HW2

June 2019

1 Q1

1.1 Q1.1

```cpp
// time
int imu_frequency = 200;
int cam_frequency = 30;
double imu_timestep = 1./imu_frequency;
double cam_timestep = 1./cam_frequency;
double t_start = 0;
double t_end = 3600 * 4;  //

// noise
double gyro_bias_sigma = 0.00005;
double acc_bias_sigma = 0.0005;

//double gyro_bias_sigma = 1.0e-5;
//double acc_bias_sigma = 0.0001;

double gyro_noise_sigma = 0.015;    // rad/s
double acc_noise_sigma = 0.019;      // m/(s^2)
```

Figure 1: Set Parameters

```yaml
%YAML:1.0
---
type: IMU
name: mytest
Gyr:
    unit: " rad/s"
    avg-axis:
        gyr_n: 2.1381116736589670e-01
        gyr_w: 7.7182588413203946e-04
    x-axis:
        gyr_n: 2.1550456104400748e-01
        gyr_w: 4.1096264824975174e-04
    y-axis:
        gyr_n: 2.1301057576929119e-01
        gyr_w: 8.5177284883701391e-04
    z-axis:
        gyr_n: 2.1291836528439143e-01
        gyr_w: 1.0527421553093526e-03
Acc:
    unit: " m/s^2"
    avg-axis:
        acc_n: 2.6791555630780811e-01
        acc_w: 3.2610301701998510e-03
    x-axis:
        acc_n: 2.6617692354504346e-01
        acc_w: 3.2849187775307060e-03
    y-axis:
        acc_n: 2.6876936764994164e-01
        acc_w: 3.1118467348363651e-03
    z-axis:
        acc_n: 2.6880037772843923e-01
        acc_w: 3.3863249982324815e-03
```

Figure 2: Estimated parameters from Allan curve

```
// time
int imu_frequency = 200;
int cam_frequency = 30;
double imu_timestep = 1./imu_frequency;
double cam_timestep = 1./cam_frequency;
double t_start = 0;
double t_end = 3600 * 4;  //


//param set here are discrete
//will process it to get continous values when adding to the imu data

// noise
//double gyro_bias_sigma = 0.00005;
//double acc_bias_sigma = 0.0005;

double gyro_bias_sigma = 1.0e-5;
double acc_bias_sigma = 0.0001;

double gyro_noise_sigma = 0.03;    // rad/s
double acc_noise_sigma = 0.019;      // m/(s^2)
```

Figure 3: Set Parameters

```yaml
%YAML:1.0
---
type: IMU
name: mytest
Gyr:
   unit: " rad/s"
   avg-axis:
      gyr_n: 4.1821896609133957e-01
      gyr_w: 6.3082294890385432e-04
   x-axis:
      gyr_n: 4.1426679891800322e-01
      gyr_w: 5.7752372532475353e-04
   y-axis:
      gyr_n: 4.1991849205255849e-01
      gyr_w: 5.9668582097415087e-04
   z-axis:
      gyr_n: 4.2047160730345695e-01
      gyr_w: 7.1825930041265835e-04
Acc:
   unit: " m/s^2"
   avg-axis:
      acc_n: 2.6841506465924975e-01
      acc_w: 1.4242219326208094e-03
   x-axis:
      acc_n: 2.6818238656844645e-01
      acc_w: 1.1643908486186996e-03
   y-axis:
      acc_n: 2.6711648188117293e-01
      acc_w: 1.6482879684031799e-03
   z-axis:
      acc_n: 2.6994632552812992e-01
      acc_w: 1.4599869808405485e-03
```

Figure 4: Estimated parameters from Allan curve

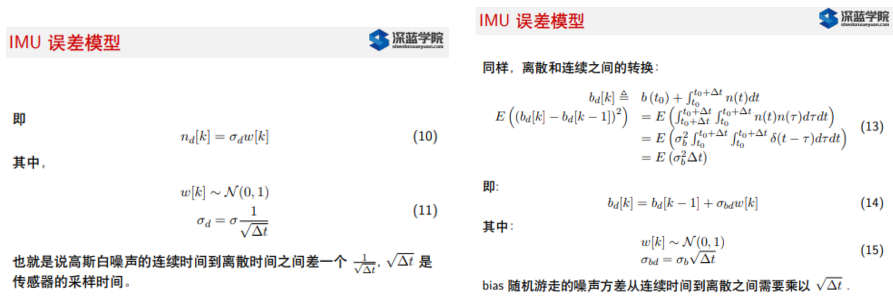
IMU Noise Values

Parameter	YAML element	Symbol	Units
Gyroscope "white noise"	`gyr_n`	σ_g	$\frac{rad}{s}\frac{1}{\sqrt{Hz}}$
Accelerometer "white noise"	`acc_n`	σ_a	$\frac{m}{s^2}\frac{1}{\sqrt{Hz}}$
Gyroscope "bias Instability"	`gyr_w`	σ_{bg}	$\frac{rad}{s}\sqrt{Hz}$
Accelerometer "bias Instability"	`acc_w`	σ_{ba}	$\frac{m}{s^2}\sqrt{Hz}$

- White noise is at tau=1;
- Bias Instability is around the minimum;

(according to technical report: `Allan Variance: Noise Analysis for Gyroscopes`)

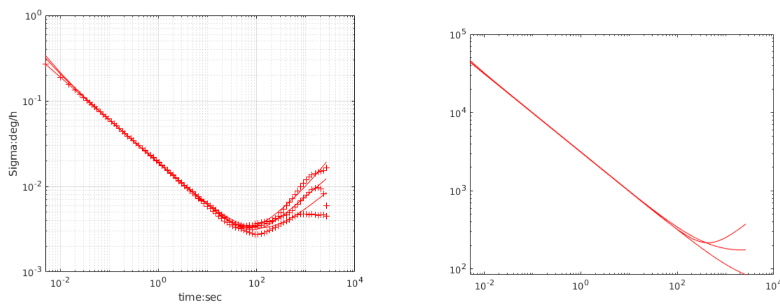
Figure 5: Units of the estimated parameters



(a) (b)

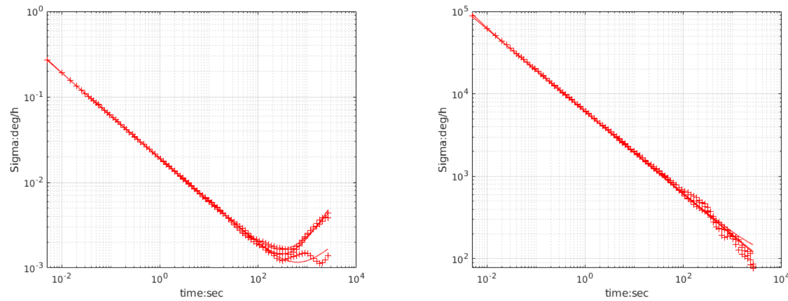
Figure 6: Relationship between discrete and continuous parameters

We can valid the estimated parameters using relationships shown in Figure 6 with the set ground truth discrete values for parameters.



(a) (b)

Figure 7: Allan curve for the first parameters. (a) Allan curve for acceleration. (b) Allan curve for gyro



(a) (b)

Figure 8: Allan curve for the second parameters. (a) Allan curve for acceleration. (b) Allan curve for gyro

1.2 Q1.2

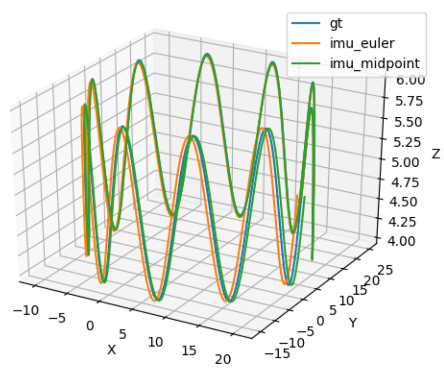


Figure 9: Result of Euler integration and Mid Point integration

2 Q2

The standard bias function of a B-Spline curve of degree $k-1$ is given by:

$$p(t) = \sum_{i=0}^{n} p_i B_{i,k}(t) \tag{1}$$

where $p_i \in R^N$ are control points at time t_i and $B_{i,k}(t)$ are basis functions.

Equation 1 can be re-arranged into its cumulative form as

$$p(t) = p_0 \tilde{B}_{0,k}(t) + \sum_{i=1}^{n} (p_i - p_{i-1}) \tilde{B}_{i,k}(t) \tag{2}$$

where

$$\tilde{B}_{i,k}(t) = \sum_{j=i}^{n} B_{j,k}(t) \tag{3}$$

We can represent the control pose p_i in $SE3$, such as

$$p_0 = T_{w,0} \tag{4}$$

and represent trajectories in $SE3$ by substituting the control point differences with the logarithmic map

$$\Omega_i = \log\left(T_{w,i-1}^{-1} T_{w,i}\right) \tag{5}$$

Then the pose $T_{w,s}(t)$ along the spline at time t can be represented as

$$T_{w,s}(u) = p(t) = \exp\left(\tilde{B}_{0,k}(t)\log\left(T_{w,0}\right)\right) \prod_{i=1}^{n} \exp\left(\tilde{B}_{i,k}(t)\Omega_i\right) \tag{6}$$

where the subscript w represents that the pose at time t and control poses are given in the world coordinate.

We focus on the particular case of cumulative B-Splines (k=4). We assume a uniform time interval between control points.

The the pose in the spline trajectory can now be defined as:

$$T_{w,s}(u) = T_{w,i-1} \prod_{j=1}^{3} \exp\left(\tilde{B}(u)_j \Omega_{i+j}\right) \tag{7}$$

where Ω_{i+j} relates the corresponding relative transformation, and $\tilde{B}(u)_j$ represents the $j-th$ element of 0-based $\tilde{B}(u)$. $\tilde{B}(u)$ is the matrix representation of a cumulative basis. We can also solve the first and the second derivative of this matrix.

As a result, the first and second derivative of the pose can be represented as:

$$\dot{\mathbf{T}}_{w,s}(u) = \mathbf{T}_{w,i-1}\left(\dot{\mathbf{A}}_0 \mathbf{A}_1 \mathbf{A}_2 + \mathbf{A}_0 \dot{\mathbf{A}}_1 \mathbf{A}_2 + \mathbf{A}_0 \mathbf{A}_1 \dot{\mathbf{A}}_2\right) \tag{8}$$

$$\ddot{\mathbf{T}}_{w,s}(u) = \mathbf{T}_{w,i-1} \begin{pmatrix} \ddot{\mathbf{A}}_0\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_0\ddot{\mathbf{A}}_1\mathbf{A}_2 + \mathbf{A}_0\mathbf{A}_1\ddot{\mathbf{A}}_2 + \\ 2\left(\mathbf{A}_0\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_0\mathbf{A}_1\mathbf{A}_2 + \mathbf{A}_0\mathbf{A}_1\dot{\mathbf{A}}_2\right) \end{pmatrix} \tag{9}$$

where

$$\mathbf{A}_j = \exp\left(\Omega_{i+j}\tilde{\mathbf{B}}(u)_j\right), \quad \dot{\mathbf{A}}_j = \mathbf{A}_j\Omega_{i+j}\dot{\tilde{\mathbf{B}}}(u)_j \tag{10}$$

$$\ddot{\mathbf{A}}_j = \dot{\mathbf{A}}_j\Omega_{i+j}\dot{\tilde{\mathbf{B}}}(u)_j + \mathbf{A}_j\Omega_{i+j}\ddot{\tilde{\mathbf{B}}}(u)_j \tag{11}$$

Furthermore, accelerometer and gyroscope measurements can be written as

$$\begin{aligned} \mathrm{Gyro}(u) &= \mathbf{R}_{w,s}^{\top}(u) \cdot \dot{\mathbf{R}}_{w,s}(u) + \text{ bias} \\ \mathrm{Accel}(u) &= \mathbf{R}_{w,s}^{\top}(u) \cdot (\ddot{\mathbf{s}}_w(u) + g_w) + \text{ bias} \end{aligned} \tag{12}$$

where $\dot{\mathbf{R}}_{w,s}$ and $\ddot{\mathbf{s}}_w$ are sub-matrices of $\dot{\mathbf{T}}_{w,s}$ and $\ddot{\mathbf{T}}_{w,s}$.

Then, we can solve for our spline and camera parameters in batch or over a window by minimizing a objective function formed by the re-projection errors and inertial errors.

$$\begin{aligned} E(\boldsymbol{\theta}) = &\sum_{\hat{\mathbf{p}}_m} \left(\hat{\mathbf{p}}_m - \mathcal{W}\left(\mathbf{p}_r; \mathbf{T}_{c,s}\mathbf{T}_{w,s}\left(u_m\right)^{-1}\mathbf{T}_{w,s}\left(u_r\right)\mathbf{T}_{s,c}, \rho\right)\right)_{\Sigma_p}^2 + \\ &\sum_{\hat{\omega}_m} \left(\hat{\omega}_m - \mathrm{Gyro}\left(u_m\right)\right)_{\Sigma_\omega}^2 + \sum_{\hat{\mathbf{a}}_m} \left(\hat{\mathbf{a}}_m - \mathrm{Accel}\left(u_m\right)\right)_{\Sigma_{\mathbf{a}}}^2 \end{aligned} \tag{13}$$