Navigation equations

Master's Degree in Geomatics Engineering and Geoinformation

Academic Year 2017-2018

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

Navigation equations

- ✓Introduction
- ✓Unified approach
- ✓Particularisations
  - *i-frame*
  - *e-frame*
  - *n-frame*
- ✓Numerical integration of navigation equations
- ✓Jekeli 4.3 pp 123

Introduction

The navigation equations describe the dynamics of moving points.

Navigation or positioning using an inertial navigation system is based fundamentally on the integration of inertially sensed accelerations with respect to time.

It involves the solution to a differential equation that relates sensed accelerations to second-order time-derivatives of position.

For example, in an inertial frame (non-rotating) they are

$$\ddot{\vec{x}}^i = \vec{g}^i(\vec{x}^i) + \vec{a}^i$$

$\ddot{\vec{x}}^i$	second time derivative of the position vector
$\vec{x}^i$	position vector in the *i-frame*
$\vec{g}^i$	acceleration vector due to gravity
$\vec{a}^i$	specific force sensed by the INS

Introduction

Obtaining positions and velocities by integrating these equations is known as free-inertial navigation. Except for initial conditions, it does not need any external information.

The formulation and subsequent method of solution of the navigation equations depend specifically on the coordinate frame in which we wish to determine positions.

The particular mechanization, whether stabilized or strapdown, is immaterial in the formulation of the navigation equations, since these mechanizations differ only in how the transformation from the instrument frame to the navigation frame is effected → same trajectory

It is desirable to formulate the navigation equations rigorously and in a unified way using an arbitrary frame (***a-frame***) and then particularise them to the frame of interest (***i-frame, e-frame***, ***n-frame***).

Unified approach

Let the ***a-frame*** be a completely arbitrary frame that rotates with respect to the inertial frame with angular rate $\vec{\omega}^a_{ia}$

The objective is to derive an equation like

$$\ddot{\vec{x}}^i = \vec{g}^i(\vec{x}^i) + \vec{a}^i$$

But for positions $\vec{x}^a$ in the *a-frame*.

We suppose that both the *a-frame* and the *i-frame* are concentric.

A vector in the *a-frame* has coordinates in the *i-frame* given by

$$\vec{x}^i = C^i_a \vec{x}^a \qquad \left\{ \begin{aligned} \dot{\vec{x}}^i &= \dot{C}^i_a \vec{x}^a + C^i_a \dot{\vec{x}}^a \\ \ddot{\vec{x}}^i &= C^i_a \ddot{\vec{x}}^a + 2\dot{C}^i_a \dot{\vec{x}}^a + \ddot{C}^i_a \vec{x}^a \end{aligned} \right.$$

Unified approach

$$\dot{C}^i_a = C^i_a \Omega^a_{ia} \quad \Rightarrow \quad \ddot{C}^i_a = C^i_a(\dot{\Omega}^a_{ia} + \Omega^a_{ia}\Omega^a_{ia})$$

$$\ddot{\vec{x}}^i = C^i_a \ddot{\vec{x}}^a + 2C^i_a \Omega^a_{ia} \dot{\vec{x}}^a + C^i_a(\dot{\Omega}^a_{ia} + \Omega^a_{ia}\Omega^a_{ia})\vec{x}^a = \vec{g}^i(\vec{x}^i) + \vec{a}^i$$

$$\boxed{\ddot{\vec{x}}^a = -2\Omega^a_{ia}\dot{\vec{x}}^a - (\dot{\Omega}^a_{ia} + \Omega^a_{ia}\Omega^a_{ia})\vec{x}^a + \vec{g}^a + \vec{a}^a}$$

$$(C^i_a)^{-1} = (C^i_a)^T = C^a_i \qquad \vec{g}^a = C^a_i \vec{g}^i \qquad \vec{a}^a = C^a_i \vec{a}^i$$

6 first-order differential equations

$$\boxed{\begin{aligned} \frac{d}{dt}\dot{\vec{x}}^a &= -2\Omega^a_{ia}\dot{\vec{x}}^a - (\dot{\Omega}^a_{ia} + \Omega^a_{ia}\Omega^a_{ia})\vec{x}^a + \vec{g}^a + \vec{a}^a \\ \frac{d}{dt}\vec{x}^a &= \dot{\vec{x}}^a \end{aligned}}$$

Unified approach

The forcing terms in these equations are the accelerations sensed by the accelerometers $\vec{a}^a$ and the gravitational acceleration $\vec{g}^a$.

If the system is stabilized such that the accelerometer platform is parallel to the *a-frame,* then the sensed accelerations are $\vec{a}^a$.

In the strapdown mechanization, these accelerations are computed from accelerometer data sensed in other frame (*b-frame,s-frame*) and they have to be transformed using

$$\vec{a}^a = C_b^a \vec{a}^b$$

where the transformation C_b^a is determined by integrating the angular rates $\vec{\omega}_{ib}^b$ obtained from the gyro data.

Navigation equations in the *i-frame*

In this case the ***a-frame*** is the ***i-frame*** $\Omega_{ia}^{a} = 0$ $\dot{\Omega}_{ia}^{a} = 0$

$$\begin{aligned} \frac{d}{dt}\dot{\vec{x}}^a &= -2\Omega_{ia}^a\dot{\vec{x}}^a - (\dot{\Omega}_{ia}^a + \Omega_{ia}^a\Omega_{ia}^a)\vec{x}^a + \vec{g}^a + \vec{a}^a \\ \frac{d}{dt}\vec{x}^a &= \dot{\vec{x}}^a \end{aligned}$$

$$\begin{aligned} \frac{d}{dt}\dot{\vec{x}}^i &= \vec{g}^i + \vec{a}^i \\ \frac{d}{dt}\vec{x}^i &= \dot{\vec{x}}^i \end{aligned}$$

+

***Strapdown** mechanisation*

$$\dot{C}_b^i = C_b^i\Omega_{ib}^b$$

Navigation equations in the *e-frame*

If the ***a-frame*** is defined to be the ***e-frame*** ⇨ $\dot{\Omega}^e_{ie} = 0$

$$\boxed{\begin{aligned} \frac{d}{dt}\dot{\vec{x}}^a &= -2\Omega^a_{ia}\dot{\vec{x}}^a - (\dot{\Omega}^a_{ia} + \Omega^a_{ia}\Omega^a_{ia})\vec{x}^a + \vec{g}^a + \vec{a}^a \\ \frac{d}{dt}\vec{x}^a &= \dot{\vec{x}}^a \end{aligned}}$$

⇩

$$\boxed{\begin{aligned} \frac{d}{dt}\dot{\vec{x}}^e &= -2\Omega^e_{ie}\dot{\vec{x}}^e - (\Omega^e_{ie}\Omega^e_{ie})\vec{x}^e + \vec{g}^e + \vec{a}^e \\ \frac{d}{dt}\vec{x}^e &= \dot{\vec{x}}^e \end{aligned}}$$

Navigation equations in the *e-frame*

$$
\begin{aligned}
\frac{d}{dt}\dot{\vec{x}}^e &= -2\Omega^e_{ie}\dot{\vec{x}}^e - (\Omega^e_{ie}\Omega^e_{ie})\vec{x}^e + \vec{g}^e + \vec{a}^e \\
\frac{d}{dt}\vec{x}^e &= \dot{\vec{x}}^e
\end{aligned}
$$

+

strapdown *mechanisation*

$$\dot{C}^e_b = C^e_b \Omega^b_{eb}$$

$$\vec{\omega}^b_{eb} = \vec{\omega}^b_{ib} - C^e_b \vec{\omega}^e_{ie}$$

$$\vec{\omega}^e_{ie} = \begin{pmatrix} 0 \\ 0 \\ \omega_e \end{pmatrix} \quad \Omega^e_{ie} = \begin{pmatrix} 0 & -\omega_e & 0 \\ \omega_e & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathrm{GRS80} \Rightarrow \omega_e = 729211510^{-11} rads^{-1}$$

Navigation equations in the *n-frame*

In this case, the ***a-frame*** cannot be simply substituded by the ***n-frame*** because the integration would take place in a frame whose axes would vary over time

The orientation of the ***n-frame*** with regard to the Earth are function of the geodetic cordinates (φ, λ)

The desired velocity vector is the ***e-frame*** (Earth-referenced) velocity vector coordinatized in a frame parallel to the ***n-frame***, which we denote as $\vec{v}^n$ and it is given by

$$\boxed{\vec{v}^n = C_e^n \dot{\vec{x}}^e}$$

NOTE ➡ $\vec{v}^n \neq \dot{\vec{x}}^n$

The corresponding equations are obtained from

$$\frac{d}{dt}\dot{\vec{x}}^e = -2\Omega_{ie}^e \dot{\vec{x}}^e - (\Omega_{ie}^e \Omega_{ie}^e)\vec{x}^e + \vec{g}^e + \vec{a}^e$$

Navigation equations in the *n-frame*

Considering $\vec{v}^n = C_e^n \dot{\vec{x}}^e \Rightarrow \dot{\vec{x}}^e = C_n^e \vec{v}^n$

$$\frac{d}{dt}\dot{\vec{x}}^e = \frac{d}{dt}\left(C_n^e \vec{v}^n\right) = \dot{C}_n^e \vec{v}^n + C_n^e \frac{d}{dt}\vec{v}^n$$

$$\dot{C}_n^e = C_n^e \Omega_{en}^n$$

$$\frac{d}{dt}\dot{\vec{x}}^e = C_n^e \left(\frac{d}{dt}\vec{v}^n + C_n^e \Omega_{en}^n \vec{v}^n\right)$$

$$\frac{d}{dt}\vec{v}^n = -2C_e^n \Omega_{ie}^e \dot{\vec{x}}^e - C_e^n \Omega_{ie}^e \Omega_{ie}^e \vec{x}^e + C_e^n \vec{g}^e + C_e^n \vec{a}^e - \Omega_{en}^n \vec{v}^n$$

$$\Omega_{ie}^e = C_e^n \Omega_{ie}^n C_n^e$$

$$\boxed{\frac{d}{dt}\vec{v}^n = \vec{a}^n - \left(2\Omega_{ie}^n + \Omega_{en}^n\right)\vec{v}^n + \vec{g}^n - C_e^n \Omega_{ie}^e \Omega_{ie}^e \vec{x}^e}$$

Navigation equations in the *n-frame*

$$\boxed{\frac{d}{dt}\vec{v}^n = \vec{a}^n - (2\Omega^n_{ie} + \Omega^n_{en})\vec{v}^n + \vec{g}^n - C^n_e \Omega^e_{ie}\Omega^e_{ie}\vec{x}^e}$$

gravitational vector

centrifugal acceleration

$$\vec{g}^n_{total} = \vec{g}^n - C^n_e \Omega^e_{ie}\Omega^e_{ie}\vec{x}^e$$

$$2\Omega^n_{ie} + \Omega^n_{en} = \Omega^n_{in} + \Omega^n_{ie}$$

$$\frac{d}{dt}\vec{v}^n = \vec{a}^n - (\Omega^n_{in} + \Omega^n_{ie})\vec{v}^n + \vec{g}^n_{total}$$

Gravity vector

Where coordinates are obtained in the *e-frame* by using $\frac{d}{dt}\vec{x}^e = C^e_n \vec{v}^n$

Navigation equations in the *n-frame*

Usually the equation formulated in terms of the geodetic latitude, *longitude, and height* (φ, λ, h) instead of the ECEF coordinates

$$\begin{pmatrix} x_1^e \\ x_2^e \\ x_3^e \end{pmatrix} = \begin{pmatrix} (\nu + h)\cos\varphi\cos\lambda \\ (\nu + h)\cos\varphi \operatorname{sen}\lambda \\ [\nu(1-e^2) + h]\operatorname{sen}\varphi \end{pmatrix} \qquad \frac{d}{dt}\begin{pmatrix} x_1^e \\ x_2^e \\ x_3^e \end{pmatrix} = \begin{pmatrix} \frac{\partial x_1^e}{\partial\varphi}\dot{\varphi} + \frac{\partial x_1^e}{\partial\lambda}\dot{\lambda} + \frac{\partial x_1^e}{\partial h}\dot{h} \\ \frac{\partial x_2^e}{\partial\varphi}\dot{\varphi} + \frac{\partial x_2^e}{\partial\lambda}\dot{\lambda} + \frac{\partial x_2^e}{\partial h}\dot{h} \\ \frac{\partial x_3^e}{\partial\varphi}\dot{\varphi} + \frac{\partial x_3^e}{\partial\lambda}\dot{\lambda} + \frac{\partial x_3^e}{\partial h}\dot{h} \end{pmatrix}$$

$$\begin{aligned} \frac{d}{d\varphi}\left[(\nu + h)\cos\varphi\right] &= -(\rho + h)\operatorname{sen}\varphi \\ \frac{d}{d\varphi}\left[\left(\nu(1-e^2) + h\right)\right] &= (\rho + h)\cos\varphi \end{aligned}$$

$$\dot{\vec{x}}^e = \begin{pmatrix} -\dot{\varphi}(\rho + h)\operatorname{sen}\varphi\cos\lambda - \dot{\lambda}(\nu + h)\cos\varphi\operatorname{sen}\lambda + \dot{h}\cos\varphi\cos\lambda \\ -\dot{\varphi}(\rho + h)\operatorname{sen}\varphi\sin\lambda - \dot{\lambda}(\nu + h)\cos\varphi\cos\lambda + \dot{h}\cos\varphi\operatorname{sen}\lambda \\ \dot{\varphi}(\rho + h)\cos\varphi + \dot{h}\operatorname{sen}\varphi \end{pmatrix}$$

Navigation equations in the *n-frame*

$$\dot{\vec{x}}^e = \begin{pmatrix} -\dot{\varphi}(\rho+h)\,\mathrm{sen}\,\varphi\cos\lambda - \dot{\lambda}(\nu+h)\cos\varphi\,\mathrm{sen}\,\lambda + \dot{h}\cos\varphi\cos\lambda \\ -\dot{\varphi}(\rho+h)\,\mathrm{sen}\,\varphi\sin\lambda - \dot{\lambda}(\nu+h)\cos\varphi\cos\lambda + \dot{h}\cos\varphi\,\mathrm{sen}\,\lambda \\ \dot{\varphi}(\rho+h)\cos\varphi + \dot{h}\,\mathrm{sen}\,\varphi \end{pmatrix}$$

$$C_e^n = \begin{pmatrix} -\mathrm{sen}\,\varphi\cos\lambda & -\mathrm{sen}\,\varphi\,\mathrm{sen}\,\lambda & \cos\varphi \\ -\mathrm{sen}\,\lambda & \cos\lambda & 0 \\ -\cos\varphi\cos\lambda & -\cos\varphi\,\mathrm{sen}\,\lambda & -\sin\varphi \end{pmatrix} \qquad \vec{v}^n = C_e^n \dot{\vec{x}}^e$$

$$\vec{v}^n = \begin{pmatrix} \dot{\varphi}(\rho+h) \\ \dot{\lambda}(\nu+h)\cos\varphi \\ -\dot{h} \end{pmatrix}$$

Where is obvious that $\vec{v}^n \neq \dot{\vec{x}}^n$

Navigation equations in the *n-frame*

Finally, the angular rates in are also expressed readily in terms of latitude and longitude rates

$$\Omega_{in}^{n} = \begin{pmatrix} 0 & (\dot{\lambda} + \omega_e)\,\mathrm{sen}\,\varphi & -\dot{\varphi} \\ -(\dot{\lambda} + \omega_e)\,\mathrm{sen}\,\varphi & 0 & -(\dot{\lambda} + \omega_e)\cos\varphi \\ \dot{\varphi} & (\dot{\lambda} + \omega_e)\cos\varphi & 0 \end{pmatrix}$$

$$\Omega_{ie}^{n} = C_n^e \Omega_{ie}^{e} C_e^n = \begin{pmatrix} 0 & \omega_e\,\mathrm{sen}\,\varphi & 0 \\ -\omega_e\,\mathrm{sen}\,\varphi & 0 & -\omega_e\cos\varphi \\ 0 & \omega_e\cos\varphi & 0 \end{pmatrix}$$

$$\Omega_{in}^{n} + \Omega_{ie}^{n} = \begin{pmatrix} 0 & (\dot{\lambda} + 2\omega_e)\,\mathrm{sen}\,\varphi & -\dot{\varphi} \\ -(\dot{\lambda} + 2\omega_e)\,\mathrm{sen}\,\varphi & 0 & -(\dot{\lambda} + 2\omega_e)\cos\varphi \\ \dot{\varphi} & (\dot{\lambda} + 2\omega_e)\cos\varphi & 0 \end{pmatrix}$$

Navigation equations in the *n-frame*

Now let the components of the Earth-referenced velocity, the sensed acceleration and the gravity vector be denoted more descriptively by their north, east, and down components

$$\vec{a}^n = \begin{pmatrix} a_N \\ a_E \\ a_D \end{pmatrix} \quad \vec{v}^n = \begin{pmatrix} v_N \\ v_E \\ v_D \end{pmatrix} \quad \vec{g}^n = \begin{pmatrix} g_N \\ g_E \\ g_D \end{pmatrix}$$

Thus obtaining a set of six non-linear differential equations in the variables $(v_N, v_E, v_D, \varphi, \lambda, h)$

$$\frac{d}{dt}\begin{pmatrix} v_N \\ v_E \\ v_D \end{pmatrix} = \begin{pmatrix} a_N + g_N - 2\omega_e v_E \operatorname{sen}\varphi + \dot{\varphi} v_D - \dot{\lambda} \operatorname{sen}\varphi v_E \\ a_E + g_E - 2\omega_e \operatorname{sen}\varphi v_N + 2\omega_e \cos\varphi v_D + \dot{\lambda} \operatorname{sen}\varphi v_N + \dot{\lambda} \cos\varphi v_D \\ a_D + g_D - 2\omega_e \cos\varphi v_E - \dot{\lambda} \cos\varphi v_E - \dot{\varphi} v_N \end{pmatrix}$$

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\lambda} \\ \dot{h} \end{pmatrix} = \begin{pmatrix} \frac{v_N}{(\rho+h)} \\ \frac{v_E}{(\nu+h)} \cos\varphi \\ -v_D \end{pmatrix}$$

Numerical integration of differential equations (revision)

Jekeli 2.4 pp 44

Inertial navigation systems (INS) yield the position vector $\vec{x}$ *and the velocity* vector $\dot{\vec{x}}$ by integrating measured accelerations $\vec{a}$

Such an equation is known as a *differential equation* → expresses a relationship between the derivative(s) of a function and possibly the independent variable(s) and the function itself

$\dot{\vec{y}}(t) = \vec{f}(t, \vec{y}(t))$ with known initial conditions $\vec{y}(t_0) = \vec{\mu}$

Since integration methods can be applied to both vectorial equation

systems $\dot{\vec{y}}(t)$ or scalar functions like $\dot{y}(t) = f(t, y(t))$

the algorithms are going to be described using the latter in order to simplify the notation

Runge-Kutta methods

Although there are many methods for solving differential equations such as

$$\dot{\vec{y}}(t) = \vec{f}(t, \vec{y}(t))$$

we will use only the Runge-Kutta methods.

It is a single-step method that is very suitable to the case of navigation equations with known initial values.

The general idea behind the Runge-Kutta numerical techniques is based on:

- Taylor series expansion of the solution
- To approximate the solution by a polynomial

Runge-Kutta methods

Taylor series expansion of the solution in the t_0 neighbourhood

$$y(t) = y(t_0) + \dot{y}(t_0)(t - t_0) + \frac{1}{2!}\ddot{y}(t_0)(t - t_0)^2 + \cdots + \frac{1}{m!}y^m(t_0)(t - t_0)^m + \cdots$$

with
$$\left\{\begin{array}{rcl} \dot{y}(t_0) & = & f \\ \ddot{y}(t_0) & = & f_t + f_y f \\ \dddot{y}(t_0) & = & f_{tt} + 2f_{ty}f + f_{yy}f^2 + f_t f_y + f_y^2 f \end{array}\right.$$

where $$f_{ty} = \left.\frac{\partial^2 f}{\partial t \partial y}\right|_{t=t_0, y=y_0}$$

Since the initial values $y(t_0) = y_0$ is known, the first derivative $f(t_0, y_0)$ and the subsequent ones can be evaluated at t_0

Runge-Kutta methods

To circumvent the usually cumbersome, if not difficult, determination of higher derivatives of $f(t, y_t)$, the Runge-Kutta methods seek to approximate the solution by a polynomial in $(t - t_0)$

$$y(t) = y(t_0) + \dot{y}(t_0)(t - t_0) + \frac{1}{2!}\ddot{y}(t_0)(t - t_0)^2 + \cdots + \frac{1}{m!}y^m(t_0)(t - t_0)^m + \cdots$$

that agrees with the Taylor series up to some specified degree.

Instead of predicting the value of the solution at t, the prediction is based on a suitably weighted average of first derivatives at several points in the neighborhood of t_0

The interval h between successive steps in t need not be constant, although in practice it is often so organized

$$h = t_{n+1} - t_n$$

Runge-Kutta methods

The Runge-Kutta method can be particularised for different orders

First-order

$$y_{n+1} = y_n + hf(t_n, y_n)$$

Second-order

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2) \\ k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + h, y_n + hk_1) \end{aligned}$$

Runge-Kutta methods

Third-order

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{6}(k_1 + 4k_2 + k_3) \\ k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 &= f(t_n + h, y_n - hk_1 + 2hk_2) \end{aligned}$$

Fourth-order

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 &= f(t_n, y_n) \\ k_2 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 &= f(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 &= f(t_n + h, y_n + hk_3) \end{aligned}$$

Numerical integration of functions

The method can be straightforwardly specialised to numerical integration of functions such as $\dot{y}(t) = f(t)$ where f does not depend on $y(t)$

First-order (rectangle rule) $y_{n+1} = y_n + hf(t_n)$

Second-order (trapezoid rule) $y_{n+1} = y_n + \frac{h}{2}\left(f(t_n) + f(t_n + h)\right)$

Third/four orden (Simpson's rule) $y_{n+1} = y_n + \frac{h}{6}\left(f(t_n) + f(t_n + \frac{h}{2}) + f(t_n + h)\right)$

In each case the error of the algorithm is on the order of the neglected power of the integration step h

Numerical integration of navigation equations

J. 4.3.6 The navigation equations in an arbitrary frame are given by

$$\begin{aligned}\frac{d}{dt}\vec{v}^a &= C_b^a\vec{a}^b + \vec{g}^a - 2\Omega_{ia}^a\vec{v}^a - (\dot{\Omega}_{ia}^a + \Omega_{ia}^a\Omega_{ia}^a)\vec{x}^a \\ \frac{d}{dt}\vec{x}^a &= \dot{\vec{v}}^a\end{aligned}$$

The three first equations can be written in a more compact way

$$\frac{d}{dt}\vec{v}^a = C_b^a\vec{a}^b + \vec{f}\left(\vec{x}^a, \vec{v}^a, \Omega_{ia}^a, \dot{\Omega}_{ia}^a, \vec{g}^a\right)$$

Initial data $\left\{\begin{array}{l}\vec{a}^b \\ \vec{\omega}_{ib}^b \Longrightarrow C_b^a \;\; \Omega_{ia}^a \;\; \dot{\Omega}_{ia}^a\end{array}\right.$

By intergrating $\vec{a}^b \Longrightarrow \vec{v}^a \Longrightarrow \vec{x}^a$

Numerical integration of navigation equations

Generally, the terms to be integrated vary differently

$$\Delta \vec{v}^a = \int_{\Delta t} C_b^a(t')\vec{a}^b(t')dt' + \vec{f}\left(\vec{x}^a, \vec{v}^a, \Omega_{ia}^a, \dot{\Omega}_{ia}^a, \vec{g}^a\right)\Delta t$$

$C_b^a \vec{a}^b \Rightarrow$ needs high-order algorithm (δt)

$\vec{f}\left(\vec{x}^a, \vec{v}^a, \Omega_{ia}^a, \dot{\Omega}_{ia}^a, \vec{g}^a\right) \Rightarrow$ first-order algorithm (Δt)

We assume that $\Delta t = 2\delta t$ $\rightarrow$ third order (Simpson)

δt INS raw data sampling interval

Δt integration interval

$$\Delta t \left\{ \begin{array}{cc} t_{l-2} & \\ & \delta t \\ t_{l-1} & \\ & \delta t \\ t_l & \end{array} \right.$$

Numerical integration of navigation equations

Acceleration vector is approximated by a Taylor series valid in Δt

$$\vec{a}(t) = \vec{a}_{l-2} + \dot{\vec{a}}_{l-2}(t - t_{l-2}) + O(\Delta t^3)$$

$$\delta\vec{v}_l^b = \int_{t_{l-1}}^{t_l} \vec{a}^b(t')dt' \qquad \left\{ \begin{array}{l} \delta\vec{v}_{l-1}^b = \displaystyle\int_{t_{l-2}}^{t_{l-1}} \vec{a}^b(t')dt' = \vec{a}_{l-2}\delta t + \frac{1}{2}\dot{\vec{a}}_{l-2}\delta t^2 + O(\Delta t^3) \\[2em] \delta\vec{v}_l^b = \displaystyle\int_{t_{l-1}}^{t_l} \vec{a}^b(t')dt' = \vec{a}_{l-2}\delta t + \frac{3}{2}\dot{\vec{a}}_{l-2}\delta t^2 + O(\Delta t^3) \end{array} \right.$$

$$\begin{aligned} \vec{a}_{l-2}^b &= \frac{1}{2\delta t}(3\delta\vec{v}_{l-1}^b - \delta\vec{v}_l^b) \\ \dot{\vec{a}}_{l-2}^b &= \frac{1}{\delta t^2}(\delta\vec{v}_l^b - \delta\vec{v}_{l-1}^b) \end{aligned} \quad \Longrightarrow \quad \begin{aligned} \hat{\vec{a}}_{l-2}^b\Delta t &= 3\delta\vec{v}_{l-1}^b - \delta\vec{v}_l^b \\ \hat{\vec{a}}_{l-1}^b\Delta t &= \delta\vec{v}_{l-1}^b + \delta\vec{v}_l^b \\ \hat{\vec{a}}_l^b\Delta t &= 3\delta\vec{v}_l^b - \delta\vec{v}_{l-1}^b \end{aligned}$$

Numerical integration of navigation equations

Assuming that a first estimation of $\hat{C}_b^a$ is known, the first part is solved by means

$$\int_{\Delta t} C_b^a(t')\vec{a} \approx \frac{\Delta t}{6}\left(\hat{C}_b^a(l-2)\hat{\vec{a}}_{l-2}^b + 4\hat{C}_b^a(l-1)\hat{\vec{a}}_{l-1}^b + \hat{C}_b^a(l)\hat{\vec{a}}_l^b\right)$$

Therefore, using the acceleration estimates, the algorithm for the estimated velocity at epoch t_l is

$$\begin{aligned}\hat{\vec{v}}_l^a =& \hat{\vec{v}}_{l-2}^a + \frac{1}{6}[\hat{C}_b^a(l-2)(3\delta\vec{v}_{l-1}^b - \delta\vec{v}_l^b) + 4\hat{C}_b^a(l-1)(\delta\vec{v}_{l-1}^b + \delta\vec{v}_l^b)\\ &+ \hat{C}_b^a(l)(3\delta\vec{v}_l^b - \delta\vec{v}_{l-1}^b)] + \vec{f}\left(\vec{x}^a, \vec{v}^a, \Omega_{ia}^a, \dot{\Omega}_{ia}^a, \vec{g}^a\right)_{t=t_{l-2}}\Delta t\end{aligned}$$

With only one set of initial values in t_0 only odd multiples of δt can be estimated $\rightarrow$ comment

Numerical integration of navigation equations

It is shown (Jekeli 2001) that the used algorithm to obtain $\hat{\vec{v}}_l^a$ represents a fourth-order algorithm for the velocity, provided the transformation matrix $\hat{C}_b^a$ was obtained from a third-order algorithm.

Once $\hat{\vec{v}}_l^a$ is known, the position can be obtained (frames)

$$\hat{\vec{x}}_l^i = \hat{\vec{x}}_{l-2}^i + \hat{\dot{\vec{x}}}_{l-1}^i \Delta t \qquad\qquad \hat{\vec{x}}_l^e = \hat{\vec{x}}_{l-2}^e + \hat{\dot{\vec{x}}}_{l-1}^e \Delta t$$

n-frame

$$\left\{ \begin{array}{rcl} \hat{\varphi}_l & = & \hat{\varphi}_{l-2} + \dfrac{(\hat{v}_N)_{l-1}\Delta t}{(\hat{\rho}_{l-1} + \hat{h}_{l-1})} \\ \hat{\lambda}_l & = & \hat{\lambda}_{l-2} + \dfrac{(\hat{v}_E)_{l-1}\Delta t}{(\hat{\nu}_{l-1} + \hat{h}_{l-1})\cos\hat{\varphi}_{l-1}} \\ \hat{h}_l & = & \hat{h}_{l-2} - (\hat{v}_D)_{l-1}\Delta t \end{array} \right.$$

How to compute the transformation matrix

Not always is needed

In *strapdown* mechanisations we need to know $\hat{C}_b^a$ from $\vec{\omega}_{ib}^b$

$$\vec{\omega}_{ab}^b = \vec{\omega}_{ai}^b + \vec{\omega}_{ib}^b \quad \left\{ \begin{array}{l} \vec{\omega}_{ai}^b = -\vec{\omega}_{ia}^b \\ \\ \vec{\omega}_{ia}^a \quad \text{known} \end{array} \right\} \quad \vec{\omega}_{ab}^b = \vec{\omega}_{ib}^b - C_a^b \vec{\omega}_{ia}^a$$

Taylor series…

$$C(t) = C_0 + \dot{C}_0 \Delta t + \frac{1}{2!}\ddot{C}_0^2 \Delta t^2 + \frac{1}{3!}\dddot{C}_0^3 \Delta t^3 + \cdots$$

with

$C(t)$ Transformation matrix C_a^i for epoch t
C_0 Transforation matrix C_a^i for epoch t_0

How to compute the transformation matrix

Being the time derivatives

$$\begin{aligned}\dot{C} &= C\Omega \\ \ddot{C} &= C(\Omega^2 + \dot{\Omega}) \\ \dddot{C} &= C(\Omega^3 + \dot{\Omega}\Omega + 2\Omega\dot{\Omega} + \ddot{\Omega})\end{aligned}$$

$$C(t) = C_0\left(I + \Omega_0\Delta t + \frac{1}{2!}(\dot{\Omega}_0\Omega_0^2)\Delta t + \frac{1}{3!}(\Omega_0^3 + \dot{\Omega}_0\Omega_0 + 2\Omega_0\dot{\Omega}_0 + \ddot{\Omega}_0)\Delta t^3 + \cdots\right)$$

where

$C(t)$	Transformation matrix C_a^i for epoch t
C_0	Transforation matrix C_a^i for epoch t_0
Ω_0	Angular rate matrix $\Omega_a^i = [\vec{\omega}_{ib}^b \times]$ for epoch t_0

How to compute the transformation matrix

Alternative solution: quaternions + Runge-Kutta third-order

(less computational load)

$$\hat{\vec{q}}_l = \hat{\vec{q}}_{l-2} + \frac{\delta t}{6}\left(\Delta\vec{q}_0 + 4\Delta\vec{q}_1 + \Delta\vec{q}_2\right)$$

with

$$\begin{aligned}
\Delta\vec{q}_0 &= f(t_{l-2}, \hat{\vec{q}}_{l-2}) \\
\Delta\vec{q}_1 &= f(t_{l-2}, \hat{\vec{q}}_{l-2} + \frac{\delta t}{2}\Delta\vec{q}_0) \\
\Delta\vec{q}_2 &= f(t_{l-2}, \hat{\vec{q}}_{l-2} + \delta t\Delta\vec{q}_0 + 2\delta t\Delta\vec{q}_1)
\end{aligned}$$