Loosely coupled INS/GNSS algorithm description (InsGnssFilterLoose)

Open Aided Navigation Project

Revision number 0.2

Dr. Fedor Baklanov

2020-12-20

Contents

1 Introduction

This document contains a description of the loosely coupled INS/GNSS sensor fusion algorithm provided in the open source project Open Aided Navigation.

Nowadays there are many books on navigation system theory. Most of them a very thick: number of pages exceeds 500! The author's experience shows that beginners find it extremely tough to navigate through such a big amount of information and that they often feel unsure of how to come from theory to implementation. This motivated the author to provide detailed algorithm description together with the code published in the Open Aided Navigation repository.

This document attempts to explain the most essential concepts of aided inertial navigation. Reading this document, you will find brief derivations of the core mathematical formulas that were implemented in the software. Mapping the formulas to code should be very straightforward.

To understand well the ideas explained in this document, a reader needs to be familiar with the following subjects:

- Linear algebra
- Calculus
- Ordinary differential equation
- Linear control and estimation
- Probability theory and stochastic processes
- Classical mechanics

Most of the information, presented in this document, can be found in textbooks, dissertations, and journal articles. The author's most favorite reference is the book (Farrell, 2008).

Enjoy reading!

2 Mathematical foundations of aided inertial navigation

2.1 Notation and symbols

Small regular letters x denote scalar values. Small boldface letters $\mathbf{x}$ denote vectors. Capital boldface letters $\mathbf{X}$ are used to denote matrices.

$\mathbf{z}$	True value of the vector quantity $\mathbf{z}$
$\tilde{\mathbf{z}}$	Erroneous or estimated value of $\mathbf{z}$
$\delta\mathbf{z} = \mathbf{z} - \tilde{\mathbf{z}}$	Error vector
$\hat{\mathbf{z}}$	Estimate of $\mathbf{z}$
$\mathbf{I}$	Identity matrix
$\mathbf{x}_a$	A column of coordinates of the vector $\vec{\mathbf{x}}$ in coordinate frame a
$[\mathbf{a}\times]$	Skew-symmetric matrix operator, i. e. given vectors $\mathbf{a}$ and $\mathbf{b}$, $[\mathbf{a}\times]\cdot\mathbf{b} = \mathbf{a}\times\mathbf{b}$
$\mathbf{C}_{ab}$	A matrix that defines rotation from coordinate frame b to coordinate frame a
$\breve{\mathbf{q}}_{ab}$	A quaternion defining rotation from coordinate frame b to coordinate frame a
$\breve{\mathbf{b}}$	A quaternion, whose scalar part is zero, and vector part is equal to the vector $\mathbf{b}$
$\mathbf{f}_x$	Vector of projections of specific force onto x-frame
$\boldsymbol{\omega}_{xy}$	Angular rate of y-frame relative to x-frame in projection onto y-frame
$*$	Quaternion product

2.2 Principles of aided inertial navigation

Accelerometers and gyroscopes are usually called *inertial sensors*. An accelerometer is a sensor that measures one or more projections of a specific force acting on a body. A gyroscope is a device for measuring projections of angular rate of the body relative to some inertial coordinate system.

Nowadays the abovementioned sensors are usually rigidly attached to a moving object, which is why the measured values present projections onto a vehicle-fixed coordinate frame.
The core idea of inertial navigation consists in determining position and orientation of a moving object by means of integrating numerically differential equations of motion of a point mass in the neighborhood of Earth. *Angular rate* and *specific force* are inputs in these equations and need to be measured by onboard inertial sensors. Additionally, initial position, velocity, and orientation are required to be known.
Accuracy of the navigation parameters determined by an inertial navigation system degrades continuously due to IMU's instrumental errors, initial state errors, and errors of numerical integration. That is why in parallel to "navigating purely inertially" it is necessary to estimate errors of position, velocity, and orientation using information from external measurement systems, such as a GNSS receiver, for example. The obtained error estimates can then be used to correct position, velocity, and orientation information. In general, external measurements are not required to be available continuously, because INS is able to do dead reckoning using IMU information only. The rate of accuracy degradation during dead reckoning strongly depends on IMU accuracy class.
Figure 1 illustrates the described principles of an aided inertial navigation system.

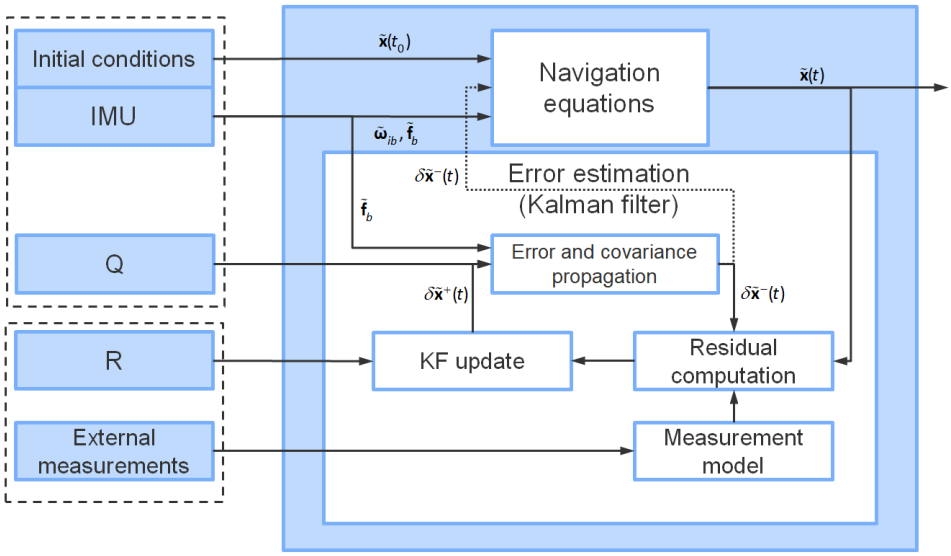

Figure 1

2.3 Coordinate frames

2.3.1 Quasi-inertial coordinate frame (i-frame)

The origin P_i of the quasi-inertial coordinate frame is located in the Earth's center of mass. The first axis $\mathbf{i}_1$ points towards the point of intersection at time t_0 of the zero meridian and equatorial plane. The third axis $\mathbf{i}_3$ points from the origin towards the IERS reference pole. The second axis $\mathbf{i}_2$ is an orthogonal completion of the $\mathbf{i}_1$ and $\mathbf{i}_3$ to a right-handed axes system.

2.3.2 Earth-centered Earth-fixed coordinate frame (e-frame)

ECEF frame (e-frame) is a coordinate system with the origin P_e in the Earth's center of mass. The third axis $\mathbf{e}_3$ points from the origin towards the IERS reference pole. The first axis $\mathbf{e}_1$ points from the origin towards the intersection of the IERS reference meridian and the plane passing though the origin perpendicular to $\mathbf{e}_3$. The second axis $\mathbf{e}_2$ completes the right-handed system.

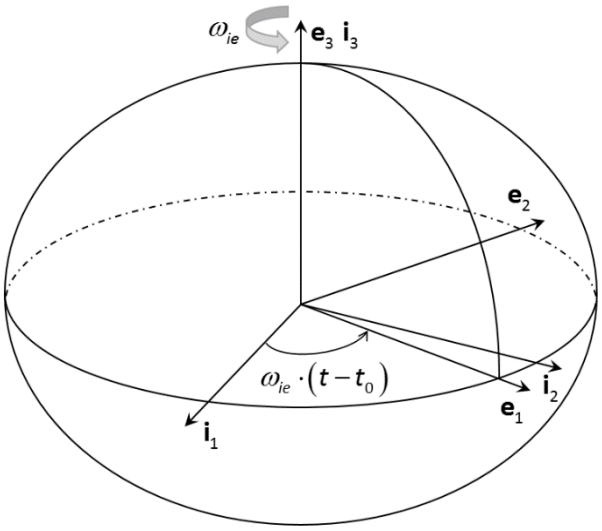

Figure 2: i-frame and e-frame

2.3.3 Local-level North-indicating coordinate frame (n-frame)

The third axis $\mathbf{n}_3$ is perpendicular to the local tangent plane to WGS84 Ellipsoid. The first axis $\mathbf{n}_1$ is parallel to the North direction in the local tangent plane to WGS84 Ellipsoid. The second axis $\mathbf{n}_2$ completes the right-handed system. Origin of the n-frame can be placed to the Earth's center of mass.

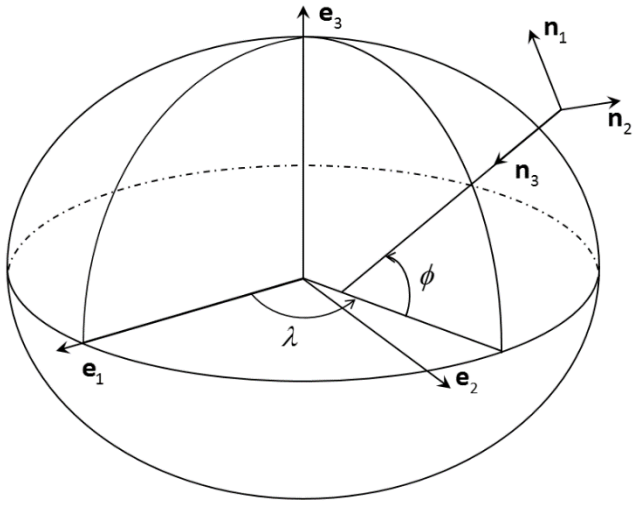

Figure 3: North-East-Down coordinate frame (n-frame)

2.3.4 Sensor coordinate frame (s-frame)

This is an orthogonal coordinate frame whose axes are parallel to the sensitivity axes of an IMU.

2.3.5 Car-fixed coordinate frame (c-frame)

The second axis $\mathbf{c}_2$ points right and is parallel to the rear axle of a car. The third axis $\mathbf{c}_3$ is parallel to the plane that is perpendicular to $\mathbf{c}_1$ and points towards the place of contact of the wheel and road. The first axis $\mathbf{c}_1$ completes the right-handed system (points forward).

2.4 Navigation equations

Motion of a point mass in the neighborhood of Earth can be modeled by the ordinary differential equation given below. This equation and its derivation can be found in the book (Farrell, 2008).

$$\begin{aligned}\dot{\mathbf{x}}_e &= \mathbf{v}_e \\ \dot{\mathbf{v}}_e &= -2[\boldsymbol{\omega}_{ie}\times]\mathbf{v}_e + \mathbf{g}_e(\mathbf{x}_e) + \mathbf{C}_{es}(\breve{\mathbf{q}}_{es})\mathbf{f}_s \\ \dot{\breve{\mathbf{q}}}_{es} &= \frac{1}{2}(\breve{\mathbf{q}}_{es} * \breve{\boldsymbol{\omega}}_{is} - \breve{\boldsymbol{\omega}}_{ie} * \breve{\mathbf{q}}_{es})\end{aligned} \tag{2.1}$$

Herein

- $\mathbf{x}_e$ is a 3×1 vector of coordinates of a vehicle in the ECEF coordinate system
- $\mathbf{v}_e$ is a 3×1 column whose elements are projections of the velocity relative to ECEF frame onto basis vectors of the ECEF frame
- $\breve{\mathbf{q}}_{es}$ is a rotation quaternion that defines coordinate transformation from the s-frame to the e-frame
- $\mathbf{f}_s$ is specific force
- $\breve{\boldsymbol{\omega}}_{is}$ is a quaternion, whose scalar part is zero, and vector part is angular rate of the sensor frame relative to the quasi-inertial coordinate frame
- $\breve{\boldsymbol{\omega}}_{ie}$ is a quaternion, whose scalar part is zero, and vector part is angular rate of the e-frame relative to the i-frame
- $\mathbf{g}_e(\mathbf{x}_e)$ is gravity vector

Note that in the (2.1) all the quantities are considered to be error-free. So, the vectors $\mathbf{x}_e$ and $\mathbf{v}_e$ are true velocity and position of the vehicle, while the quaternion $\breve{\mathbf{q}}_{es}$ defines the rotation from the s- to the e-frame without errors.

2.5 Linearized error propagation

2.5.1 Dynamics of the estimated state

Specific force and angular rate, measured by an IMU, are never ideal and deviate from true values due to measurement noise and other instrumental errors. Thus, integrating these inputs numerically causes estimates of position, velocity, and orientation being slightly erroneous. The estimates are subject to

$$\begin{aligned}\dot{\tilde{\mathbf{x}}}_e &= \tilde{\mathbf{v}}_e \\ \dot{\tilde{\mathbf{v}}}_e &= -2[\boldsymbol{\omega}_{ie}\times]\tilde{\mathbf{v}}_e + \mathbf{g}_e(\tilde{\mathbf{x}}_e) + \mathbf{C}_{\tilde{e}s}(\breve{\mathbf{q}}_{\tilde{e}s})\tilde{\mathbf{f}}_s \\ \dot{\breve{\mathbf{q}}}_{\tilde{e}s} &= \frac{1}{2}\left(\breve{\mathbf{q}}_{\tilde{e}s} * \breve{\tilde{\boldsymbol{\omega}}}_{is} - \breve{\boldsymbol{\omega}}_{ie} * \breve{\mathbf{q}}_{\tilde{e}s}\right)\end{aligned} \tag{2.2}$$

The only difference between the (2.2) and (2.1) are tilde symbols "~" above variables. These symbols emphasize that numerical integration of measured specific force and angular rate will result into position, velocity, and orientation values that deviate from truth.

Some more words need to be said about the estimated rotation from the s-frame to the e-frame. Consider the result of numerical integration $\breve{\tilde{\mathbf{q}}}_{es}$. There are two possible ways to use this rotation quaternion or the rotation matrix derived from it.

Firstly, this rotation can be used to convert vector coordinates from the s-frame to the e-frame. However, rotating from the true sensor frame using the estimated rotation matrix will result into coordinates being transformed into some erroneous coordinate frame $\tilde{e}$ that is turned slightly relatively to the true e-frame. Thus, the quaternion $\breve{\tilde{\mathbf{q}}}_{es}$ can be understood as the quaternion that defines a transformation from the true sensor frame to the computed ECEF coordinate frame. Therefore, $\breve{\tilde{\mathbf{q}}}_{es} = \breve{\mathbf{q}}_{\tilde{e}s}$.

Secondly, inverse of the quaternion $\breve{\tilde{\mathbf{q}}}_{es}$ can be employed to transform vector coordinates from the e-frame to the s-frame. However, rotating from the true ECEF frame using the estimated rotation matrix will result into coordinates being transformed into some erroneous coordinate frame $\tilde{s}$ that is turned slightly relatively to the true sensor frame. So, the quaternion $\breve{\tilde{\mathbf{q}}}_{es}$ can be understood as the

quaternion that defines a transformation from the $\tilde{s}$-frame to the true ECEF coordinate frame. Thus, $\widetilde{\mathbf{q}}_{es} = \mathbf{q}_{e\tilde{s}}$.
Note that the values of all the quaternions $\widetilde{\mathbf{q}}_{es}$, $\mathbf{q}_{\tilde{e}s}$, and $\mathbf{q}_{e\tilde{s}}$ are equal. They are denoted differently only for the sake of emphasizing various physical interpretations.

2.5.2 Position error dynamics

Define position and velocity errors as

$$\begin{aligned} \mathbf{x}_e &= \tilde{\mathbf{x}}_e + \delta\mathbf{x}_e \\ \mathbf{v}_e &= \tilde{\mathbf{v}}_e + \delta\mathbf{v}_e \end{aligned} \tag{2.3}$$

Dynamics of the true and estimated position is given by

$$\begin{aligned} \dot{\mathbf{x}}_e &= \mathbf{v}_e \\ \dot{\tilde{\mathbf{x}}}_e &= \tilde{\mathbf{v}}_e \end{aligned} \tag{2.4}$$

Subtracting the second line in the (2.4) from the first yields

$$\dot{\mathbf{x}}_e - \dot{\tilde{\mathbf{x}}}_e = \mathbf{v}_e - \tilde{\mathbf{v}}_e \tag{2.5}$$

Consequently,

$$\delta\dot{\mathbf{x}}_e = \delta\mathbf{v}_e \tag{2.6}$$

2.5.3 Velocity error dynamics

Define an error of the measured specific force as

$$\mathbf{f}_s = \tilde{\mathbf{f}}_s + \delta\mathbf{f}_s \tag{2.7}$$

Consider the estimated rotation matrix $\mathbf{C}_{\tilde{e}s}$ (derived from quaternion). The true matrix $\mathbf{C}_{es}$ is related to the estimates as

$$\mathbf{C}_{es} = \mathbf{C}_{e\tilde{e}}\mathbf{C}_{\tilde{e}s} \tag{2.8}$$

Herein $\mathbf{C}_{e\tilde{e}}$ is the matrix that defines transformation from the estimated ECEF frame to the true one. If errors are small, then

$$\mathbf{C}_{e\tilde{e}} \approx \mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}}\times] \tag{2.9}$$

and the orientation error vector $\boldsymbol{\psi}_{e\tilde{e}}$ can be understood as a vector of "small" Euler angles between the coordinate frames $\tilde{e}$ and e.
Dynamics of the true and estimated position is subject to

$$\begin{aligned} \dot{\mathbf{v}}_e &= -2[\boldsymbol{\omega}_{ie}\times]\mathbf{v}_e + \mathbf{g}_e(\mathbf{x}_e) + \mathbf{C}_{es}\mathbf{f}_s \\ \dot{\tilde{\mathbf{v}}}_e &= -2[\boldsymbol{\omega}_{ie}\times]\tilde{\mathbf{v}}_e + \mathbf{g}_e(\tilde{\mathbf{x}}_e) + \mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s \end{aligned} \tag{2.10}$$

Subtracting the second line in the (2.10) from the first yields

$$\begin{aligned} \dot{\mathbf{v}}_e - \dot{\tilde{\mathbf{v}}}_e = \delta\dot{\mathbf{v}}_e = &-2[\boldsymbol{\omega}_{ie}\times]\mathbf{v}_e + \mathbf{g}_e(\mathbf{x}_e) + \mathbf{C}_{es}\mathbf{f}_s + 2[\boldsymbol{\omega}_{ie}\times]\tilde{\mathbf{v}}_e \\ &- \mathbf{g}_e(\tilde{\mathbf{x}}_e) - \mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s \end{aligned} \tag{2.11}$$

Employing error definitions allows to decompose true quantities in (2.11) into combinations of estimates/measurements and errors. Consequently,

$$\begin{aligned} \delta\dot{\mathbf{v}}_e = &-2[\boldsymbol{\omega}_{ie}\times](\tilde{\mathbf{v}}_e + \delta\mathbf{v}_e) + \mathbf{g}_e(\tilde{\mathbf{x}}_e + \delta\mathbf{x}_e) \\ &+ (\mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}}\times])\mathbf{C}_{\tilde{e}s}\left(\tilde{\mathbf{f}}_s + \delta\mathbf{f}_s\right) + 2[\boldsymbol{\omega}_{ie}\times]\tilde{\mathbf{v}}_e \\ &- \mathbf{g}_e(\tilde{\mathbf{x}}_e) - \mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s \end{aligned} \tag{2.12}$$

Doing Taylor series expansion of $\mathbf{g}_e(\mathbf{x}_e)$ and neglecting products of errors as second order terms allows to reduce (2.12) to

$$\delta\dot{\mathbf{v}}_e = \left.\frac{\partial\mathbf{g}_e(\mathbf{x}_e)}{\partial(\mathbf{x}_e)^{\mathrm{T}}}\right|_{\tilde{\mathbf{x}}_e} \cdot \delta\mathbf{x}_e - 2[\boldsymbol{\omega}_{ie}\times]\cdot\delta\mathbf{v}_e - \left[\left(\mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s\right)\times\right]\cdot\boldsymbol{\psi}_{e\tilde{e}} + \mathbf{C}_{\tilde{e}s}\cdot\delta\mathbf{f}_s \tag{2.13}$$

2.5.4 Orientation error differential equation

Rearrange the definition (2.8). Then

$$\mathbf{C}_{es}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} = \mathbf{C}_{e\tilde{e}} \approx \mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}}\times] \tag{2.14}$$

Differentiating the left and the right parts of the equality (2.14) yields

$$[\dot{\boldsymbol{\psi}}_{e\tilde{e}}\times] = \dot{\mathbf{C}}_{es}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} + \mathbf{C}_{es}\dot{\mathbf{C}}_{\tilde{e}s}^{\mathrm{T}} \tag{2.15}$$

Then, exploiting a rule of differentiation of a rotation matrix,

$$\begin{aligned}[\dot{\boldsymbol{\psi}}_{e\tilde{e}}\times] = (\mathbf{C}_{es}[\boldsymbol{\omega}_{is}\times] - [\boldsymbol{\omega}_{ie}\times]\mathbf{C}_{es})\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} \\ + \mathbf{C}_{es}\left(-[\tilde{\boldsymbol{\omega}}_{is}\times]\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} + \mathbf{C}_{\tilde{e}s}^{\mathrm{T}}[\boldsymbol{\omega}_{ie}\times]\right)\end{aligned} \tag{2.16}$$

Now true quantities need to be substituted by compositions of estimated/measured quantities and errors. Thus,

$$\begin{aligned}[\dot{\boldsymbol{\psi}}_{e\tilde{e}}\times] = \big((\mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}}\times])\mathbf{C}_{\tilde{e}s}[(\tilde{\boldsymbol{\omega}}_{is} + \delta\boldsymbol{\omega}_{is})\times] \\ - [\boldsymbol{\omega}_{ie}\times](\mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}}\times])\mathbf{C}_{\tilde{e}s}\big)\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} \\ + (\mathbf{I} + [\boldsymbol{\psi}_{e\tilde{e}}\times])\mathbf{C}_{\tilde{e}s}\left(-[\tilde{\boldsymbol{\omega}}_{is}\times]\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} + \mathbf{C}_{\tilde{e}s}^{\mathrm{T}}[\boldsymbol{\omega}_{ie}\times]\right)\end{aligned} \tag{2.17}$$

Expanding brackets and neglecting error products as second order terms allows to reduce (2.17) to the form

$$[\dot{\boldsymbol{\psi}}_{e\tilde{e}}\times] = -[\boldsymbol{\omega}_{ie}\times][\boldsymbol{\psi}_{e\tilde{e}}\times] + [\boldsymbol{\psi}_{e\tilde{e}}\times][\boldsymbol{\omega}_{ie}\times] + \mathbf{C}_{\tilde{e}s}[\delta\boldsymbol{\omega}_{is}\times]\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} \tag{2.18}$$

Finally, using Jokobi identity and properties of $[(\cdot)\times]$ operator, we transform (2.18) to the form

$$[\dot{\boldsymbol{\psi}}_{e\tilde{e}}\times] = -[\boldsymbol{\omega}_{ie}\times][\boldsymbol{\psi}_{e\tilde{e}}\times] + [\mathbf{C}_{\tilde{e}s}\delta\boldsymbol{\omega}_{is}\times] \tag{2.19}$$

This matrix equation is equivalent to

$$\dot{\boldsymbol{\psi}}_{e\tilde{e}} = -[\boldsymbol{\omega}_{ie}\times]\cdot\boldsymbol{\psi}_{e\tilde{e}} + \mathbf{C}_{\tilde{e}s}\cdot\delta\boldsymbol{\omega}_{is} \tag{2.20}$$

2.5.5 INS error propagation: summary

Results obtained in the previous sections can be united into INS error propagation equations given below.

$$\begin{aligned}\delta\dot{\mathbf{x}}_e &= \delta\mathbf{v}_e \\ \dot{\mathbf{v}}_e &= \left.\frac{\partial\mathbf{g}_e(\mathbf{x}_e)}{\partial(\mathbf{x}_e)^{\mathrm{T}}}\right|_{\tilde{\mathbf{x}}_e}\cdot\delta\mathbf{x}_e - 2[\boldsymbol{\omega}_{ie}\times]\cdot\delta\mathbf{v}_e - \left[\left(\mathbf{C}_{\tilde{e}s}\tilde{\mathbf{f}}_s\right)\times\right]\cdot\boldsymbol{\psi}_{e\tilde{e}} + \mathbf{C}_{\tilde{e}s}\cdot\delta\mathbf{f}_s \\ \dot{\boldsymbol{\psi}}_{e\tilde{e}} &= -[\boldsymbol{\omega}_{ie}\times]\cdot\boldsymbol{\psi}_{e\tilde{e}} + \mathbf{C}_{\tilde{e}s}\cdot\delta\boldsymbol{\omega}_{is}\end{aligned} \tag{2.21}$$

Note that there are multiple other versions of INS error propagation equations. They may differ in

- Definition of orientation error
- INS mechanization
- Choice of coordinate frames
- Physical meaning of some variables

All alternatives that can be found in literature are valid and have both advantages and disadvantages. Comparison of approaches will not be discussed hear.

2.6 IMU error model

In the InsGnssFilterLoose demo we assume that accelerometer and gyroscope measurements are related to the true specific force and angular rate as

$$\begin{aligned}\mathbf{f}_s &= \mathrm{diag}(\mathbf{s}_f)\cdot\tilde{\mathbf{f}}_s + \mathbf{b}_f + \boldsymbol{\nu}_f \\ \boldsymbol{\omega}_{is} &= \mathrm{diag}(\mathbf{s}_\omega)\cdot\tilde{\boldsymbol{\omega}}_{is} + \mathbf{b}_\omega + \boldsymbol{\nu}_\omega\end{aligned} \tag{2.22}$$

The equations (2.22) should be understood almost as identities: provided that we know true scale factors, offsets, and noise, it is possible to derive true specific force and angular rate from IMU measurements. However, as sensors are never perfectly calibrated, true scale factors and biases are

not available. Instead, there are estimates $\hat{\mathbf{s}}_f$, $\hat{\mathbf{b}}_f$, $\hat{\mathbf{s}}_\omega$, $\hat{\mathbf{b}}_\omega$ that have some deviation from truth. In addition to that sensor noise cannot be compensated because it is random.
So, a user can only do the following compensation:

$$\begin{aligned}\hat{\mathbf{f}}_s &= \mathrm{diag}(\hat{\mathbf{s}}_f) \cdot \tilde{\mathbf{f}}_s + \hat{\mathbf{b}}_f \\ \hat{\boldsymbol{\omega}}_{is} &= \mathrm{diag}(\hat{\mathbf{s}}_\omega) \cdot \tilde{\boldsymbol{\omega}}_{is} + \hat{\mathbf{b}}_\omega\end{aligned} \tag{2.23}$$

Residual errors of the compensated specific force and angular rate can be expressed in terms of noise and the residual errors of the scale factors and offset as

$$\begin{aligned}\delta\hat{\mathbf{f}}_s &= \mathbf{f}_s - \hat{\mathbf{f}}_s = \mathrm{diag}(\tilde{\mathbf{f}}_s) \cdot \delta\mathbf{s}_f + \delta\mathbf{b}_f + \boldsymbol{\nu}_f \\ \delta\hat{\boldsymbol{\omega}}_{is} &= \boldsymbol{\omega}_{is} - \hat{\boldsymbol{\omega}}_{is} = \mathrm{diag}(\tilde{\boldsymbol{\omega}}_{is}) \cdot \delta\mathbf{s}_\omega + \delta\mathbf{b}_\omega + \boldsymbol{\nu}_\omega\end{aligned} \tag{2.24}$$

2.7 Measurement equations

2.7.1 GNSS position observation

In smartphones GNSS antenna and IMU are mounted within centimeters from each other. In fact, separation of the antenna and IMU appears to be much smaller than typical achievable accuracy of GNSS positioning done by a smartphone. Thus, the lever arm between IMU and GNSS antenna can be ignored.
An ideal GNSS receiver would provide a measurement of a smartphone's position:

$$\mathbf{y}_{pos} = \mathbf{x}_e \tag{2.25}$$

However, in reality the measurement is not error free, so the measured position relates to the true position as

$$\tilde{\mathbf{y}}_{pos} = \mathbf{x}_e + \delta\mathbf{y}_{pos} \tag{2.26}$$

A residual can be derived as follows.

$$\Delta\mathbf{y}_{pos} = \tilde{\mathbf{y}}_{pos} - \tilde{\mathbf{x}}_e = \delta\mathbf{x}_e + \delta\mathbf{y}_{pos} \tag{2.27}$$

The equation (2.27) relates a residual (deviation between the measurement and an approximation of the measurement derived from navigation state) to INS error state. Such equations are often called *linearized measurement equations*.

2.7.2 Non-holonomic constraints

Recall the car coordinate frame introduced in the Section 2.3.5. This coordinate frame has got an important property: when a car drives straight, projections of the velocity vector onto the second and the third axes of the c-frame are zeros. This property comes from vehicle dynamics and can be exploited in a form of virtual measurements.
Assume that projections of velocity onto car frame can be measured. Then the ideal measurement will be related to the INS state vector as

$$\mathbf{y}_{vel,c} = \mathbf{C}_{cs}\mathbf{C}_{es}^{\mathrm{T}}\mathbf{v}_e \tag{2.28}$$

Here $\mathbf{C}_{cs}$ is a matrix that defines a rotation from the s-frame to the c-frame.
Measured velocity is not error-free, thus

$$\tilde{\mathbf{y}}_{vel,c} = \mathbf{C}_{cs}\mathbf{C}_{es}^{\mathrm{T}}\mathbf{v}_e + \delta\mathbf{y}_{vel,c} \tag{2.29}$$

Assume that an approximation or an estimate of $\mathbf{C}_{cs}$ is known. This approximation can be understood as a transformation $\mathbf{C}_{\tilde{c}s}$ from the true sensor frame to some erroneous car frame $\tilde{c}$.
The estimated rotation matrix $\mathbf{C}_{\tilde{c}s}$ can be decomposed into a combination of the true value and an error using the approach described in the Section 2.5.4.
Thus,

$$\begin{aligned}\mathbf{C}_{cs} &= \mathbf{C}_{c\tilde{c}}\mathbf{C}_{\tilde{c}s} \\ \mathbf{C}_{c\tilde{c}} &\approx \mathbf{I} + [\boldsymbol{\psi}_{c\tilde{c}}\times]\end{aligned} \tag{2.30}$$

A residual can be formed as

$$\Delta\mathbf{y}_{vel,c} = \tilde{\mathbf{y}}_{vel,c} - \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}\tilde{\mathbf{v}}_e = \mathbf{C}_{cs}\mathbf{C}_{es}^{\mathrm{T}}\mathbf{v}_e - \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}\tilde{\mathbf{v}}_e + \delta\mathbf{y}_{vel,c} \tag{2.31}$$

Decomposing the true quantities into estimates and errors yields

$$\begin{aligned}\Delta\mathbf{y}_{vel,c} = (\mathbf{I} + [\boldsymbol{\psi}_{c\tilde{c}}\times])\mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}(\mathbf{I} - [\boldsymbol{\psi}_{e\tilde{e}}\times])(\tilde{\mathbf{v}}_e + \delta\mathbf{v}_e) - \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}\tilde{\mathbf{v}}_e \\ + \delta\mathbf{y}_{vel,c}\end{aligned} \tag{2.32}$$

Finally, after brackets expansion and linearization,

$$\Delta\mathbf{y}_{vel,c} = \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} \cdot \delta\mathbf{v}_e + \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}[\tilde{\mathbf{v}}_e\times] \cdot \boldsymbol{\psi}_{e\tilde{e}} - \left[\left(\mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}\tilde{\mathbf{v}}_e\right)\times\right] \cdot \boldsymbol{\psi}_{c\tilde{c}} + \delta\mathbf{y}_{vel,c} \tag{2.33}$$

2.8 Kalman filter

We will not recapitulate conventional formulas of a discrete Kalman filter. A reader is encouraged to visit dedicated Wikipedia page, read famous books (Grewal, 2014), (Jazwinski, 2007), (Farrell, 2008), and hundreds of other, or Google for tutorials.

However, we would like to mention here a very helpful and important property of a Kalman filter that often remains unnoticed. A very good demonstration of this property is available in (Farrell, 2008).

Assume that covariance matrix $\mathbf{R}_k$ of the noise of the vector measurement $\mathbf{y}$ is a *diagonal matrix*. In other words, we assume that noises of components of the vector $\mathbf{y}$ are independent (and thus uncorrelated). Then instead of one Kalman filter update with a vector measurement $\mathbf{y}$ it is possible to do m consecutive updates using *scalar elements* of the vector $\mathbf{y}$. The result will be the same.

This "feature" allows to avoid numerical matrix inversion during Kalman filter update. So, computations become much simpler and more robust.

2.9 Discretization of continuous systems

Consider linear stochastic system

$$\dot{\mathbf{z}} = \mathbf{A}(t)\mathbf{z} + \mathbf{G}(t)\mathbf{q} \tag{2.34}$$

where $\mathbf{q}$ is zero-mean Gaussian white noise input with covariance $\mathbf{Q}(t)$.

The continuous system (2.34) is equivalent to the discrete system

$$\mathbf{z}_k = \boldsymbol{\Phi}_{k+1,k}\mathbf{z}_k + \mathbf{q}_k^* \tag{2.35}$$

Therein $\boldsymbol{\Phi}_{k+1,k} = \boldsymbol{\Phi}(t_{k+1}, t_k)$ is a state transition matrix of the system (2.34). The matrix $\boldsymbol{\Phi}$ is the unique solution of the system

$$\frac{d\boldsymbol{\Phi}(t,t_0)}{dt} = \mathbf{A}(t)\boldsymbol{\Phi}(t,t_0), \boldsymbol{\Phi}(t_0,t_0) = \mathbf{I} \tag{2.36}$$

Note that a solution of (2.34) takes the form

$$\mathbf{z}(t) = \boldsymbol{\Phi}(t,t_0)\mathbf{z}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\mathbf{G}(\tau)\mathbf{q}(\tau)d\tau \tag{2.37}$$

Therefore, the term $\mathbf{q}_k^*$ in (2.35) can be written as

$$\mathbf{q}_k^* = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1},\tau)\mathbf{G}(\tau)\mathbf{q}(\tau)d\tau \tag{2.38}$$

It is possible to show that the sequence $\mathbf{q}_k^*$ presents zero-mean Gaussian white noise with covariance

$$\mathbf{Q}_k^* = \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1},\tau)\mathbf{G}(\tau)\mathbf{Q}(\tau)\mathbf{G}^{\mathrm{T}}(\tau)\boldsymbol{\Phi}^{\mathrm{T}}(t_{k+1},\tau)d\tau \tag{2.39}$$

A short proof of the statement above and a derivation of the formula (2.39) can be found in (Farrell, 2008) and in (Baklanov, 2017). Note that these are not the only sources.
To implement INS error estimation in a form of a discrete Kalman filter, system transition matrix and discrete noise covariance matrices are required. Commonly, the transition matrix of the system (2.34) is approximated as follows.

$$\mathbf{\Phi}_{k+1,k} \approx \mathbf{I} + (t_{k+1} - t_k) \cdot \mathbf{A}(t_k) \tag{2.40}$$

Alternatively, one can use the second order approximation

$$\mathbf{\Phi}_{k+1,k} \approx \mathbf{I} + (t_{k+1} - t_k) \cdot \mathbf{A}(t_k) + \frac{(t_{k+1} - t_k)^2}{2}\mathbf{A}^2(t_k) \tag{2.41}$$

The simplest way to approximate the integral in (2.39) is

$$\mathbf{Q}_k^* \approx (t_{k+1} - t_k) \cdot \mathbf{G}(t_k)\mathbf{Q}(t_k)\mathbf{G}^{\mathrm{T}}(t_k) \tag{2.42}$$

The formula (2.42) is obtained from the (2.39) by application of rectangle rule of integration. It is also possible to get better approximation using trapezoidal rule.

2.10 Measurement decorrelation

In the Section 2.8 it was mentioned that Kalman filter update is very easy to implement when measurement noise is uncorrelated. However, this is often not the case.
Nevertheless, there is a workaround that makes possible to pass uncorrelated measurements to the Kalman filter.
Consider a measurement

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{x} + \mathbf{r} \tag{2.43}$$

Assume that $\mathbf{r}$ is zero-mean Gaussian white noise with covariance matrix $\mathbf{R}$ that is strictly positive definite.
The matrix $\mathbf{R}$ can be decomposed into Cholesky factors:

$$\mathbf{R} = \mathbf{U}^{\mathrm{T}}\mathbf{U} \tag{2.44}$$

Consider now a transformed measurement $\mathbf{y}_1 = \mathbf{T}\mathbf{y}$, where

$$\mathbf{T} = (\mathbf{U}^{\mathrm{T}})^{-1} \tag{2.45}$$

Then

$$\mathbf{y}_1 = \mathbf{H}_1 \cdot \mathbf{x} + \mathbf{r}_1 \tag{2.46}$$

Where

$$\begin{aligned} \mathbf{H}_1 &= \mathbf{T}\mathbf{H} \\ \mathbf{r}_1 &= \mathbf{T}\mathbf{r} \end{aligned} \tag{2.47}$$

Covariance of the transformed noise $\mathbf{r}_1$ is identity matrix, so noise of the transformed measurement $\mathbf{y}_1$ appears uncorrelated.
Based on the above stated, the following procedure for dealing with correlated measurements can be proposed:

1. Decorrelate original vector measurement using transformation derived from Cholesky factors of the noise covariance matrix
2. Do Kalman filter updates sequentially using scalar components of the transformed measurement vector

3 Design and architecture of the filter InsGnssFilterLoose

3.1 Augmented state vector

In the navigation filter we would like to estimate sensor compensation parameters: scale factors and biases. Then, the core INS state vector needs to be augmented with the corresponding parameters. In addition to that, we will estimate alignment between the IMU frame (s-frame) and the car-fixed coordinate frame. Finally, augmented state vector will take the form

$$\mathbf{z} = \begin{pmatrix} (\mathbf{x}_e)^{\mathrm{T}} & (\mathbf{v}_e)^{\mathrm{T}} & (\mathbf{q}_{es})^{\mathrm{T}} & (\mathbf{b}_f)^{\mathrm{T}} & (\mathbf{b}_\omega)^{\mathrm{T}} & (\mathbf{s}_f)^{\mathrm{T}} & (\mathbf{s}_\omega)^{\mathrm{T}} & (\mathbf{q}_{cs})^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}} \tag{3.1}$$

Note that $\dim \mathbf{z} = 26$.

3.2 Dynamics of the augmented state

To propagate the state vector, we will use IMU measurement compensated according to (2.23). Thus, the estimate of the state vector (3.1) is subject to

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}_e &= \tilde{\mathbf{v}}_e \\ \dot{\tilde{\mathbf{v}}}_e &= -2[\boldsymbol{\omega}_{ie}\times]\tilde{\mathbf{v}}_e + \mathbf{g}_e(\tilde{\mathbf{x}}_e) + \mathbf{C}_{\tilde{e}s}\hat{\mathbf{f}}_s \\ \dot{\breve{\mathbf{q}}}_{\tilde{e}s} &= \frac{1}{2}\left(\breve{\mathbf{q}}_{\tilde{e}s} * \breve{\tilde{\boldsymbol{\omega}}}_{is} - \breve{\boldsymbol{\omega}}_{ie} * \breve{\mathbf{q}}_{\tilde{e}s}\right) \\ \dot{\hat{\mathbf{b}}}_f &= \mathbf{0} \\ \dot{\hat{\mathbf{b}}}_\omega &= \mathbf{0} \\ \dot{\hat{\mathbf{s}}}_f &= \mathbf{0} \\ \dot{\hat{\mathbf{s}}}_\omega &= \mathbf{0} \\ \dot{\breve{\mathbf{q}}}_{\tilde{c}s} &= \mathbf{0} \end{aligned} \tag{3.2}$$

3.3 State propagation

The state vector is propagated by integrating numerically the ODE (3.2). Heun's method (2^{nd} order Runge-Kutta method with $\alpha = 1$) is employed. Discrete times forming the integration grid are taken from gyroscope measurements. A smartphone cannot ensure that accelerometer data and gyroscope data are collected simultaneously, so specific force needs to be interpolated or approximated at the timestamps of gyroscope measurements.

3.4 Error state vector

An error state vector that corresponds to the state vector (3.1) is given by

$$\delta\mathbf{z} = \begin{pmatrix} (\delta\mathbf{x}_e)^{\mathrm{T}} & (\delta\mathbf{v}_e)^{\mathrm{T}} & (\boldsymbol{\psi}_{e\tilde{e}})^{\mathrm{T}} & (\delta\mathbf{b}_f)^{\mathrm{T}} & (\delta\mathbf{b}_\omega)^{\mathrm{T}} & (\delta\mathbf{s}_f)^{\mathrm{T}} & (\delta\mathbf{s}_\omega)^{\mathrm{T}} & (\boldsymbol{\psi}_{c\tilde{c}})^{\mathrm{T}} \end{pmatrix}^{\mathrm{T}} \tag{3.3}$$

Note that $\dim \delta\mathbf{z} = 24$ and $\dim \delta\mathbf{z} \neq \dim \mathbf{z}$.

3.5 Error state differential equation

Dynamics of the error state is subject to linear stochastic differential equation given below. This equation can be obtained by modeling sensor errors as random walk and by substituting in the formula (2.21) the symbols $\delta\mathbf{f}_s$ and $\delta\boldsymbol{\omega}_{is}$ with the expression (2.24).

$$\delta\dot{\mathbf{z}} = \mathbf{A}(t)\cdot\delta\boldsymbol{z} + \mathbf{G}(t)\cdot\boldsymbol{\nu} \tag{3.4}$$

Herein,

$$\mathbf{A}(t)=\begin{pmatrix}\mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & -2[\boldsymbol{\omega}_{ie}\times] & -[(\mathbf{C}_{\tilde{e}s}\hat{\mathbf{f}}_s)\times] & \mathbf{C}_{\tilde{e}s} & \mathbf{0}_{3\times3} & \mathbf{C}_{\tilde{e}s}\mathrm{diag}(\tilde{\mathbf{f}}_s) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -[\boldsymbol{\omega}_{ie}\times] & \mathbf{0}_{3\times3} & \mathbf{C}_{\tilde{e}s} & \mathbf{0}_{3\times3} & \mathbf{C}_{\tilde{e}s}\mathrm{diag}(\widetilde{\boldsymbol{\omega}}_{is}) & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\end{pmatrix} \quad (3.5)$$

Note the difference between $\hat{\mathbf{f}}_s$ and $\tilde{\mathbf{f}}_s$! *<u>They are not the same.</u>*
Note that $\widetilde{\boldsymbol{\omega}}_{is}$ is *<u>uncompensated</u>* gyroscope measurement!

$$\boldsymbol{\nu}=\left((\boldsymbol{\nu}_f)^{\mathrm{T}} \quad (\boldsymbol{\nu}_\omega)^{\mathrm{T}} \quad (\boldsymbol{\nu}_{bf})^{\mathrm{T}} \quad (\boldsymbol{\nu}_{b\omega})^{\mathrm{T}} \quad (\boldsymbol{\nu}_{sf})^{\mathrm{T}} \quad (\boldsymbol{\nu}_{s\omega})^{\mathrm{T}} \quad (\boldsymbol{\nu}_{\psi_{c\tilde{c}}})^{\mathrm{T}}\right)^{\mathrm{T}} \quad (3.6)$$

$$\mathbf{G}(t)=\begin{pmatrix}\mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{C}_{\tilde{e}s} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{C}_{\tilde{e}s} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3}\end{pmatrix} \quad (3.7)$$

In the (3.6) $\boldsymbol{\nu}_f$ and $\boldsymbol{\nu}_\omega$ are noises of the accelerometers and gyroscopes, respectively. $\boldsymbol{\nu}_{bf}$, $\boldsymbol{\nu}_{b\omega}$, $\boldsymbol{\nu}_{sf}$, $\boldsymbol{\nu}_{s\omega}$, and $\boldsymbol{\nu}_{\psi_{c\tilde{c}}}$ are the noises of sensor biases and scale factors, and the noise of misalignment between IMU and a car.

3.6 Transition matrix

System transition matrix is approximated using the rule (2.40) and the expression (3.5) for the matrix $\mathbf{A}(t)$.

3.7 Discrete noise covariance matrix

Discrete noise covariance is derived using the rule (2.42), where $\mathbf{G}(t)$ is given by (3.7) and

$$\mathbf{Q}(t)=\mathrm{Cov}[\boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}}] \quad (3.8)$$

Note that in the implementation $\mathrm{Cov}[\boldsymbol{\nu}\boldsymbol{\nu}^{\mathrm{T}}]$ is a diagonal matrix. Its first 6 elements need to be set according to IMU white noise parameters. Elements, that correspond to noises of misalignment, scale factors, and offsets are set to some small values.

3.8 Residual computation and measurement matrices

3.8.1 GNSS position measurement

3.8.1.1 Residual

$$\Delta\mathbf{y}_{pos}=\tilde{\mathbf{y}}_{pos}-\tilde{\mathbf{x}}_e=\delta\mathbf{x}_e+\delta\mathbf{y}_{pos} \quad (3.9)$$

3.8.1.2 Measurement matrix

$$\mathbf{H}_{pos}=\frac{\partial\Delta\mathbf{y}_{pos}}{\partial(\delta\mathbf{z})^{\mathrm{T}}}=(\mathbf{I}_{3\times3} \quad \mathbf{0}_{3\times21}) \quad (3.10)$$

3.8.2 Non-holonomic constraints

3.8.2.1 Residual

The second and the third elements of the following vector are used.

$$\begin{aligned}\Delta\mathbf{y}_{vel,c} = \tilde{\mathbf{y}}_{vel,c} - \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}\tilde{\mathbf{v}}_e \\ = \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} \cdot \delta\mathbf{v}_e + \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}[\tilde{\mathbf{v}}_e \times] \cdot \boldsymbol{\psi}_{e\tilde{e}} \\ - \left[\left(\mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}\tilde{\mathbf{v}}_e\right) \times\right] \cdot \boldsymbol{\psi}_{c\tilde{c}} + \delta\mathbf{y}_{vel,c}\end{aligned} \tag{3.11}$$

3.8.2.2 Measurement matrix

The second and the third rows of the following matrix need to be used.

$$\mathbf{H}_{vel,c} = \frac{\partial\Delta\mathbf{y}_{vel,c}}{\partial(\delta\mathbf{z})^{\mathrm{T}}} = \begin{pmatrix}\mathbf{0}_{3\times3} & \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}} & \mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}[\tilde{\mathbf{v}}_e \times] & \mathbf{0}_{3\times12} & -\left[\left(\mathbf{C}_{\tilde{c}s}\mathbf{C}_{\tilde{e}s}^{\mathrm{T}}\tilde{\mathbf{v}}_e\right) \times\right]\end{pmatrix} \tag{3.12}$$

3.9 Kalman filter realization

Kalman filter updates are done in a scalar-wise manner as explained in the Section 2.8. GNSS position measurements appear correlated and thus they get transformed using the approach described in the Section 2.10. A simple chi-square outlier detection is performed prior to Kalman filter update.

4 Bibliography

Baklanov, F. (2017). *Platform-autonomous Fault-tolerant Attitude/Heading Reference and Navigation Systems.* München: Technische Universität München.

Farrell, J. (2008). *Aided Navigation: GPS with High Rate Sensors.* McGraw-Hill.

Grewal, M. S. (2014). *Kalman Filtering: Theory and Practice with MATLAB.* Wiley-IEEE Press.

Jazwinski, A. H. (2007). *Stochastic Processes and Filtering Theory.* Mineola, New York: Dover Publications, Inc.