GNSS Technologies

GNSS integration with other positioning methods 2
21.3.2016
Content

- GNSS/INS integration
  - Loose coupling
  - Tight coupling
  - Deep coupling
- EKF for GNSS/INS
- Particle filtering
- Summary

Slides based on:
GNSS Applications and Methods, by S. Gleason and D. Gebre-Egziabher (Eds.), Artech House Inc., 2009
http://www.gnssapplications.org/
GNSS/INS integration architectures nowadays include basically three approaches:

- *loose*
- *tight, and*
- *deep (also called ultratight)*

**GNSS code and carrier range and phase measurements**

**Raw GNSS RF signals**

**Processed GNSS PVT**

**INS acceleration and angular velocity**

**Tightly-coupled**

**Deep/Ultratightly-coupled**

**Loosely-coupled**

**Position**

**Velocity**

**Orientation**

21.3.2016
In the **loose architecture**, the INS and GNSS receivers operate as *independent* navigation systems. The information from them is blended using an estimator to form a third navigation solution. The information fusion occurs at the PVT level: GNSS PVT estimates are blended with those from an INS. The loose integration architecture can be designed as a feed-forward or feedback configuration (open-or closed-loop). In the feedback (closed-loop) configuration, inertial sensor errors estimated by the filter are fed back to compensate for the gyro and accelerometer measurement errors. The sensors are analytically calibrated in real-time.
Loosely-coupled approach

Without the feedback path (closed-loop), the system is a feed-forward (open-loop) configuration.
The feedback path is almost always required when low-quality inertial sensors are used since they have large output errors.

The feed-forward configuration is normally used with high-quality INSs such as those in the navigation grade of sensors (commercial transport jets or fighters).

In the loosely coupled approach in general, faults in GNSS do not affect the INS solution (in the short term) and faults in the INS cannot affect the GNSS solution.

Also, easy to implement because the integration occurs at the PVT level and not the sensor level.

Ensuring that data from both systems is synchronized or time-tagged correctly can be a very challenging task.

When low-cost inertial sensors are used in the loose integration scheme, the INS cannot really be viewed as an independent sensor: without the feedback that comes from GNSS, the INS solution would diverge very quickly.

INS brings the attitude information to the system as well as increased data rate.
A basic schematic of the classic tight GNSS/INS architecture
In a tightly-coupled architecture, the INS and GNSS are reduced to their basic sensor functions: pseudorange, pseudorange rate, accelerations, and gyro measurements are used to generate a single blended navigation solution.

In general, the classical tight architecture provides a more accurate and robust solution than loose integration.

The enhanced accuracy is, in part, due to the fact that the GNSS information used by the blending filter (i.e., pseudoranges and range rates) exhibit less temporal correlation than the processed PVT solution.

It is also robust because it can continue to extract useful GNSS information to aid the INS in situations where less than four satellites are visible (however, limited time only because of GNSS receiver clock drift).
The tight integration is more complex than the loosely coupled and multiple independent navigation solutions do not exist.

In a tightly coupling scheme that has tracking loop aiding, information from the blending filter is fed back to the GNSS receiver to enhance its performance (position and/or velocity):

- Shortens the time required for acquisition or reacquisition of the GNSS signal.
- Velocity (and possible acceleration) information can be used to aid the code and carrier tracking loops in the GNSS receiver.
  - This allows the GNSS receiver to remain in lock (i.e., continue to track the signal) in high dynamic maneuvers, which would be impossible, or at least difficult, without the aiding information.
  - Tracking loop aiding can also be used to narrow the tracking loop bandwidths of the GNSS receiver thereby increasing its robustness in environments where the GNSS signal carrier-to-noise ratio is reduced.
There are two variants of the deep integration architecture—the coherent and noncoherent approaches.

Noncoherent deep integration
The distinction between the methods is that the coherent integration utilizes the measurements directly from the correlator, whereas in non-coherent algorithms the discriminator values are the measurements.

In deep integration, the INS is an integral part of the GNSS receiver: the GNSS receiver can no longer be viewed as a navigator independent of the INS.

The advantage of the deep integration architecture is that it enhances the robustness of GNSS to interference and jamming.

The enhancement afforded by the deep architecture is better than that provided by tight integration.

- Optimal fusion of the information from an INS and a GNSS receiver.
- The major drawback of the deep integration scheme is the complexity involved in integrating INS information with GNSS information deep inside the receiver.
  - much of the necessary modifications would require access to the receiver hardware or to a software receiver.
GNSS/INS integration (10)

Jamming started at 48s
Jamming ended at 68s
Max J/S 25 dB

Static XSENS

In going from the loose to the deep integration architectures, we gain robustness to GNSS outages either due to vehicle dynamics, interference, or jamming.

However, this increased robustness comes at the sacrifice of system simplicity, redundancy and independence of the INS and GNSS navigators.
Estimation algorithms

- INS and GNSS data is fused together by an estimator or a mathematical algorithm providing a systematic framework by which information from multiple sources can be combined.

- Most widely used estimator for GNSS/INS integration is the extended Kalman filter, EKF.
  - Kalman filters generate optimal estimates of the navigation states by minimizing the covariance of the estimation error.
  - Their recursive form makes them well-suited to efficient implementation.

- Also other estimation algorithms are applicable for GNSS/INS integration such as the unscented Kalman filter (UKF) and particle filter (PF).
The EKF is designed to handle the situation where the system model is nonlinear.

From the equations of motion, the INS states are clearly nonlinear, and linearization is required - the INS error equations already provide the linearized model.

The loosely coupled EKF state vector can be written as

\[
\delta x^T = [(\delta p)^T (\delta v^n)^T (\delta \psi_{nb})^T (b_a^b)^T b_g^b]^T
\]

position, velocity, orientation, acc. errors, gyro errors
For an EKF operating in a closed-loop configuration, whenever the state vector is updated, the estimated states (i.e., INS errors) are applied to the corresponding navigation parameters and the state vector is reset to zero. Thus, the state vector itself does not need to be propagated forward in time.

The state covariance can be propagated using:

\[ P_{k+1} = \Phi_k P_k \Phi_k^T + Q_k \]

- **\( P_k \)**: the elements of the state error covariance matrix are the variances and covariances of the individual error states.
- **\( \Phi_k \)**: the state error transition matrix is computed from the INS error equations.
- **\( Q_k \)**: the process noise covariance matrix represents stochastic information about the sensor errors being estimated.
The EKF measurement or observation model is given by
\[ \delta y_{k+1} = H \delta x_{k+1} + \epsilon \]

- For a tight integration, the measurements will be the difference between the pseudorange measured by the GNSS receiver and the pseudorange predicted based on the INS estimated position of the user and knowledge of the satellites’ locations.
- A loose integration architecture requires that the measurements be formed as the difference between the GNSS and INS position and velocity as error estimates.
In Kalman filtering theory, the measurement noise $\epsilon$ is assumed uncorrelated in time.

- It is a well-known fact, however, that GNSS measurement errors are time-correlated.
  - Solution:
    - to augment the state vector with additional states representing the correlated errors
    - to account for the correlation by artificially inflating the measurement noise covariance matrix (suboptimal) - weighting

- When low-cost inertial sensors are integrated with GNSS, the objective is not necessarily to get a position and velocity solution that is of a higher quality than the GNSS solution, but rather to increase bandwidth of the position and velocity solution (update rate) as well as provide attitude.

- The measurement model is used to perform the measurement update of the state vector and its covariance matrix when GNSS measurements are available.
The equations required for the measurement update includes:

- First computing the gain matrix,
- Then updating the state error estimate, and
- Finally updating the state error covariance matrix.

The Kalman gain matrix is denoted as:

\[ K_{k+1} = P_{k+1}^{-} H_{k+1}^{T} \left( H_{k+1} P_{k+1}^{-} H_{k+1}^{T} + R_{k+1} \right)^{-1} \]

The state error estimate after the measurement update is:

\[ x_{k+1}^{+} = K_{k+1} y_{k+1} \]

And the state error covariance matrix after the measurement update is:

\[ P_{k+1}^{+} = (I - K_{k+1} H_{k+1}) P_{k+1}^{-} \]

The role of the Kalman gain matrix is to weigh the relative importance of the time-updated states against the information provided by the GNSS measurements.

If the GNSS measurements are of better quality then the gain matrix will be large and place more weight on the GNSS solution, and, if the time-updated states are considered more precise, then the Kalman gain is smaller, thus giving less importance to the measurements.
The EKF for the GNSS/INS integration generally behaves as

- Initially the navigation parameters are initialized using the best available information and the state error vector is set to zero and the state covariance matrix is initialized to reflect the uncertainty in the initial INS parameter estimates.
- During the time update phase, the mechanization equations are used to propagate the position, velocity, and attitude estimates forward in time.
  - Because of the IMU sensor errors, time propagation introduces uncertainty into the system.
- When information from GNSS becomes available, a measurement update is performed.
  - The GNSS data introduces more information to the system and correspondingly decreases the uncertainty in the system.
The first step of any GNSS/INS system is to determine the initial values of position, velocity, and attitude.
- Estimates of the sensor errors can also be initialized using, for example, results of a lab calibration, or, the sensor error can be calibrated to zero.
- Diagonal entries of the covariance matrix of the state are equal to the variances of the various states.
  - For the position states, the covariance matrix can be obtained from the estimated accuracy of the GNSS position solution.
- Once the system is initialized, the inertial sensors are sampled and the mechanization equations are used to propagate the navigation states forward in time.
- If operating in a closed-loop configuration, the sensor data should be corrected for the estimated errors before the mechanization equations are executed.
- The IMU data processing is repeated (once for every IMU data sample) until a GNSS measurement is received.
- Upon receipt of GNSS measurements (e.g., position and velocity in a loose integration architecture), the errors’ states and covariance matrix are updated as well as the navigation parameters.
This is a 15 state system and the filter state vector is described by Equation (6.26). That is:

\[
\begin{align*}
\text{(1)} \quad & dp = \text{Position Errors (NED Coordinates)} \\
\text{(2)} \quad & dv = \text{Velocity Errors (NED Coordinates)} \\
\text{(3)} \quad & dpsi_nb = \text{Platform Tilt/Attitude Errors} \\
\text{(4)} \quad & db_a = \text{Accelerometer Bias Estimation Errors} \\
\text{(5)} \quad & db_g = \text{Rate Gyro Bias Estimation Errors}
\end{align*}
\]

\[dx_o = [dp \ dv \ dpsi_nb \ 10*db_a \ db_g]'; \]  % Initial navigation error state vector

\[P = 10*\text{diag}(dx_o.^2)'; \]  % Initial state error covariance
% Compute innovations process

\[
\begin{align*}
\text{posInnov} &= \text{pos_ins_ned}(k,:) \cdot - \text{gps_pos_ned}(k,:) \cdot ; \\
\text{velInnov} &= \text{vel_ins}(k,:) \cdot - \text{gps_vel_ned}(k,:) \cdot ; \\
\text{stateInnov} &= [\text{posInnov}; \text{velInnov}];
\end{align*}
\]

% Compute Kalman gain and update state covariance

\[
\begin{align*}
Pz &= H \cdot P \cdot H^\prime + R; \\
L &= (P \cdot H^\prime) / (Pz); \\
P &= (\text{eye}(15) - L \cdot H) \cdot P; \quad \% \text{Equation (6.33). Update covariance matrix} \\
\text{stateError} &= L \cdot \text{stateInnov}; \quad \% \text{Equation (6.32)} \\
\text{posFeedBack}(k,:) &= \text{stateError}(1:3) \cdot ; \quad \Rightarrow \text{same for other states}
\end{align*}
\]

% Update position, velocity, attitude, biases using computed corrections

\[
\begin{align*}
\text{pos_ins}(k,1) &= \text{pos_ins}(k,1) - \text{posFeedBack}(k,1) / (R_N + \text{pos_ins}(k,3));
\end{align*}
\]

Difference between navigation states estimated using GNSS and INS measurements.
Particle filtering

- Positioning computations are usually non-linear and with non-Gaussian noise, Kalman filtering not optimal way to solve => Particle filters provide better solutions
- Particle filter is an implementation of recursive Bayesian filter by Monte Carlo sampling
  - Posterior density is a set of random weighted particles
- High number of particles often needed => computationally expensive
Particle filtering: Pedestrian positioning (1)

- Pedestrian positioning equations utilizing a constant speed model are applied in a horizontal plane for the position estimation.
- State vector

\[ x_k = \begin{bmatrix} X_k & Y_k & \dot{X}_k & \dot{Y}_k \end{bmatrix}^T \]

- Applied state model

\[ x_k = F x_{k-1} + w_k \]

where

\[ F = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

where \( \Delta t \) is the sampling period.
The problem of tracking the pedestrian boils down to inferring the mobile state $x_k$ from the sequential measurements of speed and heading.

Since the measurement equations for the speed and the heading are nonlinear, a particle filter is used for the state estimation.

A particle filter recursively represents the required posterior density function of a state by a set of random samples with associated weights.
Particle filtering: Pedestrian positioning (3)

- The particles in the position estimation are generated according to the prior distribution of \( x_k \)
  \[ x_{k}^{(j)} = F x_{k-1}^{(j)} + w_{k}^{(j)} \]
  where \( j \) refers to the particle number.

- The corresponding weight \( w_k^{(j)} \) of each particle is updated as
  \[ w_k^{(j)} = l_{y_k,Source_1}^{(j)} * l_{y_k,Source_2}^{(j)} * l_{S_k}^{(j)} * l_{\phi_k}^{(j)} \]

where

\[ l_{y_k,Source_n}^{(j)} = N(y_k ; H x_{k}^{(j)} , R_{Source_n}) \]

\[ l_{S_k}^{(j)} = N(S_k ; \sqrt{(\dot{X}_k^{(j)})^2 + (\dot{Y}_k^{(j)})^2} , R_S) \]

\[ l_{\phi_k}^{(j)} = N(\phi_k ; \arctan(\frac{Y_k}{X_k}) , R_{\phi}) \]

\( R_* \) are the measurement variances.

\( N(\cdot) \) is a normal probability density function.
Particle filtering: Pedestrian positioning (4)

- Based on the weights, deterministic resampling is used to eliminate particles with small weights and replicate ones with large weights.
- At the end of every time step $k$ the state estimation is the weighted mean of the particles

$$\hat{x}_k = \sum_{j=1}^{N} l_k^{(j)} x_k^{(j)}$$
Particle filtering example

Indoor positioning using Particle filtering
Red: estimated position
Blue: particles
GPS/INS field tests at the Finnish Geodetic Institute

Reference GPS/INS:

- **GPS antenna**
- **IMU**
- **GPS receiver**

![Diagram showing GPS/INS field tests setup](image)
Integration examples (1) – RF positioning

- Indoor test route – individual results

![Indoor position result graph]

Mean horizontal error [m]:
- Bluetooth-only: 6.3
- WiFi-only: 8.9
- GPS-only: 13.1
Integration examples (2) – Hybrid

- Hybrid indoor test at FGI

- Standalone high-sensitivity GPS, WLAN-only and multi-sensor multi-network (MSMN) solution with respect to a reference

- The hybrid MSMN solution (GPS, WLAN fingerprinting + dead-reckoning sensors) gives better performance in accuracy and availability (1 Hz)
  - hybrid solution based on EKF
Hybrid indoor test at FGI:

2D Error comparison of GPS, WLAN and hybrid solution

Hybrid (MSMN): GPS, WLAN, accelerometer, digital compass
Integration examples (4) – Hybrid

- Vision-aided multi-sensor multi-network navigation
- Setup
  - Fastrax IT500 GPS receiver
  - Nokia 6710 mobile phone for WiFi
  - Multi-sensor positioning device with 3-axis accelerometer (VTI) and 2-axis digital compass (Honeywell)
  - Nokia N8 camera for visual-aiding
- Measurements fused using a Kalman filter
- Visual information: heading and translation

Mean error
6.7 m

Mean error
5.3 m
Integration examples (5) – Deeply-coupled

- Deeply-coupled GNSS / INS / visual integration for jamming mitigation
- GPS signals were collected using NSL Stereo frontend
- A cheap Covert GPS L1 jammer was used, power restricted to -30 dBm (nominal 13 dBm)
- Special permission from the Finnish Communications Regulatory Authority
Integration examples (6) – Deeply-coupled

Carrier-to-Noise density ratio (C/N₀)

Jamming started

GNSS only => no pvt solution
Integration examples (7) – Deeply-coupled

Carrier-to-Noise density ratio ($C/N_0$)

Jamming started

Jamming ended

GNSS+INS

GNSS+INS+visual

Time [s]

GNSS Technologies 21.3.2016
Integration examples (8) – Deeply-coupled GNSS + INS

Position

Max error 300 m

Velocity (E-frame)

Max error 7.5 m/s

Attitude (Euler angles)

Max error 5 degrees

Easting [m]  Northing [m]  Velocity [m/s]  Time [ms]

-GNSS Technologies  21.3.2016
Integration examples (9) – Deeply-coupled GNSS + INS + visual sensor

Position

Velocity (E-frame)

Attitude (Euler angles)

Max error 80 m

Max error 2 m/s

Max error 3 degrees
Integration examples (10) – Particle filtering

- Infrastructure-free tactical situational awareness (INTACT)
  - a project funded by the Ministry of defence in Finland 1.1.2015-

- Project goals
  - Accurate indoor positioning
  - A map of the unknown environment => Simultaneous Localization and Mapping (SLAM)
  - Motion and context information (run, crawl, stationary)

- Using only equipment attached to the user
  - 3 IMUs, in foot, body and helmet
  - A camera
  - Sonar for ranging
  - Barometer for vertical position
Integration examples (11) – Particle filtering

- Measurements from a foot-mounted IMU, visual-aiding, barometer and sonar integrated using a Particle filtering algorithms with 2000 samples
- Short test, ~3 minutes, 180m long route
- Challenges ahead when the positioning will be prolonged and environments more challenging

Positioning accuracy: mean error 1.8m, std 3.1m
Other methods for improving positioning

- Map matching, constraining the probable position into a map defined
  - Area: Vehicles stay on the road
  - Motion: e.g. in indoor navigation, walls can’t be penetrated

- Context recognition
  - Different motion patterns are classified and recognized using machine learning algorithms
  - Prevailing motion may be used for filtering => e.g. when the user is static, the position doesn’t change
Summary

- Selection of the integration method is always balancing between robustness requirements and processing capabilities.
- Kalman filtering is a sophisticated method for integration, but assumes linear state models and Gaussian errors.
- Extended Kalman filtering and particle filtering provide means for tackling the linearity and Gaussianity requirements.
Last lecture and course work assignment & report

- On Monday 4.4 the summary lecture 12.15-13.45 and the course work assignment discussion 14.00-15.30

- Course work report deadline Wed 19.5

- Exam: Thu 19.5