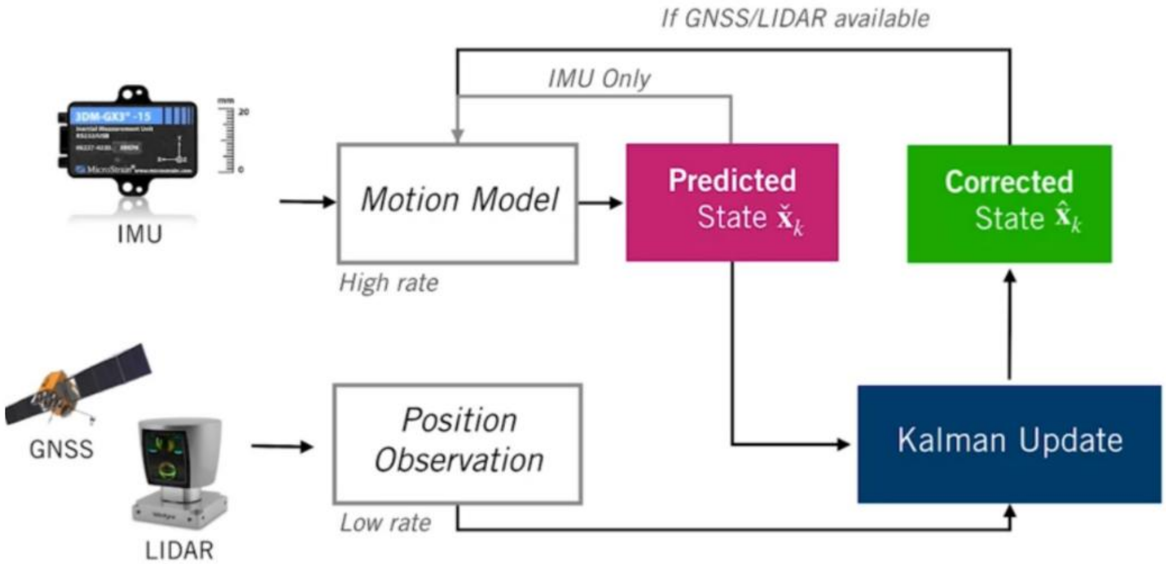

I. Contruct Motion model from IMU output

State vector definition:

- Position vector:

$$\boldsymbol{p}^{geo} = [\varphi \quad \lambda \quad h]$$

With $\varphi, \lambda, and\ h$ are latitude, longitude, and altitude of the vehicle, respectively.

- Velocity vector:

$$\boldsymbol{v}^n = [v_e \quad v_n \quad v_u]^T$$

With $v_e, v_n, and\ v_u$ are the East velocity, North velocity, and Up velocity, respectively.
- Quaternion is chosen for describing the vehicle's attitude

$$\boldsymbol{Q} = [q_0 \quad q_1 \quad q_2 \quad q_3]^T$$

The we have the rotational matrix $\boldsymbol{C}_b^n$ from body frame to navigation frame (ENU)

$$\boldsymbol{C}_b^n = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$$

1. **Position prediction model**

   When vehicle moves, the first order derivative motion model is described as

$$\dot{\varphi} = \frac{v_n}{R_M + h}$$
$$\dot{\lambda} = \frac{v_e}{(R_N + h)\cos\ \varphi}$$
$$\dot{h} = v_u$$

Where R_M and R_N are the meridian radius and normal radius, respectively. Then

$$\dot{\boldsymbol{p}}^{geo} = \boldsymbol{C}_n^{geo} \boldsymbol{v}^n$$

with

$$\boldsymbol{C}_n^{geo} = \begin{pmatrix} \frac{1}{R_M + h} & 0 & 0 \\ 0 & \frac{1}{(R_N + h)\ cos\,\varphi} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

2. **Velocity prediction model**

Output force from IMU in body frame is converted to navigation frame

$$\boldsymbol{f}^n = \boldsymbol{C}_b^n \boldsymbol{f}^b$$

with $\boldsymbol{f}^b = [f_x \quad f_y \quad f_z]^T$ and $\boldsymbol{f}^n = [f_e \quad f_n \quad f_u]^T$.

Velocity prediction model

$$\dot{\boldsymbol{v}} = \boldsymbol{f}^n - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \boldsymbol{v} + \boldsymbol{g}$$

with $\boldsymbol{\omega}_{ie}^n = [0 \quad \omega_{ie} \quad \cos\ \varphi \quad \omega_{ie}\sin\ \varphi]^T$ is the Earth rotation rate in navigation frame, $\omega_{en}^n = \left[-\frac{v_n}{R_M+h} \frac{v_e}{R_N+h} \frac{v_e \tan\ \varphi}{R_N+h}\right]^T$ is the angular velocity caused by the change of orientation of the navigation frame with respect to the Earth, and $\boldsymbol{g} = [0 \quad 0 \quad -g]^T$ is gravity force vector.

To simplify the calculation, $\boldsymbol{\omega}_{ie}^n$ and ω_{en}^n are ignored (test). Due to the high noise level of the MEMS grade inertial sensors, the earth rotation rate cannot be detected. Furthermore, when the vehicle moves in low speed, the transportation rate is negligible. Then, the velocity prediction model is rewritten as

$$\dot{\boldsymbol{v}} = \boldsymbol{f}^n + \boldsymbol{g} = \boldsymbol{C}_b^n \boldsymbol{f}^b + \boldsymbol{g}$$

3. **Attitude Prediction Model**

$$\dot{\boldsymbol{Q}} = \frac{1}{2}\begin{bmatrix} 0 & -(\boldsymbol{\omega}_{nb}^b)^T \\ \boldsymbol{\omega}_{nb}^b & -(\boldsymbol{\omega}_{nb}^b)\times \end{bmatrix} \boldsymbol{Q} = \frac{1}{2}\ \Omega\ \boldsymbol{Q}$$

where $\boldsymbol{\omega}_{nb}^{b}$ is the angular velocity of the body frame with respect to the navigation frame represented in the body frame, and

$$\boldsymbol{\omega}_{nb}^{b} = \boldsymbol{\omega}_{ib}^{b} - \boldsymbol{C}_{n}^{b}(\boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n})$$

with $\boldsymbol{\omega}_{ib}^{b} = [\omega_x \quad \omega_y \quad \omega_z]^T$ is the gyroscope measurement in the body frame, representing the angular velocity of the body frame with respect to the inertial frame.

When the earth rotation rate and the transportation rate are negligible, we obtain

$$\boldsymbol{\omega}_{nb}^{b} \approx \boldsymbol{\omega}_{ib}^{b}$$

Then

$$\begin{aligned}
\dot{q}_0 &= 0.5\left(-q_1\omega_x - q_2\omega_y - q_3\omega_z\right)\\
\dot{q}_1 &= 0.5\left(q_0\omega_x - q_3\omega_y + q_2\omega_z\right)\\
\dot{q}_2 &= 0.5\left(q_3\omega_x + q_0\omega_y - q_1\omega_z\right)\\
\dot{q}_3 &= 0.5\left(-q_3\omega_x + q_1\omega_y + q_0\omega_z\right)
\end{aligned}$$

⇨ $\dot{\boldsymbol{Q}} = \frac{1}{2}\boldsymbol{\omega}_{nb}^{b} \otimes \boldsymbol{Q}$

And the equation which converts the quaternion to Euler angle is

$p = \sin^{-1}\left(\boldsymbol{C}_b^n(3,2)\right)$

$r = \tan^{-1}\left(\frac{-\boldsymbol{C}_b^n(3,1)}{\boldsymbol{C}_b^n(3,3)}\right)$

$A = \tan^{-1}\left(\frac{-\boldsymbol{C}_b^n(1,2)}{\boldsymbol{C}_b^n(2,2)}\right)$

From section 1, 2, and 3, the motion prediction model is obtained

$$\begin{cases}\dot{\boldsymbol{p}}^{geo} = \boldsymbol{C}_n^{geo}\boldsymbol{v}\\ \dot{\boldsymbol{v}} = \boldsymbol{C}_b^n\boldsymbol{f}^b + \boldsymbol{g}\\ \dot{\boldsymbol{Q}} = \frac{1}{2}\boldsymbol{\omega}_{nb}^{b} \otimes \boldsymbol{Q}\end{cases} \text{ or } \begin{cases}\dot{\boldsymbol{p}}^{n} = \boldsymbol{v}\\ \dot{\boldsymbol{v}} = \boldsymbol{C}_b^n\boldsymbol{f}^b + \boldsymbol{g}\\ \dot{\boldsymbol{Q}} = \frac{1}{2}\boldsymbol{\Omega}\boldsymbol{Q}\end{cases}$$

Calculate the discrete form of the model

$$\begin{cases}\boldsymbol{p}_k^n = \boldsymbol{p}_{k-1}^n + \boldsymbol{v}_{k-1}^n\Delta t + \frac{1}{2}(\boldsymbol{C}_b^n\boldsymbol{f}^b + \boldsymbol{g})\Delta t^2\\ \boldsymbol{v}_k^n = \boldsymbol{v}_{k-1}^n + (\boldsymbol{C}_b^n\boldsymbol{f}^b + \boldsymbol{g})\Delta t\\ \boldsymbol{Q}_k = \boldsymbol{\Omega}\boldsymbol{Q}_{k-1}\end{cases} \qquad \text{(Use Quaternion)}$$

Or

$$\begin{cases} \boldsymbol{p}_k^n = \boldsymbol{p}_{k-1}^n + \boldsymbol{v}_{k-1}^n \Delta t + \frac{1}{2}(\boldsymbol{C}_b^n \boldsymbol{f}^b + \boldsymbol{g})\Delta t^2 \\ \boldsymbol{v}_k^n = \boldsymbol{v}_{k-1}^n + (\boldsymbol{C}_b^n \boldsymbol{f}^b + \boldsymbol{g})\Delta t \\ \boldsymbol{\Phi}_k = \boldsymbol{\Phi}_k + \boldsymbol{\omega}_{ib}^b \Delta t \end{cases}$$

(use Euler angle)

II. Error-state Kalman filter

A, Predict:

$$\check{\mathbf{x}}_k = \begin{bmatrix} \check{\mathbf{p}}_k \\ \check{\mathbf{v}}_k \\ \check{\mathbf{q}}_k \end{bmatrix} \in R^{10}$$

$$\begin{aligned} \check{\mathbf{p}}_k &= \mathbf{p}_{k-1} + \Delta t \mathbf{v}_{k-1} + \frac{\Delta t^2}{2}(\boldsymbol{C}_b^n \mathbf{f}_{k-1} + \mathbf{g}_n) \\ \check{\mathbf{v}}_k &= \mathbf{v}_{k-1} + \Delta t(\boldsymbol{C}_b^n \mathbf{f}_{k-1} + \mathbf{g}_n) \\ \check{\mathbf{q}}_k &= \Omega \mathbf{q}_{k-1} \end{aligned}$$

B, Error state model

$$\delta \mathbf{x}_k = \begin{bmatrix} \delta \mathbf{p}_k \\ \delta \mathbf{v}_k \\ \delta \boldsymbol{\phi}_k \end{bmatrix} \in R^9$$

Error linearized state model:

$$\delta \mathbf{x}_k = \mathbf{F}_{k-1} \delta \mathbf{x}_{k-1} + \mathbf{L}_{k-1} \mathbf{n}_{k-1}$$

$\mathbf{F}_{k-1} = \begin{bmatrix} 1 & 1\Delta t & 0 \\ 0 & 1 & -[\mathbf{C}_{ns} \mathbf{f}_{k-1}]_\times \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{L}_{k-1} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, and\ \mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ is measurement noise.

C, Uncertainty propagation

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1} \check{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{L}_{k-1} \mathbf{Q}_{k-1} \mathbf{L}_{k-1}^T$$

D, Check if GNSS data available

$$\begin{aligned} \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\sigma}_k \\ &= \mathbf{H}_k \mathbf{x}_k + \boldsymbol{\sigma}_k = [\mathbf{1}\ \ \mathbf{1}\ \ \mathbf{0}] \mathbf{x}_k + \boldsymbol{\sigma}_k = [\mathbf{p}_k\ \ \mathbf{v}_k] + \boldsymbol{\sigma}_k \end{aligned}$$ (assume GNSS generate position and velocity)

$$\boldsymbol{\sigma}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{\mathrm{GNSS}})$$

E, Calculate Kalman gain

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

F, Error State update

$$\delta \mathbf{x}_k = \mathbf{K}_k (\mathbf{y}_k - [\mathbf{p}_k \;\; \mathbf{v}_k])$$

G, Correct

$$\text{State} \quad \begin{aligned} \hat{\mathbf{p}}_k &= \check{\mathbf{p}}_k + \delta \mathbf{p}_k \\ \hat{\mathbf{v}}_k &= \check{\mathbf{v}}_k + \delta \mathbf{v}_k \\ \hat{\mathbf{q}}_k &= \mathbf{q}(\delta \boldsymbol{\phi}) \otimes \check{\mathbf{q}}_k \end{aligned}$$

$$\text{Uncertainty } \hat{\mathbf{P}}_k = (\mathbf{1} - \mathbf{K}_k \mathbf{H}_k) \check{\mathbf{P}}_k$$