Integrated navigation

Master's Degree in Geomatics Engineering and Geoinformation

Academic Year 2017-2018

E.T.S.I.Geodesica, Cartografíca y Topográfica

Integrated navigation

- ✓Introduction
- ✓Advanced Kalman filtering
- ✓Principles of sensor fusion
- ✓Typical multisensor systems
- ✓ Based on Hofmann-Wellenhof *et al.*

Introducción

Integrated navigation → when a system uses more than one sensor simultaneously, and the determination of the state vector (position, velocity, and attitude) has to follow principles of sensor fusion

Appropriate methods of signal processing are required (methods of computing and updating the actual state vector) → extended Kalman filtering

Aside from obvious advantages, optimal sensor fusion depends on both the redundancy of the information and the filter design → principles of sensor fusion

The integration of different navigation sensor requires careful handling of the information provided by the individual systems (data synchronization or location of physical origins) → typical multisensor systems: GNSS/INS

Extended Kalman filtering

✓ Motivation

Kalman filtering (as seen in Unit 3) is a linear estimation technique

In real applications, however, the functional dependencies describing the observation equations and the system dynamics are nonlinear.

This problem may be circumvented by considering only small changes of the state vector parameters, which may lead to an almost linear behavior of the functional models.

Consequently, linear approximation may be applied to the effects of small deviations of the state vector from given values.

These values may be either derived from a predetermined trajectory (considerable computations prior to the mission) or from a prediction of the actual trajectory.

The linearization of the system with respect to a prediction of the actual trajectory leads to extended Kalman filtering (EKF). This is the more general approach since there is no need for a previous knowledge of the trajectory.

Extended Kalman filtering

✓ Observation equations

Linear equations for Kalman filtering (as seen in Unit 3)

$$\mathbf{z} = \mathbf{H}\,\mathbf{x} + \mathbf{v} \qquad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$$

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\,\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \qquad \mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$$

Nonlinear equations for EFK

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \qquad \mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$$ observation noise

$$\mathbf{x}_{k+1} = \varphi_k(\mathbf{x}_k) + \mathbf{w}_k \qquad \mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$$ system noise

$h_{k\,(i)}(\mathbf{x}_k)$, for $i = 1, ..., l_k$, vector of "observation functions"

$\varphi_{k\,(j)}(\mathbf{x}_k)$, for $j = 1, ..., n$ vector of "dynamic functions"

Extended Kalman filtering

In some cases, the dynamic functions are not directly accessible

Instead of that, continuous differential equations modeling the change of the state vector with time may be given. These differential equations are known as system equations. The general expression is given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$$

The dynamic functions may be obtained by integration of this equation

It is worth noting that $\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$ defines "a kinematic modeling of motion describing the trajectory of a moving object by a time-variant state vector derived from measurements", whereas $\mathbf{x}_{k+1} = \varphi_k(\mathbf{x}_k) + \mathbf{w}_k$ is subject to a dynamic model of motion: "dynamic modeling explains motion by specifying forces which result in time-variant state vectors" (Schwarz 1998)

Extended Kalman filtering

Linearizing the observation functions with respect to the predicted trajectory represented by $\tilde{\mathbf{x}}_k$ yields the elements of the $l_k \times n$ design matrix $\mathbf{H}_k$ for epoch t_k:

$$H_{k\,(i,p)} = \left.\frac{\partial h_{k\,(i)}(\mathbf{x})}{\partial x_p}\right|_{\mathbf{x}=\tilde{\mathbf{x}}_k} \quad \text{where} \quad \begin{cases} i = 1, \ldots, l_k\,, \\ p = 1, \ldots, n\,. \end{cases} \tag{13.4}$$

Linearizing the dynamic functions with respect to the corrected (i.e., updated by current measurements) trajectory represented by $\hat{\mathbf{x}}_k$ yields the elements of the $n \times n$ transition matrix $\mathbf{\Phi}_k$:

$$\Phi_{k\,(j,q)} = \left.\frac{\partial \varphi_{k\,(j)}(\mathbf{x})}{\partial x_q}\right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \quad \text{where} \quad \begin{cases} j = 1, \ldots, n\,, \\ q = 1, \ldots, n\,. \end{cases} \tag{13.5}$$

Extended Kalman filtering

Step 1: Gain computation (Kalman weight):

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}_k^{\mathrm{T}} \left(\mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right)^{-1} . \tag{13.6}$$

Step 2: Measurement update (correction):

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{h}_k(\tilde{\mathbf{x}}_k)\right) \tag{13.7}$$

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \tilde{\mathbf{P}}_k . \tag{13.8}$$

$$\boxed{\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k \left(\mathbf{z}_k - \mathbf{H}_k \tilde{\mathbf{x}}_k\right)}$$

Step 3: Time update (prediction):

$$\tilde{\mathbf{x}}_{k+1} = \boldsymbol{\varphi}_k(\hat{\mathbf{x}}_k) \tag{13.9}$$

$$\tilde{\mathbf{P}}_{k+1} = \boldsymbol{\Phi}_k \mathbf{P}_k \boldsymbol{\Phi}_k^{\mathrm{T}} + \mathbf{Q}_k . \tag{13.10}$$

$$\boxed{\tilde{\mathbf{x}}_{k+1} = \boldsymbol{\Phi}_k \hat{\mathbf{x}}_k}$$

Extended Kalman filtering

- ✓ Uncorrelated group of measurements

In general, the sets of measurements of different epochs are considered to be uncorrelated. In addition, the observations within a single epoch t_k are fully or at least partly (in groups) uncorrelated for most of the applications. This means that the covariance matrix $\mathbf{R}_k$ of the measurements has a diagonal or at least block-diagonal structure. For such applications, it is preferable to consider the $m_k \leq l_k$ independent parts of $\mathbf{z}_k$ as independent subvectors $\mathbf{z}_k^{(j)}(j = 1, ..., m_k)$ or even scalars rather than to use a full vector representation.

Extended Kalman filtering

- ✓ Uncorrelated group of measurements

For example, two independent groups of measurements $\mathbf{z}^{(j)}$, for $j = 1, 2$, are considered at a given epoch t_k. For the sake of simplicity, linear observation equations are assumed again. Then, the set of observation equations yield

$$\begin{bmatrix} \mathbf{z}^{(1)} \\ \mathbf{z}^{(2)} \end{bmatrix}_k = \begin{bmatrix} \mathbf{H}^{(1)} \\ \mathbf{H}^{(2)} \end{bmatrix}_k \mathbf{x}_k + \begin{bmatrix} \mathbf{v}^{(1)} \\ \mathbf{v}^{(2)} \end{bmatrix}_k , \tag{13.11}$$

and the corresponding covariance matrix is given by

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{R}^{(1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{(2)} \end{bmatrix}_k . \tag{13.12}$$

Extended Kalman filtering

- ✓ Uncorrelated group of measurements

Exploiting now the (partly) uncorrelated character of the measurements, the measurement update (including the gain computation) of Kalman filtering becomes an iterative procedure with m_k iterations per measurement update at epoch t_k. The formalism of recursive least squares (Sect. 3.6.2) given by Eqs. (3.88) through (3.90) is directly applied to the gain and correction computations according to Eqs. (3.92) through (3.94). With the exception of the first step, i.e., for observation groups $j = 2, ..., m_k$, each iteration yields

$$\mathbf{K}_k^{(j)} = \mathbf{P}_k^{(j-1)} \mathbf{H}_k^{(j)\mathrm{T}} \left(\mathbf{H}_k^{(j)} \mathbf{P}_k^{(j-1)} \mathbf{H}_k^{(j)\mathrm{T}} + \mathbf{R}_k^{(j)} \right)^{-1} \tag{13.13}$$

$$\hat{\mathbf{x}}_k^{(j)} = \hat{\mathbf{x}}_k^{(j-1)} + \mathbf{K}_k^{(j)} \left(\mathbf{z}_k^{(j)} - \mathbf{H}_k^{(j)} \hat{\mathbf{x}}_k^{(j-1)} \right) \tag{13.14}$$

$$\mathbf{P}_k^{(j)} = \left(\mathbf{I} - \mathbf{K}_k^{(j)} \mathbf{H}_k^{(j)} \right) \mathbf{P}_k^{(j-1)} . \tag{13.15}$$

Extended Kalman filtering

- ✓ Uncorrelated group of measurements

The main advantages of processing uncorrelated sets of measurements iteratively during the Kalman filter correction step are a reduced computation time and a better numerical stability since the inversion of an $l_k \times l_k$ matrix as required for the gain computation is avoided (cf. Eq. (13.6)). Besides, the iteration procedure at epoch t_k allows the filter to omit certain groups of measurements if they are not available at this epoch. In other terms, the filter may operate on a data-driven basis. This means that as soon as new measurements are available, the state is predicted to the new epoch and the new measurements are used for the corresponding correction step.

Principles of sensor fusion

✓ Motivation

The requirements regarding quality parameters (accuracy, availability, reliability, integrity, etc.) of today's navigation systems are increasing

To meet all the demands, some of the appropriate innovations are:

- the improvement of existing sensors and services, e.g., providing receiver autonomous integrity monitoring (RAIM) techniques to GPS receivers, densification of an existing Loran-C network by new transmitter stations, etc.
- the development of new sensors and services, e.g., the development of indoor GPS receivers, design of new satellite-based navigation systems like Galileo, etc.
- the realization of sensor fusion, e.g., the support of GPS by inertial navigation, the integration of satellite-based and terrestrial radio navigation, etc.

Sensor fusion seems to be a rather cost-effective way

Moreover, the availability of a multisensor navigation system often is a prerequisite for an operational approval of a navigation service

Principles of sensor fusion

- ✓ Types of redundancy

An essential feature of sensor fusion is the presence of redundant information, i.e., more information than required to solve a defined task is available about a given process. Four types of redundancy are distinguished:

Parallel redundancy arises by using several identical sensors or devices. Voting systems directly compare the signals of the sensors to get a unique solution.

Complementary redundancy arises if two or more sensors with different physical operation principles and varying characteristics are used. The sensors complement each other in the way that the advantage of the one could be the disadvantage of the other and vice versa. As an example, the combination of inertial navigation and GNSS may be considered.

Dissimilar redundancy occurs in case of two or more non identical sensors which do not fully complement each other. A typical example is the integration of GNSS and Loran-C: both systems provide position fixes based on RF techniques but differ in terms of system architecture, signal structure, etc.

Analytical redundancy is based on a predefined knowledge of the system models. This knowledge may refer to kinematic modeling with respect to the measurement environment, e.g., in case of a line-based trajectory on a given network; map aiding belongs to this category. As far as the dynamic model is concerned, pre-knowledge of velocity and acceleration limitations may be given.

Principles of sensor fusion

✓ Updating process

The multisensor technique requires appropriate methods of updating the navigation solution by redundant information. Several methods may solve this task:

Signal blending (averaging) is usually applied in case of parallel redundancy. When using several sensors of different quality, weighted averaging is applied. Signal blending does not take into account a dynamic model.

Filtering tries to achieve a more realistic processing of the signals by involving a dynamic model of the motion. In case of conventional filtering, stationary stochastic covariance models are used for the updating process.

Optimal filtering employs time-variant stochastic covariance models and is achieved by Kalman filtering which is commonly applied for updating the state vector gained by multisensor navigation systems.

Principles of sensor fusion

✓ Filter design

Besides the choice of an appropriate dynamic model, several other aspects of filter architecture strongly influence the behavior of the filter and all the characteristics of its output.

Examples: a one-stage filter has a more extensive workload than the master filter of a comparable two-stage filter; an adaptive filter has more convenient properties with respect to error detection than a nonadaptive filter

One-stage vs. two-stage filters

Question → Need the signals of multiple sensors to be processed in a common filter or are certain preprocessing steps are applicable

Three possible approaches:

- The central filter is a one-stage filter where two or more sensors are integrated on the measurement level, i.e., the measurement data are not preprocessed within the single sensors.
- In case of a two-stage decentralized filter architecture, the measurement data of a given sensor are processed internally before importing them into a common master filter.
- The federated filter is a special case of decentralized filtering and works on an information sharing principle. The dynamic model and its system noise are shared among the individual prefilters.

Principles of sensor fusion

- ✓ Filter design

Adaptive filtering

When implementing a Kalman filter for a given application, the required covariance matrices of the measurement and system noise have to be properly defined to obtain an optimal performance of the filter.

Since these matrices are not exactly known a priori, only model-driven estimates are used.

The aim of adaptive filtering is to improve the noise models during operation due to a nonoptimal behavior of the filter.

The term "adaptive filtering‟ is generally applied to situations where certain filter settings are optimized during operation.

Principles of sensor fusion

- ✓ Filter design

Failure detection and isolation

The main task of failure detection and isolation (FDI) in a redundant sensor system is a two-step procedure in which the (final) isolation of an error requires significantly more computational effort and skill than the (initial) detection of the error itself.

Usually, FDI is realized within an adaptive Kalman filter denoted as monitoring system.

The principal difficulty is that an FDI algorithm should detect errors as fast as possible.

Thus, it must be sensitive with respect to highly frequent effects.

Since noise is also a high-frequency effect, the system becomes more sensitive to noise.

Consequently, the filter performance under normal conditions is deteriorated. In addition, raising the sensitivity of the system also increases the probability of wrong alarms.

Typical multisensor systems

- ✓ GNSS/INS

The integration of satellite-based and inertial navigation (which is actually a DR technique) represents another complementary approach of sensor fusion.

Typically, strapdown inertial navigation systems (INS) are used together with different types of GNSS receivers. The main limitation to the wide-spread use of GNSS/INS solutions are the still high costs of INS. This situation may change considerably by the further development of low-cost INS sensors.

The main benefits of the GNSSjINS integration are that GNSS tends to bound the long-term drift effects experienced by an INS, and inertial navigation allows bridging GNSS outages due to signal blockage, jamming, or spoofing.

A general problem of complementary sensor fusion, however, is that unusual failures affecting one of the subsystems are difficult to detect and identify. An INS, e.g., can scarcely reveal possible long-term GNSS positionbiases.

Typical multisensor systems

- ✓ GNSS/INS

Comparison of GNSS and INS

Characteristics	GNSS	INS
Operation	Nonautonomous	Autonomous
Information	Absolute	Relative
Output rate	Typically 1 Hz	50 Hz or more
Short-term accuracy	Low	High
Long-term accuracy	High	Low
Availability	Limited	Unlimited
Vulnerability	High	None

Typical multisensor systems

✓ GNSS/INS

Implementation options

Three principal options for implementing an integrated GNSS/INS are distinguished:

- Uncoupled integration
- Loosely coupled integration
- Tightly coupled integration

Uncoupled integration (open loop mechanization)

- GNSS $\rightarrow$ position,velocity, and time
- INS $\rightarrow$ position,velocity, and attitude
- 2D implementation

$$\begin{bmatrix} \varphi \\ \lambda \end{bmatrix}_k = \begin{cases} \begin{bmatrix} \varphi_{\mathrm{GNSS}} \\ \lambda_{\mathrm{GNSS}} \end{bmatrix}_k \\ \begin{bmatrix} \varphi_{\mathrm{INS}} \\ \lambda_{\mathrm{INS}} \end{bmatrix}_k + \begin{bmatrix} \varphi_{\mathrm{GNSS}} - \varphi_{\mathrm{INS}} \\ \lambda_{\mathrm{GNSS}} - \lambda_{\mathrm{INS}} \end{bmatrix}_0 \end{cases}$$

Typical multisensor systems

✓ GNSS/INS

Loosely coupled integration

In the case of loosely coupled integration, the individual sensors still perform a preprocessing:

GNSS $\rightarrow$ position, velocity, and time information
INS $\rightarrow$ position, velocity, and attitude

In contrast to the uncoupled integration, the INS sensor errors (e.g., gyro bias, accelerometer bias, nonorthogonality of the sensors) are calibrated by a feedback link in a federated Kalman filter algorithm.

Another feedback link from the master filter towards the GNSS receiver may be established. This link provides an improved estimate of the current vehicle velocity mainly on the basis of the INS data to the GNSS receiver.

Such type of velocity aiding enables the GNSS receiver to better predict the Doppler shifts of the satellite signals and, thus, to enhance satellite (re-) acquisition and tracking.
The resulting Kalman filter equations are not discussed in all details here.
Only one option for the system equations is presented. The linear system equation for GNSS is given by

Typical multisensor systems

- ✓ GNSS/INS

Loosely coupled integration

The linear system equation for GNSS is given by

$$\begin{bmatrix} \dot{\mathbf{x}}^e \\ \dot{\mathbf{v}}^e \end{bmatrix} = \begin{bmatrix} \mathbf{v}^e \\ \mathbf{a}^e \end{bmatrix}_{\text{GNSS}} \qquad \begin{bmatrix} \dot{\mathbf{x}}^e \\ \dot{\mathbf{v}}^l \\ \dot{\mathbf{R}}_b^l \end{bmatrix} = \begin{bmatrix} \mathbf{R}_l^e \mathbf{v}^l \\ \bar{\mathbf{g}}^l + \mathbf{f}^l - (\mathbf{\Omega}_{il}^l + \mathbf{\Omega}_{ie}^l)\mathbf{v}^l \\ -\mathbf{\Omega}_{il}^l \mathbf{R}_b^l + \mathbf{R}_b^l \mathbf{\Omega}_{ib}^b \end{bmatrix}_{\text{INS}}$$

To achieve consistency between the GNSS and INS parts, the GNSS velocity vector may be transformed to the l-frame by $\mathbf{R}_e^l$

Typical multisensor systems

✓ GNSS/INS

Tightly coupled integration (closed-loop mechanization)

It is based on a centralized processing of the (raw) measurement data of the subsystems.

The GNSS receiver provides code pseudorange and/or carrier phase measurements, the INS provides specific-force and angular-rate measurements.

The INS sensor errors are calibrated within the centralized Kalman filter, and a velocity aiding feedback link is established towards the GNSS receiver.

The main advantage of tight coupling is that limited information from GNSS may also be used for the integration: two or three satellites already provide valuable information to support the navigation solution.

In addition, the tightly coupled approach may facilitate GNSS ambiguity resolution when carrier phase measurements are performed (see Jekeli Sect. 10.3).

Some textbooks dedicated to the integration of satellite-based and inertial navigation are Farrell and Barth (1999) and Grewal et al. (2001).

Typical multisensor systems

✓ Map aiding

The map aiding technique refers to using suitable contents of a digital map as some kind of pseudoobservation to complement the information provided by one or more primary sensors.

Usually based on a vector-type structure, the map serves as an artificial sensor, and its geometric data play the role of an artificial signal.

Map aiding (MA) is only useful together with at least one other sensor and is tailored to sensor fusion.

The combination of a digital map with other sensors is a typical example of analytical redundancy, which, per definition, relies on predefined system models.

In case of MA, the map geometry determines the measurement environment a priori as far as a line-based trajectory is concerned.

Typical multisensor systems

✓ Map aiding vs. map matching

Map matching (MM) is defined as the projection of a vehicle position or trajectory onto a vector-based digital map represented by the node-edge structure of a graph.

MM is - similar to MA - involved in a kind of map-based geo-referencing.

In contrast to MA, there are essential differences:

• MM results in a location situated directly on an edge relative to a node, whereas MA derives a position not necessarily linked to an edge.

• MM cannot directly participate in a filtering process and its dynamic model since exact edge fixing stands in contradiction to a measurement update based on a given measurement noise model.

• MM is always acting as a decentralized procedure, usually after a onestage or two-stage filtering process.