parameterized by the mean vector μ and covariance matrix Σ. The function $f(\cdot)$ describes the system dynamics and is given by the inertial navigation equations and the dynamics of the auxiliary states ξ_k. Moreover, Q_k denotes the covariance of the process noise, and $\Pi_{i,j}$ denotes the i:th row and j:th column entry of the mode transition probability matrix. Finally, $p(a|b)$ denotes the probability density (mass) function of a given b. Note that it is straightforward to extend the model (3) to also include real observations from various sensors, but it is out of the scope of this paper, as the focus is on motion-constrained inertial navigation.

III. State Estimation Using a Filter Bank

A variety of inference techniques for jump Markov models exist. Here a commonly used filter banks solution to the inference problem will be recapitulated; for details the reader is referred to [8] and [9]. Thereafter, necessary approximations needed to adapt the filter bank solution to the considered navigation problem will be presented.

A. Linear model with normal distributed noise

The goal of the inference process is to estimate the a posteriori distribution $p(x_k|y_{1:k})$ of the state x_k given all the observations up until time k, denoted as $y_{1:k}$. If the motion mode sequence $\delta_{1:k}$ is known, the state-space model in the Markov jump system is linear, and the process and measurement noise are normally distributed, then the a posteriori distribution can be estimated with the Kalman filter. That is,

$$p(x_k|\delta_{1:k}, y_{1:k}) = \mathcal{N}\big(x_k; \hat{x}_{k|k}^{\delta_{1:k}}, P_{k|k}^{\delta_{1:k}}\big), \tag{4}$$

where $\hat{x}_{k|k}^{\delta_{1:k}}$ and $P_{k|k}^{\delta_{1:k}}$ denote the Kalman filter state estimate and state covariance given the mode sequence $\delta_{1:k}$, respectively. In reality, the motion mode sequence is unknown and must be estimated from the measurements. Let $p(\delta_{1:k}|y_{1:k})$ denote the a posteriori distribution of the motion mode sequences $\delta_{1:k}$ given the measurements $y_{1:k}$. The sought-after a posteriori distribution of the state x_k can then be found as

$$\begin{aligned} p(x_k|y_{1:k}) &= \sum_{i=1}^{L^k} p(x_k|\delta_{1:k}^i, y_{1:k}) p(\delta_{1:k}^i|y_{1:k}) \\ &= \sum_{i=1}^{L^k} w_k^i \mathcal{N}\big(x_k; \hat{x}_{k|k}^{\delta_{1:k}^i}, P_{k|k}^{\delta_{1:k}^i}\big), \end{aligned} \tag{5}$$

where $w_k^i \triangleq p(\delta_{1:k}^i|y_{1:k})$ is probability of the i:th motion mode sequence. Hence, the posteriori distribution of the state x_k is given by a mixture of L^k weighted normal distributions where the expected value and covariance of each mixture component are calculated via a Kalman filter. That is, the posteriori distribution can be calculated via a bank of Kalman filters where the size of the filter bank grows according to a tree structure. The weights w_k^i in the mixture, i.e., the probability for each branch in the tree, can be recursively calculated as

$$w_k^i \propto p(y_k|y_{1:k-1}, \delta_{1:k}^i) p(\delta_k^i|\delta_{k-1}^i) w_{k-1}^i, \tag{6a}$$

where

$$\sum_{i=1}^{L^k} w_k^i = 1, \tag{6b}$$

and

$$p(y_k|y_{1:k-1}, \delta_{1:k}^i) = \mathcal{N}\big(y_k; \hat{y}_{k|k-1}^{\delta_{1:k}^i}, S_k^{\delta_{1:k}^i}\big). \tag{6c}$$

Here $\hat{y}_{k|k-1}^{\delta_{1:k}}$ and $S_k^{\delta_{1:k}}$ denote the Kalman filter measurement prediction and innovation covariance, respectively.

From the posteriori distribution, the minimum variance estimate of the state x_k can be calculated as

$$\hat{x}_k^{\text{mv}} = \sum_{i=1}^{L^k} w_k^i \hat{x}_{k|k}^{\delta_{1:k}^i} \tag{7a}$$

and the associated conditional covariance matrix as

$$P_k^{\text{mv}} = \sum_{i=1}^{L^k} w_k^i \big(P_{k|k}^{\delta_{1:k}^i} + (\hat{x}_{k|k}^{\delta_{1:k}^i} - \hat{x}_k^{\text{mv}})(\hat{x}_{k|k}^{\delta_{1:k}^i}) - \hat{x}_k^{\text{mv}})^\top\big). \tag{7b}$$

B. Approximative solution

The outlined filter bank solution to the inference problem can in general not be used without modifications due to the growing number of Kalman filters needed. Therefore, many strategies have been developed for pruning and merging branches in the growing filter bank tree so that a fixed complexity is achieved [10].

Beyond the challenges with the exponentially increasing complexity, the outlined general solution cannot still be applied to the Markov jump system model in (3). The system is nonlinear and the attitude states belong to a manifold of Euclidean space. This implies that $p(x_k|\delta_{1:k}, y_{1:k})$ in (4) is generally not a normal distribution, neither can the minimum variance estimate of the attitude q_k be calculated as in (7a). A common way to handle that the system is nonlinear and the attitude not belonging to Euclidian space in the filtering process is to use an error state Kalman filter [11]. The a posteriori distribution $p(x_k|\delta_{1:k}, y_{1:k})$ is then, at every time instant, approximated as normal distributed with the mean and covariance given by the error state Kalman filter [12].

One way to calculate a point estimate of the attitude in terms of the Euler angles ϕ is via

$$\hat{\phi}_k^{\text{mv}} = \underset{\phi \in \Omega_\phi}{\arg\min} \sum_i^{L^k} w_k^i \|C(\hat{q}_{k|k}^{\delta_{1:k}^i}) - C(\phi)\|_F^2, \tag{8}$$

where $C(q)$ denotes the rotation matrix that transforms a vector from the body frame to the navigation frame [13]; to simplify the notation, a slight misusage of notation is admitted, and the rotation matrix is here interchangeably parameterized by the attitude quaternion q and the corresponding Euler angles ϕ. Further, the set $\Omega_\phi = [0, 2\pi) \times [-\pi/2, \pi/2) \times [0, 2\pi)$. The covariance of $\hat{\phi}_k^{\text{mv}}$ can be approximately calculated as

$$\text{Cov}(\hat{\phi}_k^{\text{mv}}) \approx \sum_i^{L^k} w_k^i \big(\Sigma_{k|k}^{\delta_{1:k}^i} + \Delta\phi_k^{\delta_{1:k}^i} (\Delta\phi_k^{\delta_{1:k}^i})^\top\big). \tag{9}$$

Here $\Sigma_{k|k}^{\delta_{1:k}^i}$ denotes the covariance of the attitude (in terms of Euler angles) calculated by the error state Kalman filter for the branch corresponding to the motion sequence $\delta_{1:k}^i$. Further, the difference between the attitude estimate calculated by the error state Kalman filter and the minimum variance attitude estimate in (8) is given by

$$\Delta\phi_{k|k}^{\delta_{1:k}^i} = \{\phi \in \Omega_\phi \text{ s.t. } C(\hat{\phi}_k^{\text{mv}}) = C(\phi) C(\hat{q}_{k|k}^{\delta_{1:k}^i})\}. \tag{10}$$

Note that the optimization problem in (8) can be efficiently solved using a polar decomposition [13].

IV. Learning of Unknown Model Parameters

The signal model in (3) may have unknown parameters, such as the transition probability matrix Π, whose values can be hard to specify accurately. Instead of resorting to cumbersome hand-tuning of these parameters, they may be learned from data. A variety of methods to learn model parameters in jump Markov models has been suggested [14], [15]. Here a maximum likelihood method, similar to that presented in [12], for learning the parameters from the data will be outlined.

Let θ denote the unknown parameters in the model. The maximum likelihood estimate of these parameters is then given by

$$\hat{\theta} = \arg\max_\theta p(y_{1:k}; \theta), \tag{11a}$$

where

$$p(y_{1:k}; \theta) = \prod_{n=2}^{k} p(y_n|y_{1:n-1}) p(y_1), \tag{11b}$$

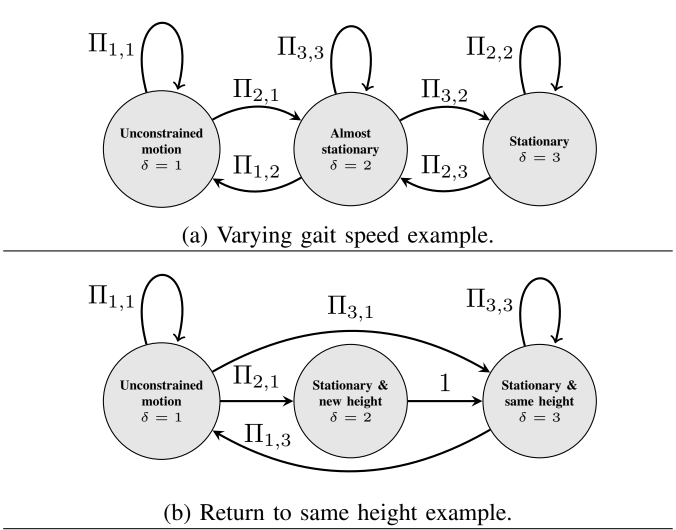


Fig. 1: Mode transition diagrams for the application examples.

and

$$p(y_n|y_{1:n-1}) = \sum_{i=1}^{L^n} \mathcal{N}\big(y_n; \hat{y}_{n|n-1}^{\delta_{1:n}^i}, S_n^{\delta_{1:n}^i}\big) p(\delta_n^i|\delta_{n-1}^i) w_{n-1}^i. \tag{11c}$$

Here $p(y_1)$ denotes the a priori probability of the observation y_1. Note that the number of modes in the full mixture distribution, which subsequently must be summed to marginalize away the mode dependence, increases exponentially in n. However, in practice, the actual number of modes to consider is reduced using pruning or merging in the filter bank. Hence, the number of modes to consider when evaluating (11c) can be kept tractable.

V. Application Examples

Next, the proposed inference framework is used to realize two foot-mounted inertial navigation systems that incorporate different motion modes. The first system adaptively selects the covariance matrix used in the zero-velocity update, and the second system automatically detects if the system returns to the same height after a step. For both systems, the motion mode state transition matrix is learned from training data using the method outlined in Sec. IV. The performances of both systems are compared to the OpenShoe system presented in [16].

A. Example: Varying gait speed

The challenge of designing and tuning a foot-mounted zero-velocity aided inertial navigation system so that it works well for multiple gait speeds is well-known [1]. To use the proposed framework to design a system that adaptively selects the detection threshold, as well as the covariance matrix used in the zero-velocity updates, we define the following three motion modes: the unconstrained motion mode ($\delta_k = 1$), the almost stationary motion mode ($\delta_k = 2$), and the stationary motion mode ($\delta_k = 3$). A motion mode transition diagram illustrating how the system is assumed to transition between these motion modes is shown in Fig. 1a. The associated motion mode transition probabilities $\Pi_{i,j}$ are also shown.

Next, introduce the following system dynamics

$$f(x_k, \delta_k, u_k, \eta_k) = \begin{bmatrix} r_k + \Delta t v_k \\ v_k + \Delta t\big(C(q_k)(s_k + \eta_k^s) + g\big) \\ \Omega\big(\Delta t\,(\omega_k + \eta_k^\omega)\big) q_k \end{bmatrix}. \tag{12a}$$

Here Δt and g denote the sampling period and gravity vector, respectively. Further, $\Omega(\cdot)$ denotes the quaternion update matrix (see [17] for details). Moreover, η_k^s and η_k^ω denote the process noise associated with the accelerometers and gyroscopes, respectively.

Finally, define the stochastic motion constraints imposed during the different motion modes as follows

$$h(x_k, u_k, \delta_k) = \begin{cases} 0 \quad , & \delta_k = 1 \\ h_0(x_k, u_k), & \delta_k = \{2,3\} \end{cases}, \tag{12b}$$

where

$$h_0(x_k, u_k) = \begin{bmatrix} v_k \\ \omega_k \\ C(q_k)s_k + g \end{bmatrix}. \tag{12c}$$

The "hardness" of the constraints is controlled by the covariance matrix

$$R(\delta_k) = \begin{cases} \sigma_{nc}^2 I, & \delta_k = 1 \\ R_0(\delta_k), & \delta_k = \{2,3\} \end{cases}, \tag{12d}$$

where

$$R_0(\delta_k) = \sigma_v^2(\delta_k) I \oplus \sigma_\omega^2(\delta_k) I \oplus \sigma_s^2(\delta_k) I. \tag{12e}$$

Here $\oplus$ denotes the direct sum matrix operator and I denotes an identity matrix of appropriate size. Further, $\sigma_v^2(\delta_k)$, $\sigma_\omega^2(\delta_k)$, and $\sigma_s^2(\delta_k)$ denote the mode-dependent variance associated with the stochastic constraints on the velocity, angular velocity, and acceleration, respectively. Moreover, σ_{nc}^2 is a design parameter that controls the measurement likelihood associated with no constraints.

A filter bank with the jump Markov model defined by (12) was designed and two data sets were recorded with a sensor array consisting of 32 InvenSense MPU9150 IMUs while a person walked and ran back and forth along a straight line. From the recorded data, two data sets with 50 measurement sequences each were created by repeatedly drawing four random IMUs and averaging their measurements. These measurement sequences were then processed using the OpenShoe system algorithm with the zero-velocity detector threshold tuned to generate a detection at least once per gait cycle. The measurements were also processed using the designed filter bank. Pruning was used to keep the tree size to a maximum of nine leaves, and a pruning strategy where the most probable leaves were retained was used.

The measurement sequences in the first data set were used to learn the mode transition matrix Π. The initial value and end result of the learning process were

$$\widehat{\Pi}^{\text{init}} = \begin{bmatrix} 1/2 & 1/3 & 0 \\ 1/2 & 1/3 & 1/2 \\ 0 & 1/3 & 1/2 \end{bmatrix} \text{ and } \widehat{\Pi}^{\text{end}} = \begin{bmatrix} 0.993 & 0.073 & 0 \\ 0.007 & 0.893 & 0.005 \\ 0 & 0.034 & 0.995 \end{bmatrix},$$

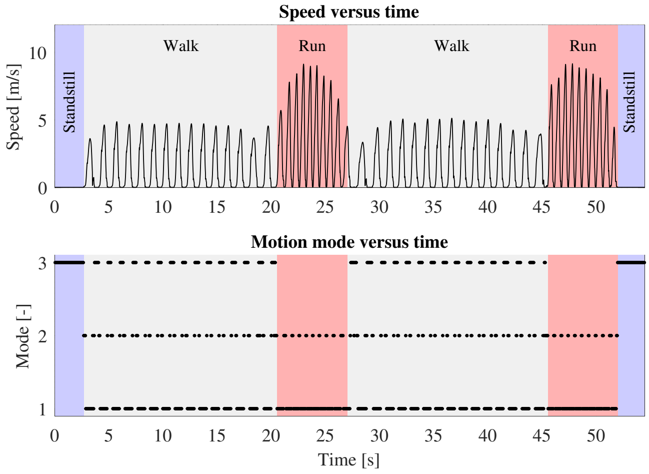
respectively, and the optimization converged in less than 10 iterations. The matrix $\widehat{\Pi}^{\text{end}}$ corresponds to a mode transition system with low-pass characteristics. That is, the probability of staying in a mode is much higher than transitioning to another mode. On average, the percentage of time spent in the motion modes one, two, and three is 57%, 5%, and 38%, respectively.

The learned mode transition matrix was then used in the filter bank when processing the measurements in the second data set. The result is shown in Fig. 2. In Fig. 2a the speed and most likely mode sequence $\hat{\delta}_{1:k}$ estimated by the filter bank from one of the measurement sequences are shown. And in Fig. 2b the horizontal plane errors at the end of the trajectory are shown.

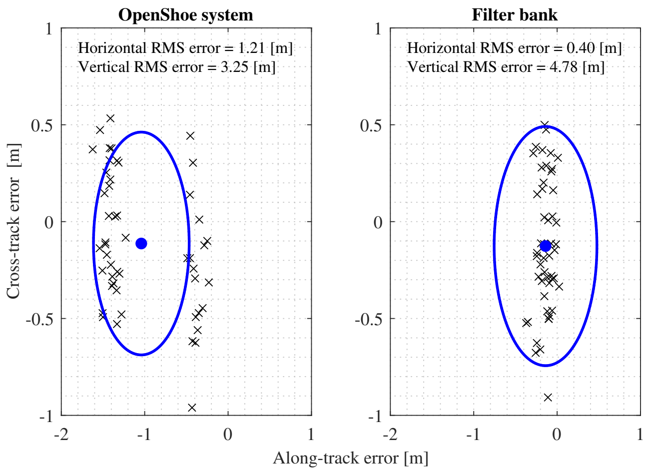
From the figures, the following things can be observed. The filter bank adaptively selects different motion modes depending on the gait speed. That is, when walking, the filter bank frequently selects the stationary motion mode as the most likely mode, whereas when running, the filter bank never selects the stationary mode. Comparing the horizontal positioning error of the OpenShoe system and the filter bank the systems have about the same cross-track error, whereas the filter bank has a significantly smaller along-track error. The OpenShoe system has an along-track bias error of about minus one meter. This is likely due to the high detection threshold used in the zero-velocity detector, causing zero-velocity updates to be applied even when the system is moving. This, in turn, causes part of each step to be cut away, especially when the user walks and the foot's transition from stationary to moving is less distinct. Thanks to the ability of the filter bank to adaptively select the detection threshold and the covariance used in the zero-velocity update, this effect is reduced. However, looking at the vertical root mean square (RMS) error, both systems have an error of several meters. How to reduce this error will be illustrated next.

B. Example: Return to same height

The vertical position error often grows faster than the horizontal position error in foot-mounted zero-velocity aided inertial navigation systems. This is because the beginning and end of each step, when the foot is mainly moving in the vertical direction, are cut away by the zero-velocity updates. However, the vertical error growth can be significantly reduced by noting that most of the time a person is walking on flat ground and the foot should return to the same height [3]. To design a system using the proposed framework that incorporates this information, we introduce the following three motion modes: the unconstrained motion mode ($\delta_k = 1$), the stationary at new height motion mode ($\delta_k = 2$), and the stationary at the same height motion mode ($\delta_k = 3$). A motion mode transition

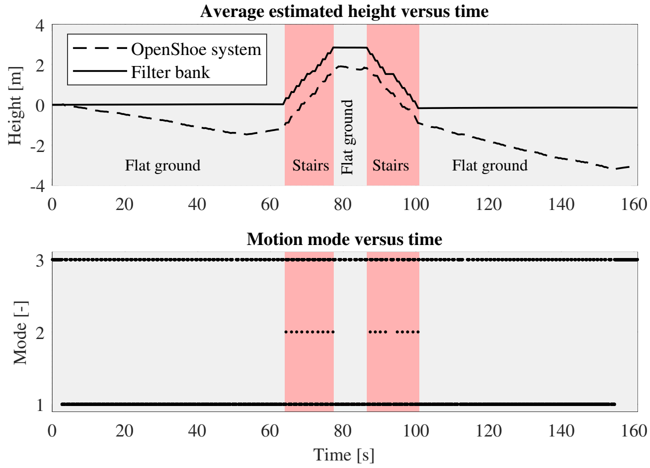


(a) Estimated speed and most likely motion mode versus time.

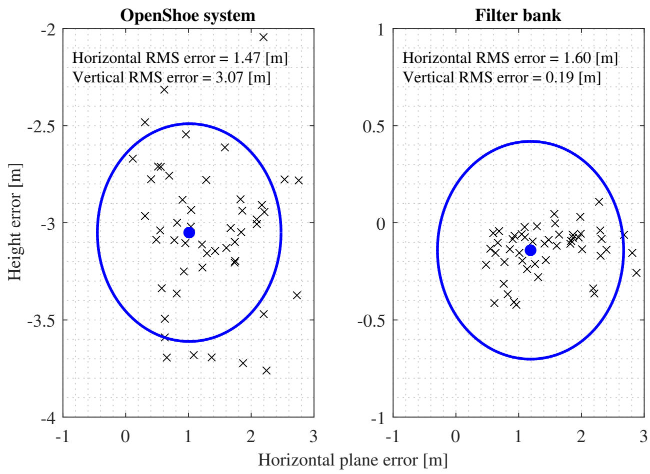


(b) Horizontal position error at end of the trajectory. Also shown are the mean error (blue dot) and 95% confidence interval (blue circle) calculated from the filter covariance, assuming the error to be normally distributed.

Fig. 2: Results from varying gait speed application example.



(a) Estimated height and most likely motion mode versus time.



(b) Height versus horizontal position error at end of the trajectory. Also shown are the mean error (blue dot) and 95% confidence interval (blue circle) calculated from the filter covariance, assuming the error to be normally distributed.

Fig. 3: Results from return to same height application example.

diagram illustrating how the system is assumed to transition between these motion modes is shown in Fig. 1b. Here it has been assumed that if the system becomes stationary at a new height, the system will be stationary for at least two samples. Hence the probability of transitioning from mode $\delta_k = 2$ to mode $\delta_k = 3$ is one.

Next, we introduce the following extended system dynamics

$$f_{\text{ext}}(x_k, \delta_k, u_k, \eta_k) = \begin{bmatrix} f(x_k, \delta_k, u_k, \eta_k) \\ \mathbb{1}(\delta_k = 1)\xi_k + \mathbb{1}(\delta_k \neq 1)[p_k]_3 \end{bmatrix}, \tag{13a}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function and $[a]_j$ denotes j:th element of the vector a. Furthermore, the auxiliary state ξ_k stores the height from the time when the system was last stationary.

The stochastic motion constraints are as follows

$$h_{\text{ext}}(x_k, u_k, \delta_k) = \begin{cases} 0 \quad , & \delta_k = 1 \\ h_0(x_k, u_k), & \delta_k = 2 \\ \begin{bmatrix} h_0(x_k, u_k) \\ [p_k]_3 - \xi_k \end{bmatrix}, & \delta_k = 3 \end{cases}. \tag{13b}$$

The "hardness" of these constraints is controlled by the covariance matrices

$$R_{\text{ext}}(\delta_k) = \begin{cases} \sigma_{nc}^2 I, & \delta_k = 1 \\ R_0(\delta_k), & \delta_k = 2 \\ R_0(\delta_k) \oplus \sigma_h^2(\delta_k) I, & \delta_k = 3 \end{cases}, \tag{13c}$$

where $\sigma_h^2(\delta_k)$ denotes the variance associated with the stochastic constraints on height.

The jump Markov model defined by (13a), (13b), and (13c) was used to design a filter bank, and two data recordings were collected while a person walked on flat ground and then climbed and descended a stair. Following the same procedure as in the previous example, the data recordings were used to create two data sets with 50 measurement sequences each. The measurement sequences in the first data set were used to learn the mode transition matrix Π. The initial value and end result of the learning process were

$$\widehat{\Pi}^{\text{init}} = \begin{bmatrix} 1/3 & 0 & 1/2 \\ 1/3 & 0 & 0 \\ 1/3 & 1 & 1/2 \end{bmatrix} \text{ and } \widehat{\Pi}^{\text{end}} = \begin{bmatrix} 0.976 & 0 & 0.031 \\ 0.003 & 0 & 0 \\ 0.021 & 1 & 0.969 \end{bmatrix},$$

respectively, and the optimization converged in less than 10 iterations. Once again the matrix corresponds to a mode transition system with low-pass characteristics, where the average percentages of time spent in motion modes one, two, and three is 56.2%, 0.2%, and 43.6%, respectively.

The learned mode transition matrix was then used in the filter bank when processing the measurements in the second data set. Once again, pruning was used to keep the tree size to a maximum of nine leaves and a pruning strategy where the most leaves were retained was used. The result is shown in Fig. 3. In Fig. 3a the height and the most likely mode sequence $\hat{\delta}_{1:k}$ estimated by the filter bank is shown. And in Fig. 3b the height versus horizontal plane errors at the end of the trajectory are shown.

From the figures, the following things can be observed. The filter bank can detect whether the user walks on flat ground or not, i.e., whether the foot returns to the same height as in the previous step or not. Therefore it effectively reduces the height error from several meters to a few decimeters. A minor increase in the horizontal position error is observed with the filter bank.

VI. Discussion and Conclusions

A framework for tightly integrated motion classification and state estimation in motion-constrained inertial navigation systems has been presented. The framework provides a structured and theoretically sound way to design motion-constrained inertial navigation systems that can learn different motion patterns. The application of the framework has been illustrated via two examples of foot-mounted zero-velocity aided inertial navigation systems. The examples show that a significant performance gain can be achieved compared to a standard foot-mounted inertial navigation system. However, the price paid is an increased computational complexity (proportional to the number of filters in the filter bank) and an increased number of system parameters to tune. The challenges with tuning the system parameters can to some extent be alleviated via the proposed parameter learning method. Still, tuning the system parameters can be challenging, especially if many motion modes are included in the system model.

VII. Outlook and Future Research

A basic version of the filter bank framework has been presented and tested with models that have a few motion modes. Many possible improvements and open research questions exist. First and foremost, the framework should be compared against other adaptive zero-velocity detector frameworks. More complex models that include a variety of motion modes should also be explored. Related to that, hieratical model structures constructed of several linked small hidden Markov models for the motion states are of special interest to obtain a computationally attractive algorithm. Further, to obtain a smoother transition between motion modes and a more robust algorithm framework, the feasibility of substituting the normal distribution in (6c) with a more heavy-tailed distribution, such as the Student-t distribution, should be explored. Moreover, since the probability of transitioning between motion modes varies both with time and the motion dynamics, the use of a constant mode transition probability matrix Π is clearly suboptimal. Therefore, the feasibility of learning a time-varying and motion-dynamic dependent transition matrix should be explored.

Acknowledgment

This work has been partially funded by the Swedish Research Council project 2020-04253 *Tensor-field based localization* and the Dutch Research Council (NWO) research program Veni project 18213 *Sensor Fusion For Indoor Localisation Using The Magnetic Field*.

References

[1] J. Wahlström and I. Skog, "Fifteen years of progress at zero velocity: A review," *IEEE Sensors J.*, vol. 21, no. 2, pp. 1139–1151, Jan. 2021.
[2] W. Zhang, D. Wei *et al.*, "The improved constraint methods for foot-mounted PDR system," *IEEE Access*, vol. 8, pp. 31 764–31 779, Feb. 2020.
[3] K. Abdulrahim, C. Hide *et al.*, "Using constraints for shoe mounted indoor pedestrian navigation," *The Journal of Navigation*, vol. 65, no. 1, p. 15–28, 2012.
[4] N. Kronenwett, S. Qian *et al.*, "Elevator and escalator classification for precise indoor localization," in *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, Nantes, France, Sep. 2018, pp. 1–8.
[5] J. Wahlström, I. Skog *et al.*, "Zero-velocity detection – A Bayesian approach to adaptive thresholding," *IEEE Sensors L.*, vol. 3, no. 6, pp. 1–4, Jun. 2019.
[6] C.-S. Jao and A. M. Shkel, "ZUPT-aided INS bypassing stance phase detection by using foot-instability-based adaptive covariance," *IEEE Sensors J.*, vol. 21, no. 21, pp. 24 338–24 348, Nov. 2021.
[7] E. Mazor, A. Averbuch *et al.*, "Interacting multiple model methods in target tracking: a survey," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 1, pp. 103–123, Jan. 1998.
[8] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.
[9] X. Rong Li and V. Jilkov, "Survey of maneuvering target tracking. part v. multiple-model methods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1255–1321, Oct. 2005.
[10] X.-R. Li and Y. Bar-Shalom, "Multiple-model estimation with variable structure," *IEEE Transactions on Automatic Control*, vol. 41, no. 4, pp. 478–493, Apr. 1996.
[11] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.
[12] M. Kok, J. D. Hol *et al.*, "Using inertial sensors for position and orientation estimation," *Foundations and Trends in Signal Processing*, vol. 11, no. 1, 2017.
[13] M. Moakher, "Means and averaging in the group of rotations," *SIAM J. Matrix Anal. Appl.*, vol. 24, no. 1, p. 1–16, Jan. 2002.
[14] V. Jilkov and X. Li, "Online Bayesian estimation of transition probabilities for Markovian jump systems," *IEEE Trans. Signal Process.*, vol. 52, no. 6, pp. 1620–1630, Jun. 2004.
[15] E. Özkan, F. Lindsten *et al.*, "Recursive maximum likelihood identification of jump Markov nonlinear systems," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 754–765, Feb. 2015.
[16] J.-O. Nilsson, A. K. Gupta *et al.*, "Foot-mounted inertial navigation made easy," in *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, Busan, South Korea, Oct. 2014, pp. 24–29.
[17] J. A. Farrell and M. Barth, *The Global Positioning System and Inertial Navigation*. McGraw-Hill, 1998.