Inertial Navigation Systems: INS

Kevin Urrutia Avila, Dr.Jay Farrell *†

April 19, 2023

1 Attitude Heading Reference System In the Inertial Frame

The following section is an guide for implementing an Attitude Heading Reference System that only uses attitude measurements to propagate and the integrated specific force measurement to determine the true orientation.

The IMU provides the attitude increments, $\widetilde{\boldsymbol{\alpha}}^b_{ib}$ in the following form at a defined rate:

$$\widetilde{\boldsymbol{\alpha}}^b_{ib} = \boldsymbol{\alpha}^b_{ib} + \boldsymbol{b}_{\boldsymbol{\omega}}\tau + \boldsymbol{\epsilon}_{\boldsymbol{\omega}}\tau \tag{1}$$

where $\boldsymbol{\alpha}^b_{ib}$ is the true value measured by the gyroscope, $\boldsymbol{b}_{\boldsymbol{\omega}}$ is a run time bias that can be estimated on the fly and finally $\boldsymbol{\epsilon}_{\boldsymbol{\omega}}$ is error modeled as white noise with power spectral density S_{rg}, the angular random walk determined from the IMU's data sheet after some manipulation.

The IMU provides this measurement as a numeric integration of the angular rate which is expressed as:

$$\widetilde{\boldsymbol{\omega}}^b_{ib} = \boldsymbol{\omega}^b_{ib} + \boldsymbol{b}_{\boldsymbol{\omega}} + \boldsymbol{\epsilon}_{\boldsymbol{\omega}} \tag{2}$$

1.1 Initialization

1.1.1 Leveling

To begin integrating the attitude the initial orientation in the form of a rotation matrix must be known, $\boldsymbol{C}^i_{b_o}$. Through the process of leveling the initial roll, ϕ_{ib}, and pitch, θ_{tb}, used to construct this initial rotation matrix.

$$\begin{bmatrix} \nu^b_{ib,x} \\ \nu^b_{ib,y} \\ \nu^b_{ib,z} \end{bmatrix} = \begin{bmatrix} -sin(\theta_{ib}) \\ cos(\theta_{ib})sin(\phi_{ib}) \\ cos(\theta_{ib})cos(\phi_{ib}) \end{bmatrix} \boldsymbol{g}^t, \tag{3}$$

where $\boldsymbol{g}^t$ is the down component of acceleration due to gravity. The data can not give information about the yaw, ψ_{ib}, and needs to be determined by other means. Since this experiment is known the initial yaw is determined to be 0 rad, roll and pitch are determined using:

$$\theta_{ib} \doteq arctan\left(\frac{\nu^b_{ib,x}}{\sqrt{\nu^{b2}_{ib,y} + \nu^{b2}_{ib,z}}}\right), \tag{4}$$

$$\phi_{ib} \doteq arctan_2(-\nu^b_{ib,y}, -\nu^b_{ib,z}). \tag{5}$$

The roll and pitch are computed over a one second interval where IMU is stationary. The roll and pitch are averaged to get the estimate for these values.

*Kevin Urrutia Avila is with the Dept. of Electrical and Computer Engineering, University of California, Riverside. Email: {kurru006}@ucr.edu.
†We gratefully acknowledge the support and tutelage of the following entities: Dr. Jay Farrell.

1.1.2 Initial Rotation Matrix

Using the initial Euler angles constructed in Section 1.1.1 the initial rotation matrix, $\boldsymbol{C}^i_{b_o}$, is constructed. The first step is to determine the rotation matrix, $\boldsymbol{C}^b_{i_o}$, dropping the subscripts for tangent to body for a moment:

$$\boldsymbol{C}^b_{i_o} \doteq \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ -c_\phi s_\psi + s_\phi s_\theta c_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & s_\phi c_\theta \\ s_\phi s_\psi + c_\phi s_\theta c_\psi & -s_\phi c_\psi + c_\phi s_\theta s_\psi & c_\phi c_\theta \end{bmatrix}. \tag{6}$$

This rotation matrix is used to determine the true orientation that will be propagated in the AHRS.

$$\boldsymbol{C}^{ti}_{b_o} \doteq \boldsymbol{C}^{b^T}_{i_o}. \tag{7}$$

The reverse conversion is:

$$\phi_{ib} \doteq arctan_2(\boldsymbol{C}^i_{b,3,2}, \boldsymbol{C}^i_{b,3,3}), \tag{8}$$

$$\theta_{ib} \doteq -atan\left(\frac{\boldsymbol{C}^i_{b,3,1}}{\sqrt{1-(\boldsymbol{C}^i_{b,3,1})^2}}\right), \tag{9}$$

$$\psi_{ib} \doteq arctan_2(\boldsymbol{C}^i_{b,2,1}, \boldsymbol{C}^i_{b,1,1}). \tag{10}$$

1.1.3 Bias Initialization

During a one second interval where the IMU is stationary averaging the attitude increments gives an initial estimate for the bias.

$$\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}} \doteq \frac{1}{N}\sum_{k=1}^{N}(\widetilde{\boldsymbol{\alpha}}^b_{ib})_k. \tag{11}$$

1.2 Unaided Attitude Propagation

To begin the integration first the calculated bias needs to be removed from the attitude increment measurement:

$$\widehat{\boldsymbol{\alpha}}^b_{ib} \doteq \widetilde{\boldsymbol{\alpha}}^b_{ib} - \widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}\,\tau \tag{12}$$

The attitude update in the local tangent frame uses the newly calculated $\widehat{\boldsymbol{\alpha}}^b_{ib}$ in the form of the skew symmetric matrix, $\boldsymbol{\Omega}^b_{ib}$:

$$\boldsymbol{\Omega}^b_{ib} \doteq [\widehat{\boldsymbol{\alpha}}^b_{ib}\times] \tag{13}$$

Using the $\boldsymbol{\Omega}^b_{ib}$ the differential equation for $\boldsymbol{C}^i_b$ is defined as:

$$\dot{\boldsymbol{C}}^i_b = \boldsymbol{\Omega}^b_{ib}\,\boldsymbol{C}^i_b \tag{14}$$

in [1] section 2.7.1. Integrating this differential equation to propagate the the attitude for a small time interval τ yields:

$$\boldsymbol{C}^i_b(t+\tau) \approxeq \boldsymbol{C}^i_b(t)\; exp(\boldsymbol{\Omega}^b_{ib}) \tag{15}$$

Applying a small angle approximation by truncating to a first order expansion gives:

$$\widehat{\boldsymbol{C}}^i_b(-) \doteq \widehat{\boldsymbol{C}}^i_b(-)(\boldsymbol{I}_3 + \boldsymbol{\Omega}^b_{ib}) \tag{16}$$

as defined by [2] section 5.2.1. Therefore the the following state is propagated through the unaided attitude increment:

$$\widehat{\boldsymbol{x}} = \begin{bmatrix} \phi_{ib} \\ \theta_{ib} \\ \psi_{ib} \\ \boldsymbol{b}_{\boldsymbol{\omega}} \end{bmatrix} \tag{17}$$

where the roll, pitch and yaw are extracted from the propagated rotation matrix, $\boldsymbol{C}^i_b(-)$ using eqns. 8 - 10.

1.3 AHRS State Space and Attitude Error Representation

1.3.1 Attitude Error Representation

The integration of the rotation matrix places the IMU in a frame corrupted by errors, $\widehat{\boldsymbol{C}}_b^t$. Therefore the rotation matrix requires a multiplication of an intermediate error rotation matrix to correct itself, $\delta \boldsymbol{C}_b^t$ to the true orientation:

$$\boldsymbol{C}_{b_o}^i = \delta \boldsymbol{C}_b^i \, \widehat{\boldsymbol{C}}_b^i(-) \tag{18}$$

$\delta C_b^i = C_{b_o}^i \, \hat{C}_b^{i\,T}$

$= C_{b_o}^i \, \hat{C}_i^b$

1.3.2 State Space Model and State Transition Matrix

Using Section 1.3.1 the error state is defined as:

$$\delta \boldsymbol{x} = \begin{bmatrix} \boldsymbol{\rho} \\ \delta \boldsymbol{b}_{\boldsymbol{\omega}} \end{bmatrix} \tag{19}$$

where $\boldsymbol{\rho}$ is the small angle attitude error and $\delta \boldsymbol{b}_{\boldsymbol{\omega}}$ is:

$$\delta \boldsymbol{b}_{\boldsymbol{\omega}} = \boldsymbol{b}_{\boldsymbol{\omega}} - \widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}. \tag{20}$$

Making the continuous time state space model:

$$\delta \dot{\boldsymbol{x}} = \boldsymbol{F}\, \delta \boldsymbol{x} + \boldsymbol{G}\, \boldsymbol{u} \tag{21}$$

where $\boldsymbol{u} = [\boldsymbol{\epsilon}_{\boldsymbol{\alpha}}, \boldsymbol{\epsilon}_{\boldsymbol{b}}]^T$ is the input to the state space model. And $\boldsymbol{\epsilon}_{\boldsymbol{b}}$ is the bias error modeled as white noise with power spectral density S_{bg} determined from the IMU's data sheet.

The differential equation for $\boldsymbol{\rho}$ is derived by first establishing the following definitions:

$$\boldsymbol{C}_b^i = (\boldsymbol{I}_3 + \boldsymbol{P})\, \widehat{\boldsymbol{C}}_b^i = \widehat{\boldsymbol{C}}_b^i + \boldsymbol{P}\, \widehat{\boldsymbol{C}}_b^i \tag{22}$$

where $\boldsymbol{P} = [\boldsymbol{\rho}\times]$. And

$$\dot{\boldsymbol{C}}_b^i = \boldsymbol{C}_b^i\, \boldsymbol{\Omega}_{ib}^b = (\boldsymbol{I}_3 + \boldsymbol{P})\, \widehat{\boldsymbol{C}}_b^i\, \boldsymbol{\Omega}_{ib}^b \tag{23}$$

Taking the derivative of eqn. 22 yeilds:

$$\dot{\boldsymbol{C}}_b^i = \dot{\boldsymbol{P}}\, \widehat{\boldsymbol{C}}_b^i + (\boldsymbol{I}_3 + \boldsymbol{P})\, \dot{\widehat{\boldsymbol{C}}}_b^i \tag{24}$$

Setting eqns. 23 and 24 equal to another:

$$\dot{\boldsymbol{P}}\, \widehat{\boldsymbol{C}}_b^i + (\boldsymbol{I}_3 + \boldsymbol{P})\, \dot{\widehat{\boldsymbol{C}}}_b^i = (\boldsymbol{I}_3 + \boldsymbol{P})\, \widehat{\boldsymbol{C}}_b^i\, \boldsymbol{\Omega}_{ib}^b \tag{25}$$

Defining:

$$\dot{\widehat{\boldsymbol{C}}}_b^i = \widehat{\boldsymbol{C}}_b^i\, \widehat{\boldsymbol{\Omega}}_{ib}^b \tag{26}$$

and

$$\delta \boldsymbol{\Omega}_{ib}^b = \boldsymbol{\Omega}_{ib}^b - \widehat{\boldsymbol{\Omega}}_{ib}^b \tag{27}$$

Substituting back into eqn. 25:

$$\begin{aligned} \dot{\boldsymbol{P}}\, \widehat{\boldsymbol{C}}_b^i &= (\boldsymbol{I}_3 + \boldsymbol{P})\, \widehat{\boldsymbol{C}}_b^i\, (\boldsymbol{\Omega}_{ib}^b - \widehat{\boldsymbol{\Omega}}_{ib}^b) \\ \dot{\boldsymbol{P}} &= (\boldsymbol{I}_3 + \boldsymbol{P})\, \widehat{\boldsymbol{C}}_b^i\, \delta \boldsymbol{\Omega}_{ib}^b\, \widehat{\boldsymbol{C}}_i^b \\ \dot{\boldsymbol{P}} &= \widehat{\boldsymbol{C}}_b^i\, \delta \boldsymbol{\Omega}_{ib}^b\, \widehat{\boldsymbol{C}}_i^b + \boldsymbol{P}\, \widehat{\boldsymbol{C}}_b^i\, \delta \boldsymbol{\Omega}_{ib}^b\, \widehat{\boldsymbol{C}}_i^b \end{aligned} \tag{28}$$

After dropping the second order term:

$$\dot{\boldsymbol{P}} = \widehat{\boldsymbol{C}}_b^i\, \delta \boldsymbol{\Omega}_{ib}^b\, \widehat{\boldsymbol{C}}_i^b \tag{29}$$

Therefore the differential equations for each of the error states are defined as:

$$
\begin{aligned}
\boldsymbol{f}_1 = \dot{\boldsymbol{\rho}} &= \widehat{\boldsymbol{C}}_b^i\,(\delta\boldsymbol{\omega}_{ib}^b) \\
&= \widehat{\boldsymbol{C}}_b^i\,(\boldsymbol{\omega}_{ib}^b - \widehat{\boldsymbol{\omega}}_{ib}^b) \\
&= \boldsymbol{C}_b^i(\boldsymbol{\omega}_{ib}^b - (\widetilde{\boldsymbol{\omega}}_{ib}^b - \widehat{\boldsymbol{b}}_{\boldsymbol{\omega}})) \\
&= \boldsymbol{C}_b^i(\boldsymbol{\omega}_{ib}^b - (\boldsymbol{\omega}_{ib}^b + \boldsymbol{b}_{\boldsymbol{\omega}} + \boldsymbol{\epsilon}_{\boldsymbol{\omega}}) + \widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}) \\
&= \boldsymbol{C}_b^i\,(-\delta\,\boldsymbol{b}_{\boldsymbol{\omega}} - \boldsymbol{\epsilon}_{\boldsymbol{\omega}})
\end{aligned} \tag{30}
$$

$$
\boldsymbol{f}_2 = \delta\dot{\boldsymbol{b}}_{\boldsymbol{\omega}} = \boldsymbol{\epsilon}_{\boldsymbol{b}} \tag{31}
$$

The continuous state transition model is computed using:

$$
\boldsymbol{F}(t) = \partial f(\boldsymbol{x}, \boldsymbol{u}/\partial \boldsymbol{x})\Big|_{\widehat{\boldsymbol{x}}(t),\widehat{\boldsymbol{u}}(t)} \tag{32}
$$

Derivation State Transition Matrix	
$\frac{\partial \boldsymbol{f_1}}{\partial \boldsymbol{\rho}} = \boldsymbol{0}_3$	$\frac{\partial \boldsymbol{f_1}}{\partial \boldsymbol{b_\alpha}} = -\widehat{\boldsymbol{C}}_b^i$
$\frac{\partial \boldsymbol{f_2}}{\partial \boldsymbol{\rho}} = \boldsymbol{0}_3$	$\frac{\partial \boldsymbol{f_2}}{\partial \boldsymbol{b_\alpha}} = \boldsymbol{0}_3$

In matrix form:

$$
\boldsymbol{F} = \begin{bmatrix} \boldsymbol{0}_3 & -\widehat{\boldsymbol{C}}_b^i \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} \tag{33}
$$

$\boldsymbol{F}$ is converted to discrete time using a first order Taylor series expansion where τ is a interval increment.

$$
\begin{aligned}
\boldsymbol{\Phi} &= \boldsymbol{I}_6 + \boldsymbol{F}\tau \\
&= \boldsymbol{I}_6 + \begin{bmatrix} \boldsymbol{0}_3 & -\widehat{\boldsymbol{C}}_b^i\,\tau \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_3 & -\widehat{\boldsymbol{C}}_b^i\,\tau \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix}
\end{aligned} \tag{34}
$$

$\boldsymbol{G}$ is similarly computed by using:

$$
\boldsymbol{G}(t) = \partial f(\boldsymbol{x}, \boldsymbol{u}/\partial \boldsymbol{x})\Big|_{\widehat{\boldsymbol{x}}(t),\widehat{\boldsymbol{u}}(t)} \tag{35}
$$

Derivation State Transition Matrix	
$\frac{\partial \boldsymbol{f_1}}{\partial \boldsymbol{\epsilon_\alpha}} = -\widehat{\boldsymbol{C}}_b^i$	$\frac{\partial \boldsymbol{f_1}}{\partial \boldsymbol{\epsilon_\alpha}} = \boldsymbol{0}_3$
$\frac{\partial \boldsymbol{f_2}}{\partial \boldsymbol{\epsilon_b}} = \boldsymbol{0}_3$	$\frac{\partial \boldsymbol{f_2}}{\partial \boldsymbol{\epsilon_b}} = \boldsymbol{I}_3$

in matrix form:

$$
\boldsymbol{G} = \begin{bmatrix} -\widehat{\boldsymbol{C}}_b^i & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \tag{36}
$$

1.4 Covariance Initialization and Propagation

1.4.1 Initialization

Using $\delta\boldsymbol{x}$ defined in Section 1.3.2 the initial covariance, $\boldsymbol{P}$ is defined as:

$$
\boldsymbol{P} = \begin{bmatrix} \boldsymbol{V_\rho} & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{V_\alpha} \end{bmatrix} \tag{37}
$$

where $\boldsymbol{V_\rho}$ is sufficiently large since there is no known information about the state, $\boldsymbol{\rho}$. The uncertainty is initialized as, $\boldsymbol{\sigma}_{\boldsymbol{\rho}}^2 = \pi$ rad:

$$
\boldsymbol{V_\rho} = \boldsymbol{\sigma}_{\boldsymbol{\rho}}^2\,\boldsymbol{I}_3 \tag{38}
$$

and $\boldsymbol{V_\alpha}$ is defined as:

$$\boldsymbol{V_\alpha} = \begin{bmatrix} \sigma^2_{\alpha,x} & 0 & 0 \\ 0 & \sigma^2_{\alpha,y} & 0 \\ 0 & 0 & \sigma^2_{\alpha,z} \end{bmatrix} \tag{39}$$

where σ_α is calculated from $\boldsymbol{\alpha}^b_{tb}$ during a one second interval where the IMU is known to be stationary.

1.4.2 Measurement Noise

The system covariance is obtained by integrating the power spectral densities of system's inputs over the state propagation interval. Therefore assuming six states are estimated, the continuous time system covariance is defined as:

$$\boldsymbol{Q}_c = \begin{bmatrix} S_{rg}\,\boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & S_{bg}\,\boldsymbol{I}_3 \end{bmatrix} \tag{40}$$

The system covariance is discretized using:

$$\boldsymbol{Q}_d = \int_{t_k}^{t_k+1} exp(\boldsymbol{F}\,\tau)\,\boldsymbol{G}\,\boldsymbol{Q}_c\,\boldsymbol{G}^T\,exp(\boldsymbol{F}^T\,\tau)\,d\tau \tag{41}$$

Calculating an intermediate matrix $\boldsymbol{Q}$:

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{G}\,\boldsymbol{Q}_c\,\boldsymbol{G}^T \\ &= \begin{bmatrix} -\widehat{\boldsymbol{C}}^i_b & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} S_{rg}\,\boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & S_{bg}\,\boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} -\widehat{\boldsymbol{C}}^b_i & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \\ &= \begin{bmatrix} -\widehat{\boldsymbol{C}}^i_b\,S_{rg}\,\boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & S_{bgd}\,\boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} -\widehat{\boldsymbol{C}}^b_i & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \\ &= \begin{bmatrix} \widehat{\boldsymbol{C}}^i_b\,S_{rg}\,\boldsymbol{I}_3\,\widehat{\boldsymbol{C}}^b_i & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & S_{bgd}\,\boldsymbol{I}_3 \end{bmatrix} \end{aligned} \tag{42}$$

The intermediate matrix is calculated to then be substituted into eqn. 41.

$$\begin{gathered} exp(\boldsymbol{F}\,\tau)\,\boldsymbol{G}\,\boldsymbol{Q}_c\,\boldsymbol{G}^T\,exp(\boldsymbol{F}^T\,\tau) \\ \begin{bmatrix} \boldsymbol{I}_3 & -\widehat{\boldsymbol{C}}^i_b\,\tau \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{C}}^i_b\,S_{rg}\,\boldsymbol{I}_3\,\widehat{\boldsymbol{C}}^b_i & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & S_{bgd}\,\boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ -\widehat{\boldsymbol{C}}^b_i\tau & \boldsymbol{I}_3 \end{bmatrix} \\ \begin{bmatrix} \widehat{\boldsymbol{C}}^i_b\,S_{rg}\,\boldsymbol{I}_3\,\widehat{\boldsymbol{C}}^b_i & -\widehat{\boldsymbol{C}}^i_b\,S_{bg}\,\boldsymbol{I}_3\,\tau \\ \boldsymbol{0}_3 & S_{bg}\,\boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ -\widehat{\boldsymbol{C}}^b_i\,\tau & \boldsymbol{I}_3 \end{bmatrix} \\ \begin{bmatrix} \widehat{\boldsymbol{C}}^i_b\,S_{rg}\,\boldsymbol{I}_3\,\widehat{\boldsymbol{C}}^b_i + \widehat{C}^i_b\,S_{bg}\,\boldsymbol{I}_3\,\widehat{\boldsymbol{C}}^b_i\,\tau^2 & -\widehat{\boldsymbol{C}}^i_b\,S_{bg}\,\boldsymbol{I}_3\,\tau \\ -\widehat{\boldsymbol{C}}^b_i\,S_{bg}\,\boldsymbol{I}_3\,\tau & S_{bg}\,\boldsymbol{I}_3 \end{bmatrix} \end{gathered} \tag{43}$$

Substituting eqn. 43 into eqn. 41 the integration is taken element wise:

$$\begin{aligned} \boldsymbol{Q}_d[1:3,1:3] &= \int_{t_k}^{t_k+1} \widehat{\boldsymbol{C}}^i_b\,S_{rg}\,\boldsymbol{I}_3\,\widehat{\boldsymbol{C}}^b_i + \widehat{C}^i_b\,S_{bg}\,\boldsymbol{I}_3\,\widehat{\boldsymbol{C}}^b_i\,\tau^2\,d\tau \\ &= \widehat{\boldsymbol{C}}^i_b\,S_{rg}\boldsymbol{I}_3\,\widehat{\boldsymbol{C}}^b_i\,\tau + \frac{1}{3}\,\widehat{C}^i_b\,S_{bg}\,\boldsymbol{I}_3\,\widehat{\boldsymbol{C}}^b_i\,\tau^3 \end{aligned} \tag{44}$$

$$\begin{aligned} \boldsymbol{Q}_d[1:3,4:6] &= \int_{t_k}^{t_k+1} -\widehat{\boldsymbol{C}}^i_b\,S_{bg}\,\boldsymbol{I}_3\,\tau\,d\tau \\ &= -\frac{1}{2}\,\widehat{\boldsymbol{C}}^i_b\,S_{bg}\,\boldsymbol{I}_3\,\tau^2 \end{aligned} \tag{45}$$

$$\begin{aligned}\boldsymbol{Q}_d[4:6,1:3] &= \int_{t_k}^{t_k+1} -\widehat{\boldsymbol{C}}_i^b\, S_{bg}\, \boldsymbol{I}_3\, \tau\, d\tau \\ &= -\frac{1}{2}\, \widehat{\boldsymbol{C}}_i^b\, S_{bg}\, \boldsymbol{I}_3\, \tau^2\end{aligned} \tag{46}$$

$$\begin{aligned}\boldsymbol{Q}_d[4:6,4:6] &= \int_{t_k}^{t_k+1} S_{bg}\, \boldsymbol{I}_3\, d\tau \\ &= S_{bg}\, \boldsymbol{I}_3\, \tau\end{aligned} \tag{47}$$

In matrix form:

$$\boldsymbol{Q}_d = \begin{bmatrix} \widehat{\boldsymbol{C}}_b^i\, S_{rg}\, \boldsymbol{I}_3\, \widehat{\boldsymbol{C}}_i^b\, \tau + \frac{1}{3}\, \widehat{\boldsymbol{C}}_b^i\, S_{bg}\, \boldsymbol{I}_3\, \widehat{\boldsymbol{C}}_i^b\, \tau^3 & -\frac{1}{2}\, \widehat{\boldsymbol{C}}_b^i\, S_{bg}\, \boldsymbol{I}_3\, \tau^2 \\ -\frac{1}{2}\, \widehat{\boldsymbol{C}}_i^b\, S_{bg}\, \boldsymbol{I}_3\, \tau^2 & S_{bg}\, \boldsymbol{I}_3\, \tau \end{bmatrix} \tag{48}$$

1.4.3 Covariance Propagation

The covariance can be propagated using $\boldsymbol{Q}$ and $\boldsymbol{\Phi}$:

$$\boldsymbol{P}(-) \doteq \boldsymbol{\Phi}\boldsymbol{P}(-)\boldsymbol{\Phi}^T + \boldsymbol{Q}_d \tag{49}$$

1.5 Measurement Update

The measurement used in the Kalman Filter update is the true orientation of the IMU, $\boldsymbol{C}_{b_o}^i$. A manipulation of eq. 18 places the measurement into the error state to be used in the Kalman Filter:

$$\delta\boldsymbol{C}_b^i \doteq \boldsymbol{C}_{b_o}^{i^T}\, \widehat{\boldsymbol{C}}_b^i(-) \tag{50}$$

$\boldsymbol{\rho}$ is then extracted from the error rotation matrix:

$$[\boldsymbol{\rho}\times] \doteq (\boldsymbol{I}_3 - \delta\boldsymbol{C}_b^i) \tag{51}$$

$\boldsymbol{\rho}$ therefore defines the measurement as:

$$\boldsymbol{y} = \boldsymbol{\rho} + \boldsymbol{\omega} \tag{52}$$

where $\boldsymbol{\omega} \sim N(\boldsymbol{0}, \boldsymbol{R})$. With $\boldsymbol{R}$ is the covariance matrix representing the process noise. The noise is small compared to the true attitude of the device at $\sigma_\omega = 1 * 10^{-5}$ rad.

$$\boldsymbol{R} = \sigma_\omega \boldsymbol{I}_3 \tag{53}$$

Based on $\delta\boldsymbol{x}$ the measurement estimate is defined as:

$$\widehat{\boldsymbol{y}} \doteq \boldsymbol{H}\, \delta\widehat{\boldsymbol{x}}(-) \tag{54}$$

where $\boldsymbol{H} = \begin{bmatrix}\boldsymbol{I}_3 & \boldsymbol{0}_3\end{bmatrix}$. The residual is already computed from extracting $\boldsymbol{\rho}$ from the error rotation matrix:

$$\boldsymbol{r} = \boldsymbol{\rho} \tag{55}$$

it's covariance is computed as:

$$\boldsymbol{S} \doteq \boldsymbol{H}\, \boldsymbol{P}(-)\, \boldsymbol{H}^T + \boldsymbol{R} \tag{56}$$

The Kalman gain is then computed:

$$\boldsymbol{K} \doteq \boldsymbol{P}(-)\, \boldsymbol{H}^T\, \boldsymbol{S}^{-1} \tag{57}$$

updating the covariance and the error state:

$$\boldsymbol{P}(+) \doteq \boldsymbol{P}(-) - \boldsymbol{K}\, (\boldsymbol{H}\, \boldsymbol{P}(-)) \tag{58}$$

$$\delta\widehat{\boldsymbol{x}}(+) \doteq \delta\widehat{\boldsymbol{x}}(-) + \boldsymbol{K}\, \boldsymbol{r} \tag{59}$$

The new error state is now used to update the attitude rotation matrix.

$$\widehat{\boldsymbol{C}}_b^i(+) \doteq \widehat{\boldsymbol{C}}_b^i(-)\left(\boldsymbol{I}_3 + [\widehat{\boldsymbol{\rho}}(+)\times]\right)^T \tag{60}$$

From this updated rotation matrix the improved roll, pitch and yaw are extracted using eqns. 8 - 10. The bias is then updated from the extracted difference in bias from the error state:

$$\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}(+) = \widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}(-) + \delta\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}(+) \tag{61}$$

1.6 Power Spectral Density Determination

The Allen Variance is related to the two-sided PSD by

$$\sigma_u^2(\tau) = 4\int_0^\infty S_u(f)\,\frac{sin^4(\pi\, f\,\tau)}{(\pi\, f\,\tau)^2}df. \tag{62}$$

This equation shows the AV being proportional to the total noise power in the signal u.

The power spectral $S_u(f)$ shown in eqn. 62 can be represented as a power series in frequency f,

$$S_u(f) = \cdots + N^2 + \frac{B^2}{2\pi\, f} + \frac{K^2}{(2\pi\, f)^2} + \cdots. \tag{63}$$

This form of the PSD is convenient since by superposition is corresponds to the power spectrum of the signal

$$u(t) = \cdots + z_N(t) + z_B(t) + z_K(t) + \cdots \tag{64}$$

where the signals $z_N(t)$, $z_B(t)$, and $z_K(t)$ are mutually independent, zero-mean noise processes. Using this assumption the AV also takes on the superimposed form:

$$\sigma_u^2(\tau) = \cdots + \sigma_{ZN}^2(\tau) + \sigma_{ZB}^2(\tau) + \sigma_{ZK}^2(\tau) + \cdots \tag{65}$$

While any number of terms can be included in the power series representation of eqn. 63 in commercial grade IMUs the N, B and K terms are typically dominant.

1.6.1 Random Walk Errors: Angular: $z_N(t)$

The PSD term N^2 in 63 is constant with respect to frequency, which corresponds to the power sprectrum of white noise:

$$z_N(t) = \omega_N(t) \tag{66}$$

where $\omega_N(t)$ is white Gaussian random noise with PSD:

$$S_N = N^2. \tag{67}$$

In manufacturer specifications, generally in manufacturer specifications this error is called *angular random walk*.

By applying the transformation of 62 to 67 yeilds:

$$\sigma_{ZN}^2 = \frac{N^2}{\tau} \text{ or } \sigma_{ZN} = \frac{N}{\tau^{1/2}} \tag{68}$$

1.6.2 Bias Instability: $z_B(t)$

The $S_{ZB}(f) = (B^2/2\pi\, f)$ is referred to as *bias instability* (or flicker noise). According to [3] during the portion where the bias instability contributes most to the maximum value of the SD plot the contributing AV can be shown to be:

$$\sigma_{ZB}^2(\tau) \approxeq \frac{2B^2\, ln(2)}{\pi}. \tag{69}$$

Since the power spectrum of the bias instability term is not an even power of $s = j2\pi\, f$ there is no finite-order linear state space model that fits it exactly. Therefore, it is up to the designer to choose how to best approximate the PSD from the AV.

Gauss Markov Model

A first-order continuous-time Gauss-Markov Model is:

$$\dot{z}_G(t) = -\mu_b z_G(t) + \omega_B(t) \tag{70}$$

where $\mu_B = \frac{1}{T_B}$, with T_B representing the correlating time of the process and ω_B represents the white driving noise with PSD S_B.

The transfer function corresponding to this system is:

$$T(s) = \frac{1}{s + \mu_B} \tag{71}$$

which has the PSD:

$$S_{ZG}(\omega) = \frac{S_B}{\omega^2 + \mu_B^2} \tag{72}$$

as shown in [3] the power spectral density for larger cluster times can be solved for using:

$$S_B = \frac{2B^2\, ln(2)}{\pi(0.4365)^2\, T_B}. \tag{73}$$

If the manufacturer does not provide T_B is can be selected so that the flat region of the ASD plot lies near $1.89\, T_B$

1.6.3 PSD Implementation

For the Epson G730 the manufacturer supplies only the ASD and the ASD plots. For the Angular Random Walk Error they provide the ASD: $\sigma_{ZN}(\tau) = 0.6\frac{deg}{\sqrt{(s)}}$, first converting to the system units yields: $\sigma_{ZN}(\tau) = 1.7453 * 10^{-4}\frac{rad}{\sqrt{(s)}}$. Using eqn. 68 the power spectral density can easily be computed:

$$S_{rg} = S_N = ((1.7453 * 10^{-4}) * (\sqrt{0.08}))^2 = 2.3638 * 10^{-10} rad^2. \tag{74}$$

Similarly the manufacture gave the ASD for the angular bias instability as: $\sigma_{ZB}(\tau) = 0.8\frac{deg}{hr}$, converting this to the system's units yeilds: $\sigma_{ZB}(\tau) = 8.7266 * 10^{-6}\frac{rad}{s}$. Using the given ASD eqn. 69 is used to solve for the B^2 component:

$$B^2 = \frac{(8.7266 * 10^{-6})^2\pi}{2\, ln(2)} = 1.7258 * 10^{-10}\left(\frac{rad}{s}\right)^2 \tag{75}$$

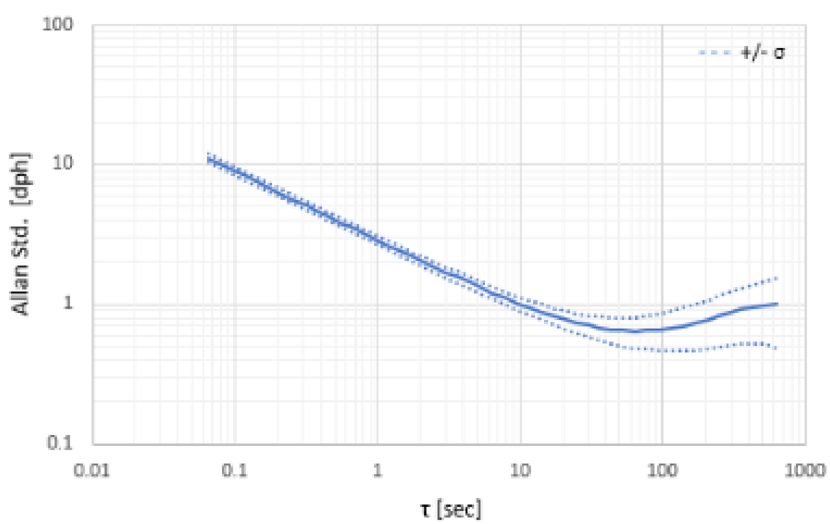


Figure 1: ASD plot provided by the Epson G370 manufacture in its data sheet.

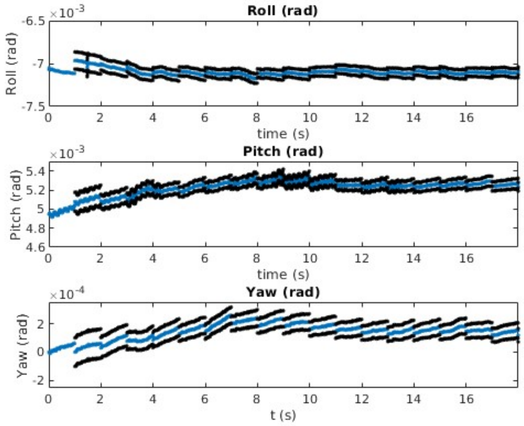
Looking at the ASD plot provided by the manufacture in fig. 1 the flat region occurs at about 70s, meaning that $T_B = \frac{70}{1.89} = 37.037s$. Using this and eqn. 73 the PSD is:

$$S_{bg} = S_b = 1.8624 * 10^{-21} rad^2. \tag{76}$$

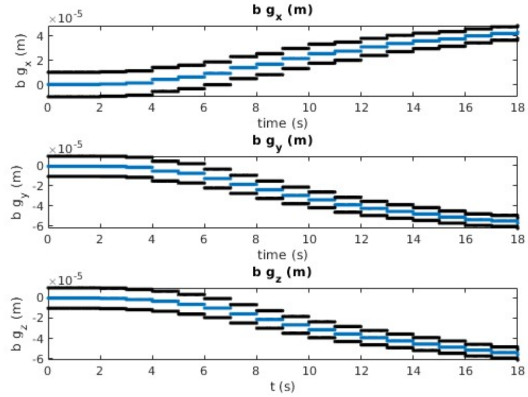
These newly calculated PSD can be placed into $\boldsymbol{Q}_c$ to calculate the measurement noise.

2 AHRS Implementation

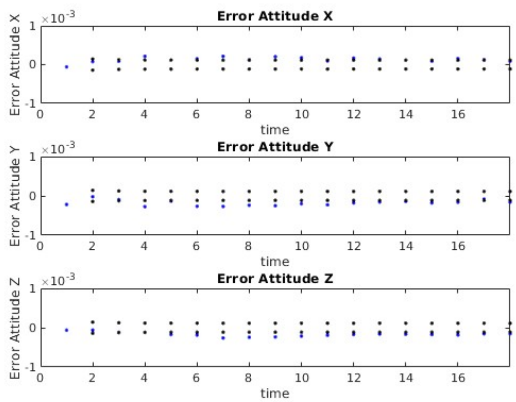
Using Section 1, the AHRS is implemented for the first 18s where the IMU is stationary this includes the Kalman Filter updates at 1s each.



(a) Calculated roll, pitch and yaw with extracted standard deviation from covariance matrix P.



(b) Calculated gyroscope bias with extracted standard deviation from covariance matrix P.



(c) Calculated residual with extracted standard deviation from covariance matrix S.

Figure 2: Implementation for AHRS for the first stationary 18s.

Fig. 2a shows the calculated roll, pitch and yaw for the time where the IMU is stationary. The orientation stays fairly constant, this is expected since the IMU is not moving. The uncertainty grows but then shrinks every time there is a measurement, this is further shown in Fig. 3 The covariance of the small angle error, $\boldsymbol{\rho}$ is shown. This covariance is the same as the roll, pitch, and yaw's when the IMU is stationary. The covariance for $\boldsymbol{\rho}$ is not shown for the first 1s is not shown in the plot because it had been set very large and therefore is outside of the bounds for the axis. It is only after the KF update that the uncertainty shrinks enough to be shown in the Figure.

Fig. 2b shoes the calculated gyroscope bias for the time the IMU is stationary, the bias is correcting itself overtime appearing to start settling on a value around the end of the 18s interval. The uncertainty also appears to be shrinking slightly after each KF update.

Fig. 2c shows the calculated residual for the same time. The residual uncertainty is very small this is expected since it is a function of the chosen $\boldsymbol{R}$. It appears the the sequence is somewhat white during this time interval.

Fig. 3 shows a closer look at the calculated roll, pitch and yaw each shown with one standard deviation. The roll plot also shows two data points the first at the beginning of the second and then one at the end.. Fitting a line to the growth of the

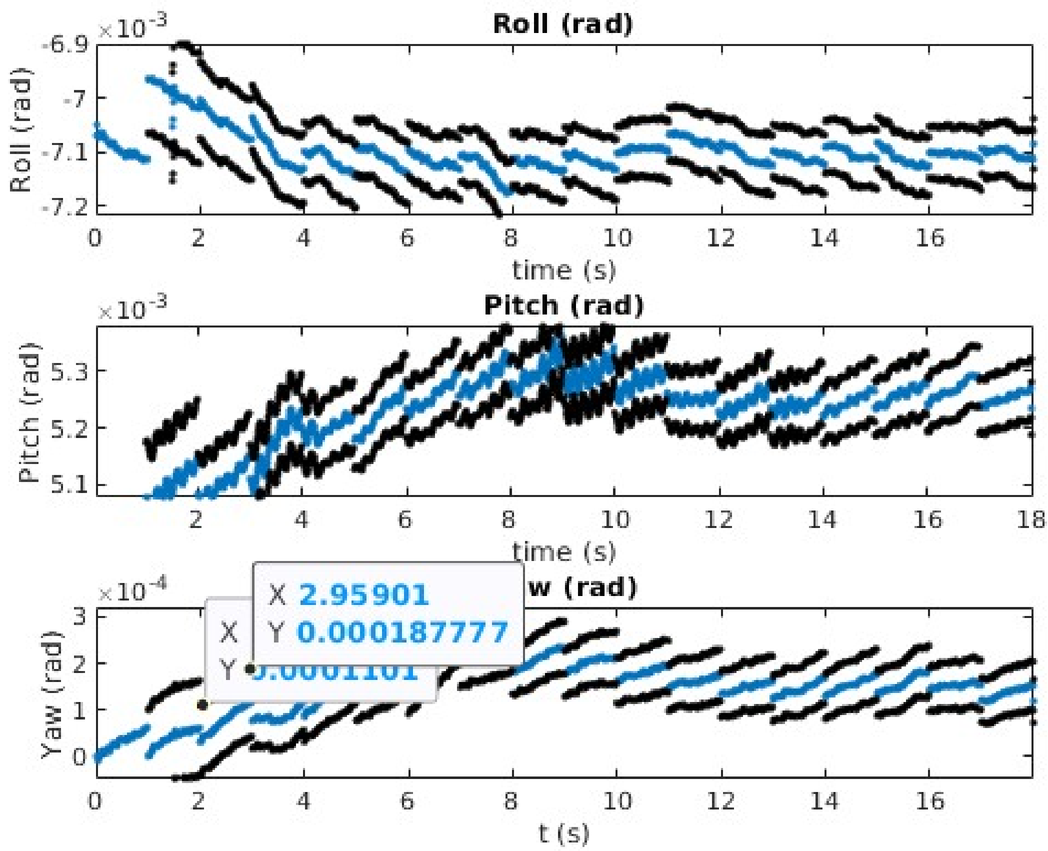


Figure 3: A zoomed in view of the calculated roll, pitch and yaw and the extracted uncertainty from the covariance matrix P.

uncertainty: $\frac{(1.877*10^{-4}-1.1101*10^{-4})}{1} = 7.76*10^{-5}\frac{rad}{s}$.

References

[1] J. A. Farrell, “Aided navigation: GPS with high rate sensors.” *McGraw Hill*, 2008.

[2] P. D. Groves, “Principles of GNSS, inertial and multisensor integrated navigation systems,” *Artech House*, 2013.

[3] J. A. Farrell, F. O. Silva, F. Rahman, and J. Wendel, “Inertial measurement unit error modeling tutorial: Inertial navigation system state estimation with real-time sensor calibration,” *IEEE Control Systems*, 2022.