Introduction to Multicopter Design and Control

Lesson 09 State Estimation

Quan Quan , Associate Professor

qq_buaa@buaa.edu.cn

BUAA Reliable Flight Control Group, http://rfly.buaa.edu.cn/

Beihang University, China

Outline

1. Attitude estimation
2. Position estimation
3. Velocity estimation
4. Obstacle estimation
5. Conclusion

1. Attitude estimation

☐ Measuring Principle

（1）Pitch angle and roll angle measuring principle

According to the model in Lesson 6, by ignoring the cross term of velocity and angular velocity, the specific force ${}^{\mathrm{b}}\mathbf{a}_{\mathrm{m}}$ satisfies

$$\begin{bmatrix} a_{x_{\mathrm{b}}\mathrm{m}} \\ a_{y_{\mathrm{b}}\mathrm{m}} \end{bmatrix} = \begin{bmatrix} \dot{v}_{x_{\mathrm{b}}} + g\sin\theta \\ \dot{v}_{y_{\mathrm{b}}} - g\sin\phi\cos\theta \end{bmatrix} \approx \begin{bmatrix} -\dfrac{k_{\mathrm{drag}}}{m} v_{x_{\mathrm{b}}} \\ -\dfrac{k_{\mathrm{drag}}}{m} v_{y_{\mathrm{b}}} \end{bmatrix}$$

The accelerometer measures the *specific force*, namely the non-gravitational force per unit mass. Also, it measures scaled velocity.

where, ${}^{\mathrm{b}}\mathbf{a}_{\mathrm{m}} = \begin{bmatrix} a_{x_{\mathrm{b}}\mathrm{m}} & a_{y_{\mathrm{b}}\mathrm{m}} & a_{z_{\mathrm{b}}\mathrm{m}} \end{bmatrix}^{\mathrm{T}}$ is the measurement from the accelerometer. when the angular velocity is small so that the system is in static equilibrium, i.e., $\dot{v}_{x_{\mathrm{b}}} = \dot{v}_{y_{\mathrm{b}}} = 0$

$$\Rightarrow \quad \begin{bmatrix} a_{x_{\mathrm{b}}\mathrm{m}} \\ a_{y_{\mathrm{b}}\mathrm{m}} \end{bmatrix} = \begin{bmatrix} g\sin\theta \\ -g\sin\phi\cos\theta \end{bmatrix}$$

1. Attitude estimation

☐ Measuring Principle

（1） Pitch angle and roll angle measuring principle

Therefore, observation of low-frequency pitch and roll angle can be acquired by accelerometer measurement illustrated as

$$\theta_{\mathrm{m}} = \arcsin\left(\frac{a_{x_{\mathrm{b}}\mathrm{m}}}{g}\right)$$

$$\phi_{\mathrm{m}} = -\arcsin\left(\frac{a_{y_{\mathrm{b}}\mathrm{m}}}{g\cos\theta_{\mathrm{m}}}\right)$$

where ${}^{\mathrm{b}}\mathbf{a}_{\mathrm{m}} = \begin{bmatrix} a_{x_{\mathrm{b}}\mathrm{m}} & a_{y_{\mathrm{b}}\mathrm{m}} & a_{z_{\mathrm{b}}\mathrm{m}} \end{bmatrix}^{\mathrm{T}}$ is the measurement from the accelerometer.

P.S. If the amplitude of the vibration is large, $a_{x_{\mathrm{b}}\mathrm{m}}, a_{y_{\mathrm{b}}\mathrm{m}}$ would be polluted by noise severely and further affect the estimation of $\theta_{\mathrm{m}}, \phi_{\mathrm{m}}$. Thus, the vibration damping is very important.

1. Attitude estimation

☐ Measuring Principle

（2） Yaw angle measuring principle

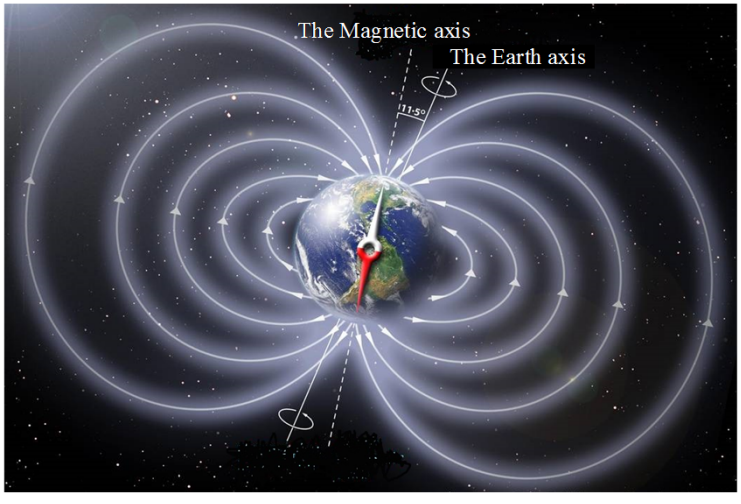


Figure 9.1. Schematic diagram of Earth‘s magnetic field. (photo courtesy of Peter Reid from the University of Edinburgh)

Figure 9.1 illustrates that the Earth‘s field points down towards the north in the northern hemisphere, points up towards the north in the southern hemisphere. It is horizontal and points north at the equator. In all cases, the direction of the Earth's field is always pointing to the magnetic north, which is called the Earth's magnetic pole position. But it differs from true, or geographic, north by about 11.5 degrees. At different locations around the globe magnetic north, the true north can differ by 25 degrees. This difference is called the declination angle and can be determined from a lookup table based on the geographic location. The key to accurately finding a compass heading is a two-step process.

1. Attitude estimation

☐ Measuring Principle

（2）Yaw angle measuring principle

1）First, determine the magnetic field direction in the horizontal plane of the vector and then obtain azimuth.

Suppose that the magnetometer measurements are ${}^{\mathrm{b}}\mathbf{m}_{\mathrm{m}} = \begin{bmatrix} m_{x_{\mathrm{b}}} & m_{y_{\mathrm{b}}} & m_{z_{\mathrm{b}}} \end{bmatrix}^{\mathrm{T}}$. Since a magnetometer is possibly not horizontally placed, the angles $(\theta_{\mathrm{m}}, \phi_{\mathrm{m}})$ measured by a dual-axis tilt sensor are used to project the magnetometer measurement onto the horizontal plane as[1]

$$\bar{m}_{x_{\mathrm{e}}} = m_{x_{\mathrm{b}}} \cos\theta_{\mathrm{m}} + m_{y_{\mathrm{b}}} \sin\phi_{\mathrm{m}} \sin\theta_{\mathrm{m}} + m_{z_{\mathrm{b}}} \cos\phi_{\mathrm{m}} \sin\theta_{\mathrm{m}}$$

$$\bar{m}_{y_{\mathrm{e}}} = m_{y_{\mathrm{b}}} \cos\phi_{\mathrm{m}} - m_{z_{\mathrm{b}}} \sin\phi_{\mathrm{m}}$$

where $\bar{m}_{x_{\mathrm{e}}}, \bar{m}_{y_{\mathrm{e}}}$ are the horizontal projections of magnetometer readings.

[1] Caruso M J. Applications of magnetoresistive sensors in navigation systems[R]. SAE Technical Paper, 1997.

1. Attitude estimation

☐ Measuring Principle

（2） Yaw angle measuring principle

1） First, determine the magnetic field direction in the horizontal plane of the vector and then obtain azimuth.

Let $\psi_{\text{mag}} \in [0, 2\pi]$. Then:

$$\psi_{\text{mag}} = \begin{cases} \pi - \tan^{-1}\left(\bar{m}_{y_e} / \bar{m}_{x_e}\right) & \text{if } \bar{m}_{x_e} < 0 \\ 2\pi - \tan^{-1}\left(\bar{m}_{y_e} / \bar{m}_{x_e}\right) & \text{if } \bar{m}_{x_e} > 0,\ \bar{m}_{y_e} > 0 \\ -\tan^{-1}\left(\bar{m}_{y_e} / \bar{m}_{x_e}\right) & \text{if } \bar{m}_{x_e} > 0,\ \bar{m}_{y_e} < 0 \\ \pi/2 & \text{if } \bar{m}_{x_e} = 0,\ \bar{m}_{y_e} < 0 \\ 3\pi/2 & \text{if } \bar{m}_{x_e} = 0,\ \bar{m}_{y_e} > 0 \end{cases}$$

Let $\psi_{\text{mag}} \in [-\pi, \pi]$. Then:

$$\psi_{\text{mag}} = \arctan 2\left(\bar{m}_{y_e}, \bar{m}_{x_e}\right)$$

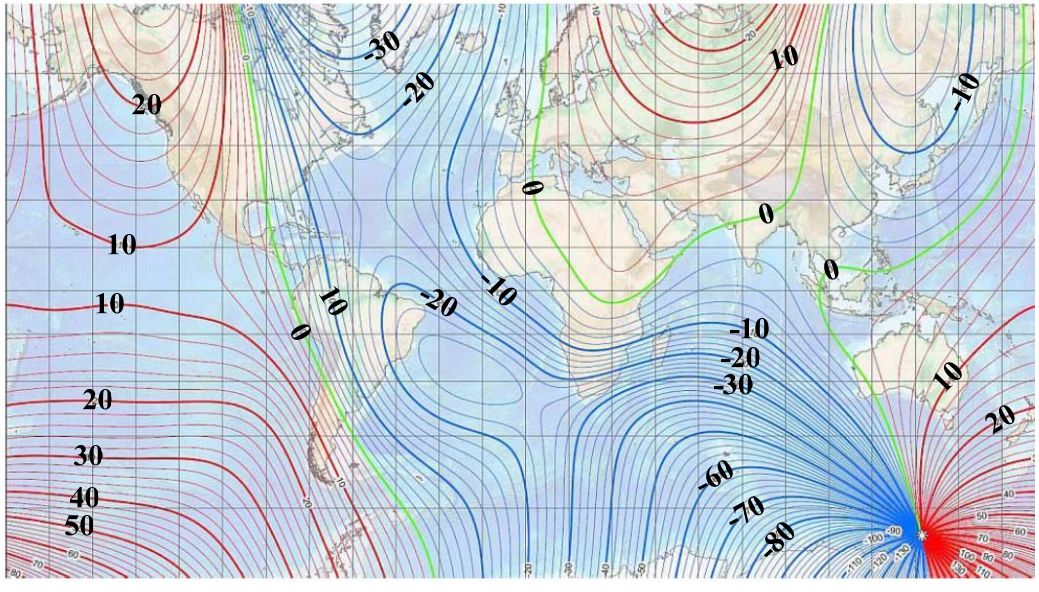
If a multicopter turns clockwise, the yaw angle is positive.

1. Attitude estimation

☐ Measuring Principle

（2） Yaw angle measuring principle

2） Secondly, the yaw angle is corrected by adding or subtracting a declination.



The magnetic field orientation of Beijing is 6 degrees west of north, then 6 degrees are added to the magnetic field orientation to obtain the north direction.

Figure 9.3. 2015 world magnetic field declination contour map, adapted from http://www.ngdc.noaa.gov/geomag/WMM/image.shtml.

1. Attitude estimation

☐ Measuring Principle

（2） Yaw angle measuring principle

2） Secondly, the yaw angle is corrected by adding or subtracting a declination.

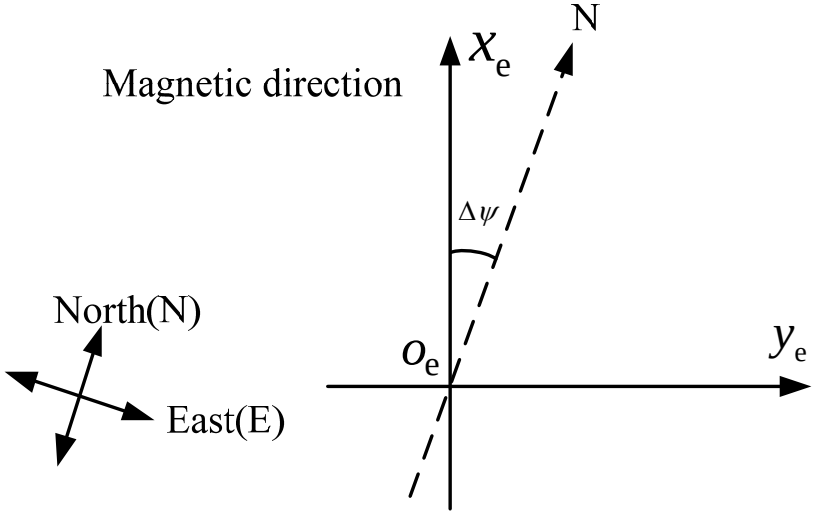


Figure 9.2. Local magnetic direction and the north direction

When multicopters are under the Semi-Autonomous Control (SAC) manner, the direction of the local magnetic field can be chosen as the x_e axis of the Earth-Fixed Coordinate Frame (EFCF)

During a mission under the Fully-Autonomous Control (FAC) manner, the x_e axis has to point to the north in order to be consistent with the latitude and longitude. Then declination is required to correct the magnetic direction to the north direction.

1. Attitude estimation

☐ Measuring Principle

（2）Yaw angle measuring principle

Besides magnetometers, for a large multicopter, a dual-antenna GPS receiver system can be used to estimate yaw angles measured by the two antennas on the multicopter head and rear, respectively. However, it is hard for a small multicopter to get a precise yaw angle limited by GPS measurement precision. Therefore, at present, yaw angles are mainly measured by magnetometers.

1. Attitude estimation

☐ Linear complementary filter

The attitude rates $\dot{\theta},\dot{\phi},\dot{\psi}$ and angular velocity ${}^{\mathrm{b}}\boldsymbol{\omega}=\begin{bmatrix}\omega_{x_\mathrm{b}} & \omega_{y_\mathrm{b}} & \omega_{z_\mathrm{b}}\end{bmatrix}^{\mathrm{T}}$ have the following relationship:

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}1 & \tan\theta\sin\phi & \tan\theta\cos\phi\\ 0 & \cos\phi & -\sin\phi\\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta\end{bmatrix}\begin{bmatrix}\omega_{x_\mathrm{b}}\\ \omega_{y_\mathrm{b}}\\ \omega_{z_\mathrm{b}}\end{bmatrix}$$

Since multicopters often work under condition that $\theta\approx 0,\phi\approx 0$, the above equation is approximated as

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}\approx\begin{bmatrix}\omega_{x_\mathrm{b}}\\ \omega_{y_\mathrm{b}}\\ \omega_{z_\mathrm{b}}\end{bmatrix}$$

The attitude can be estimated by the accelerometers and magnetometers with large noise but small drifts. On the other hand, integrating the angular velocity will result in the attitude angle with small noise but large drifts. The basic idea of complementary filtering is to use their complementary characteristics to obtain more accurate attitude estimation.

1. Attitude estimation

☐ Linear complementary filter

(1) Pitch angle

The Laplace transform of pitch angle θ is expressed as

$$\theta(s)=\frac{1}{\tau s+1}\theta(s)+\left(1-\frac{1}{\tau s+1}\right)\theta(s)$$

Low-pass filter, $\tau \in \mathbb{R}_+$ is a time constant

High-pass filter $\frac{\tau s}{\tau s+1}=1-\frac{1}{\tau s+1}$

1) Since the pitch angle obtained by an accelerometer has a large noise but a small drift, for simplicity, it is modeled as

$$\theta_{\mathrm{m}}=\theta+n_\theta$$

where n_θ indicates high-frequency noise, θ is the true value of the pitch angle.

2) Considering that the pitch angle estimated by integrating angular velocity has a little noise but a large drift, the integration is modeled as $\frac{\omega_{y_{\mathrm{b}}\mathrm{m}}(s)}{s}=\theta(s)+c\frac{1}{s}$

Laplace transform of the integration of angular velocity

Laplace transform of the constant drift

1. Attitude estimation

☐ Linear complementary filter

(1) Pitch angle

The Laplace transform of pitch angle θ is expressed as

$$\theta(s)=\frac{1}{\tau s+1}\theta(s)+\left(1-\frac{1}{\tau s+1}\right)\theta(s)$$

Low-pass filter, $\tau \in \mathbb{R}_+$ is a time constant

High-pass filter $\frac{\tau s}{\tau s+1}=1-\frac{1}{\tau s+1}$

The standard form of a linear complementary filter is expressed as

$$\hat{\theta}(s)=\frac{1}{\tau s+1}\theta_{\mathrm{m}}(s)+\frac{\tau s}{\tau s+1}\left(\frac{1}{s}\omega_{y_{\mathrm{b}}\mathrm{m}}(s)\right)$$

Pitch angle estimated by an accelerometer

Pitch angle estimated by integrating angular velocity

1. Attitude estimation

☐ Linear complementary filter

(1) Pitch angle

The Laplace transform of pitch angle θ is expressed as

$$\theta(s)=\frac{1}{\tau s+1}\theta(s)+\left(1-\frac{1}{\tau s+1}\right)\theta(s)$$

Low-pass filter, $\tau \in \mathbb{R}_+$ is a time constant

High-pass filter $\frac{\tau s}{\tau s+1}=1-\frac{1}{\tau s+1}$

The linear complementary filter is expressed as the transfer function form

$$\hat{\theta}(s)=\frac{1}{\tau s+1}\theta_{\mathrm{m}}(s)+\frac{\tau s}{\tau s+1}\left(\frac{1}{s}\omega_{y_{\mathrm{b}}\mathrm{m}}(s)\right)$$

Pitch angle estimated by an accelerometer

Pitch angle estimated by integrating angular velocity

$$\hat{\theta}(s)=\theta(s)+\left[\frac{1}{\tau s+1}n_\theta(s)+\frac{\tau s}{\tau s+1}c\frac{1}{s}\right]$$

Attenuated to zero

$$\hat{\theta}(s)\approx\theta(s)$$

1. Attitude estimation

☐ Linear complementary filter

(1) Pitch angle

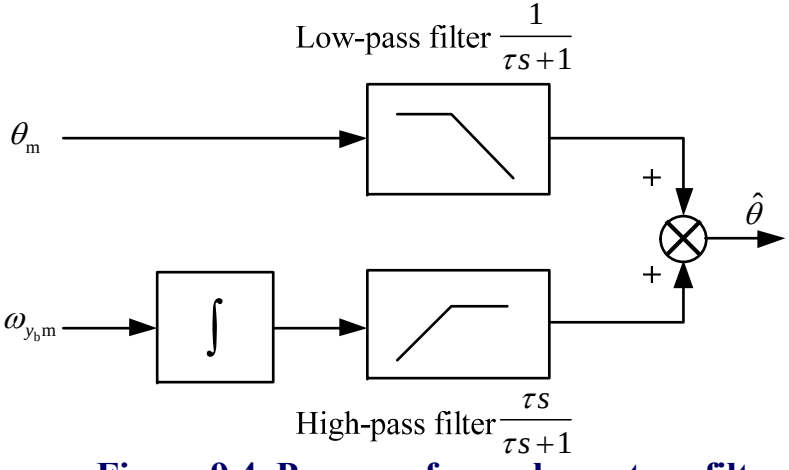


Figure 9.4. Process of complementary filter.

Transfer the filter to a discrete-time differential form

$$\hat{\theta}(s)=\frac{1}{\tau s+1}\theta_m(s)+\frac{\tau s}{\tau s+1}\left(\frac{1}{s}\omega_{y_b m}(s)\right)$$

Through the first-order backward difference [2],

$$s=\left(1-z^{-1}\right)/T_s$$

$T_s\in\mathbb{R}_+$ is the filtering sampling period

Furthermore,

$$\hat{\theta}(z)=\frac{1}{\tau\frac{1-z^{-1}}{T_s}+1}\theta_m(z)+\frac{\tau}{\tau\frac{1-z^{-1}}{T_s}+1}\omega_{y_b m}(z)$$

Transform into a discrete-time difference form as

$$\hat{\theta}(k)=\frac{\tau}{\tau+T_s}\left(\hat{\theta}(k-1)+T_s p_{bm}(k)\right)+\frac{T_s}{\tau+T_s}\theta_m(k)$$

The low-frequency filter keeps the advantage that θ_m has a small drift, while the high-frequency filter keeps the advantage that $\omega_{y_b m}(s)/s$ has a little noise

[2]. Perdikaris G A. *Computer Controlled Systems*. Berlin: Springer-Netherlands, 1991.

1. Attitude estimation

☐ Linear complementary filter

(1) Pitch angle

If $\tau/(\tau+T_s)=0.95$,then $T_s/(\tau+T_s)=0.05$.

The complementary filter for the pitch angle is $\hat{\theta}(k)=0.95\left(\hat{\theta}(k-1)+T_s\omega_{y_b m}(k)\right)+0.05\theta_m(k)$

A Pixhawk is used for measurement and the complementary filter is used to estimate the pitch angle. Results are shown in Figure 9.5. It is observed that the pitch angle estimated by the complementary filter is correct and smooth, while that derived by integrating the angular velocity is divergent. The conclusion is the same for the roll angle.

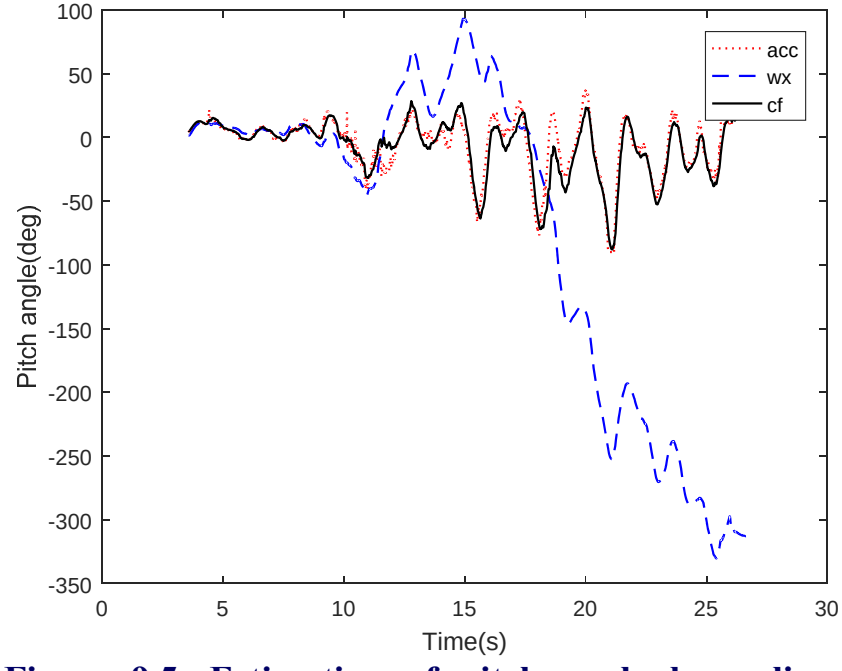


Figure 9.5. Estimation of pitch angle by a linear complementary filter. In this figure, acc, wx and cf denote the pitch angle estimated by acceleration, angular velocity integration and complementary filter, respectively.

1. Attitude estimation

☐ Linear complementary filter

（2） Yaw angle

Yaw angle can be measured by both GPS and electronic compass, denoted by ψ_{GPS} and ψ_{mag} ,respectively. A simple method of obtaining ψ_{m} is to sum the weighted measurement of the two sensors, written as

$$\psi_{\mathrm{m}} = \left(1-\alpha_{\psi}\right)\psi_{\mathrm{GPS}} + \alpha_{\psi}\psi_{\mathrm{mag}}$$

where $\alpha_{\psi} \in [0,1]$ is the weighting factor. Since the sampling periods of an electronic compass and a gyroscope are often higher than that of a GPS, the yaw angle can be obtained by

$$\psi_{\mathrm{m}}(k) = \begin{cases} \left(1-\alpha_{\psi}\right)\psi_{\mathrm{GPS}}(k) + \alpha_{\psi}\psi_{\mathrm{mag}}(k), & \text{when } \psi_{\mathrm{GPS}} \text{ is updated} \\ \psi_{\mathrm{mag}}(k), & \text{else} \end{cases}$$

After getting ψ_{m} , the yaw angle is estimated by

$$\hat{\psi}(k) = \frac{\tau}{\tau+T_{\mathrm{s}}}\left(\hat{\psi}(k-1) + T_{\mathrm{s}}\omega_{z_{\mathrm{b}}\mathrm{m}}(k)\right) + \frac{T_{\mathrm{s}}}{\tau+T_{\mathrm{s}}}\psi_{\mathrm{m}}(k)$$

1. Attitude estimation

☐ Nonlinear complementary filter

Similar to the linear complementary filter, they both benefit from the advantage of complementary characteristics of accelerometers and gyroscopes. The difference is that nonlinear complementary filters are based on a nonlinear relationship between the angular velocity and the angle of rotation.

Denote $\hat{\mathbf{R}}$ as a rotation matrix estimated by the complementary filter, $\mathbf{R}_m$ is a rotation matrix measured by accelerometers and magnetometers, and $\tilde{\mathbf{R}}$ is the error between $\hat{\mathbf{R}}$ and $\mathbf{R}_m$ defined as

$$\tilde{\mathbf{R}} = \hat{\mathbf{R}}^{\mathrm{T}}\mathbf{R}_m$$

The rotation matrix is filtered out by the following equation (detailed in [3])

$$\dot{\hat{\mathbf{R}}} = \left(\mathbf{R}_m{}^{b}\boldsymbol{\omega}_m + k_p\hat{\mathbf{R}}\boldsymbol{\xi}\right)_{\times}\hat{\mathbf{R}}$$

Angular Velocity by the gyroscope

Feedback gain

$$\boldsymbol{\xi} = \mathrm{vex}\left(\frac{1}{2}\left(\tilde{\mathbf{R}} - \tilde{\mathbf{R}}^{\mathrm{T}}\right)\right), \mathrm{vex}\left([\mathbf{x}]_{\times}\right) = \mathbf{x}$$

Angular measurement

[3].Mahony R, Hamel T, Pflimlin J M. Nonlinear complementary filters on the special orthogonal group [J]. IEEE Transactions on Automatic Control, 2008, 53(5): 1203-1218.

1. Attitude estimation

☐ Kalman filter

As shown before, a nonlinear complementary filter needs to use nine or twelve states and it is also hard to choose optimal parameters. For such a purpose, a Kalman filter is adopted to estimate the attitude [4]

$$\mathbf{x}=\begin{bmatrix}-\sin\theta\\ \sin\phi\cos\theta\\ \cos\phi\cos\theta\end{bmatrix}$$

$$\dot{\mathbf{R}}_{\mathrm{b}}^{\mathrm{e}}=\mathbf{R}_{\mathrm{b}}^{\mathrm{e}}\left[{}^{\mathrm{b}}\boldsymbol{\omega}\right]_{\times}\Rightarrow\dot{\mathbf{R}}_{\mathrm{e}}^{\mathrm{b}}=-\left[{}^{\mathrm{b}}\boldsymbol{\omega}\right]_{\times}\mathbf{R}_{\mathrm{e}}^{\mathrm{b}}$$

chosen from the third column

$$\dot{\mathbf{x}}=-\left[{}^{\mathrm{b}}\boldsymbol{\omega}\right]_{\times}\mathbf{x}$$

Process model

Since the accelerations along $x_{\mathrm{b}},y_{\mathrm{b}}$ axes are small, the measurement model is simplified as

$$\mathbf{C}^{\mathrm{T}}\cdot{}^{\mathrm{b}}\mathbf{a}_{\mathrm{m}}=-g\mathbf{C}^{\mathrm{T}}\cdot\mathbf{x}+\mathbf{n}_{\mathrm{a}}$$

Measurement model

where $\mathbf{C}=\begin{bmatrix}\mathbf{I}_2 & \mathbf{0}_{2\times1}\end{bmatrix}^{\mathrm{T}}\in\mathbb{R}^{3\times2},\mathbf{n}_{\mathrm{a}}\in\mathbb{R}^{2}.$

[4]. Kang C W, Park C G. Attitude estimation with accelerometers and gyros using fuzzy tuned Kalman filter[C]. Control Conference (ECC), 2009 European. IEEE, 2009: 3713-3718.

2. Position estimation

☐ GPS-based position estimation

GPS-based position estimation often utilizes an IMU, a GPS receiver and a barometer. The kinematic model of the multicopter and different information obtained by these sensors are then fused by a Kalman filter. ${}^{\mathrm{e}}\mathbf{p} = \begin{bmatrix} p_{x_\mathrm{e}} & p_{y_\mathrm{e}} & p_{z_\mathrm{e}} \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ is the absolute position.

The process model is formulated as

$$
\begin{aligned}
{}^{\mathrm{e}}\dot{\mathbf{p}} &= {}^{\mathrm{e}}\mathbf{v} \\
{}^{\mathrm{e}}\dot{\mathbf{v}} &= \mathbf{R}\left({}^{\mathrm{b}}\mathbf{a}_{\mathrm{m}} - \mathbf{b}_{\mathrm{a}} - \mathbf{n}_{\mathrm{a}}\right) + g\mathbf{e}_3 \\
\dot{\mathbf{b}}_{\mathrm{a}} &= \mathbf{n}_{\mathbf{b}_{\mathrm{a}}} \\
\dot{b}_{d_{\mathrm{baro}}} &= n_{b_{d_{\mathrm{baro}}}}
\end{aligned}
$$

The measurement model is derived as

$$
\begin{aligned}
p_{x\mathrm{GPS}} &= p_{x_\mathrm{e}} + n_{p_{x\mathrm{GPS}}} \\
p_{y\mathrm{GPS}} &= p_{y_\mathrm{e}} + n_{p_{y\mathrm{GPS}}} \\
d_{\mathrm{baro}} &= -p_{z_e} + b_{d_{\mathrm{baro}}} + n_{d_{\mathrm{baro}}}
\end{aligned}
$$

- What if considering the bias of GPS measurement?
- What if regarding the height by GPS as one observation?
- Consider the problems when measuring the height using GPS and barometers?

The sensor measurement models are referred to Lesson 7.

2. Position estimation

☐ SLAM-based position estimation

SLAM, Simultaneous Localization And Mapping, is a computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of the agent's location with it. Related references could be found in [5][6]. SLAM always uses different types of sensors, including distance sensors like ultrasonic range finders or laser range finders, direction sensors like cameras. Combination of distance sensors and direction sensors like 3D cameras is also commonly used. Several SLAM systems and SLAM datasets are listed in Table 9.1 and Table 9.2, respectively.

[5]. Whyte H, Baliey T. Simultaneous Localization and Mapping (SLAM) Part 1 The Essential Algorithms[J]. IEEE Robotics & Automation Magazine, 2006.

[6]. Bailey T, Durrant-Whyte H. Simultaneous localization and mapping (SLAM): Part II[J]. IEEE Robotics & Automation Magazine, 2006, 13(3): 108-117.

2. Position estimation

Table 9.1 Open source SLAM algorithms

Name	Description	Websites
CyrillStachniss, UdoFrese, Giorgio Grisetti	OpenSLAM: A platform for SLAM researchers to publish their algorithms.	http://openslam.org
Kai Arras	A FNU GPL licenced MATLAB toolbox for robot localization and mapping.	http://www.cas.kth.se/toolbox
Tim Bailey	The code is written in MATLAB and performs EFK, UKF, FastSLAM 1, and FastSLAM 2.	https://openslam.informatik.uni-freiburg.de/bailey-slam.html
Mark Paskin	Java and MATLAB hybrid programming SLAM system using thin junction tree filters.	http://ai.stanford.edu/~paskin/slam
Andrew Davison	An open source C++ library for SLAM designed and implemented.	http://www.doc.ic.ac.uk/~ajd/Scene/index.html
José Neira	A simple SLAM simulation file.	http://webdiis.unizar.es/~neira/software/slam/slamsim.htm
Dirk Hahnel	A grid-based Fast-SLAM system implemented in C.	http://dblp.uni-trier.de/pers/hd/h/H=auml=hnel:Dirk.html
Durrant Whyte, EduardoNebot,et al	MATLAB code of SLAM Summer School 2002 in Sweden.	http://www.cas.kth.se/SLAM/schedule.html

2. Position estimation

Table 9.2 Datasets for SLAM

Name	Description	Websites
Andrew Howard and Nicholas Roy	Standard data sets for robotics community, including laser and sonar data in real scenario and different maps and sensor data in simulation scenario.	http://radish.sourceforge.net
Jose Guivant，Juan Nieto and Eduardo Nebot	A large number of outdoor data sets including the famous Victoria Park Dataset.	http://www.acfr.usyd.edu.au/index.shtml
Radish（The Robotics Data Set Repository）	A large number of indoor data sets including the Claxton CS building at UTK.	http://radish.sourceforge.net
IJRR（The International Journal of Robotics Research）	Websites of research papers in IJRR.	http://www.ijrr.org

2. Position estimation

☐ SLAM-based position estimation

（1） 2D Laser-based SLAM

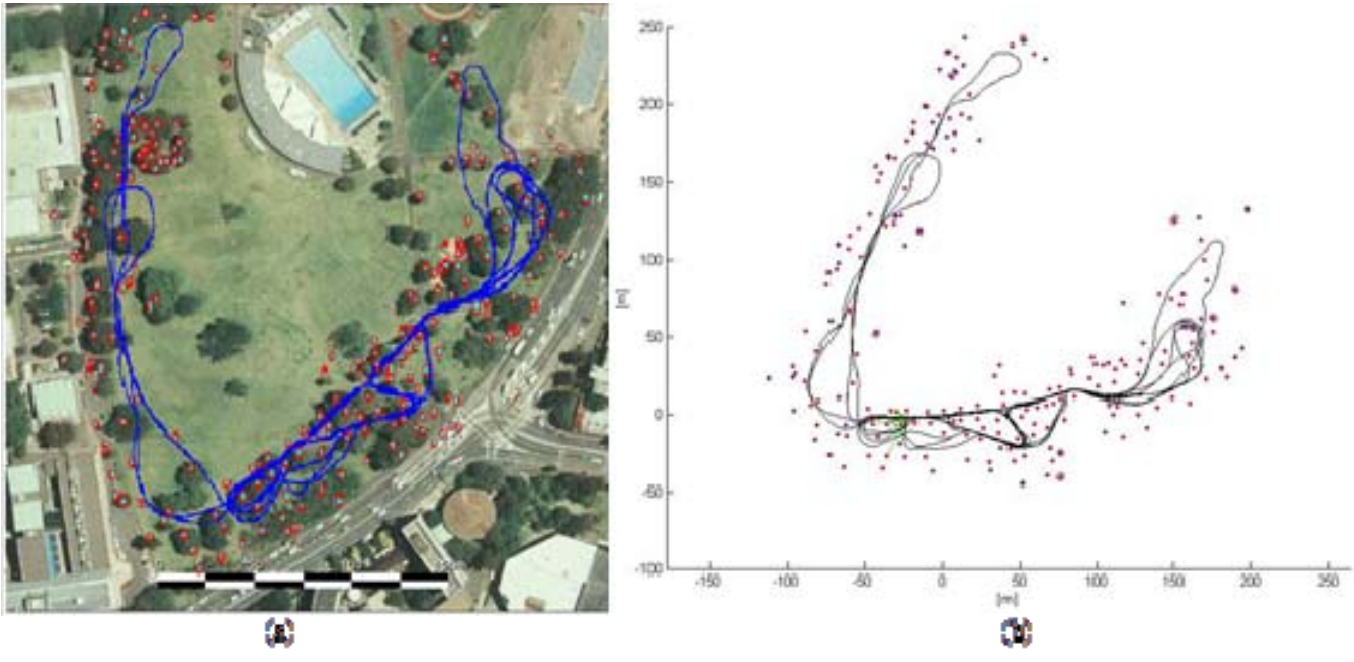


Figure 9.6. Results of laser-based SLAM. Points are landmarks and the line is the trajectory of the estimated motion.

By using the Vitoria Park data, the MATLAB code of 2D-laser SLAM algorithm is tested. Results are shown in Figure 9.6. With information of IMU and the 2D-laser, landmarks are detected and updated and then location is estimated.

2. Position estimation

☐ SLAM-based position estimation

（1） 2D Laser-based SLAM

Most algorithms mentioned in Table 9.1 are designed for robots in flat, 2D environments, which are applicable to some special conditions for multicopters. For example, a multicopter flies at a fixed altitude, especially in an environment like corridors where the horizontal cross-section of the space at different altitudes can be regarded as the same. Then 2D-SLAM algorithms are applicable.

In cases that a multicopter flies in a complex 3D environment, 3D SLAM algorithms are required. A 3D-Laser or a LiDAR can be used to acquire data. Through extending 2D SLAM algorithms, 3D SLAM results can be obtained.

2. Position estimation

☐ SLAM-based position estimation

（2）Monocular-vision-based SLAM

Vision-based SLAM (VSLAM) consists of two core steps: tracking and mapping. During the tracking step, position and attitude are estimated from scene structure information, while during the mapping step, the 3D map of the scene are built on the basis of position and attitude of the camera.

- Frame-by-Frame SLAM

Tracking and mapping are alternately performed. Tracking relies on the scene structure information from the 3D map and mapping in turn needs the motion information provided by tracking.

- Key-Frame SLAM

Tracking and mapping can be split into two separate tasks, processed in parallel threads on a dual-core computer at different rates. Tracking works in a higher rate in order to ensure real-time performance, while mapping works in a lower rate in order to get a high accuracy scene structure.

2. Position estimation

☐ SLAM-based position estimation

（2） Monocular-vision-based SLAM

A well-performed system called Parallel Tracking And Mapping (PTAM) with monocular system is widely used in micro-UAVs. In addition, a monocular vision system cannot obtain the absolute scale information. To recover the real motion of the camera, the information from IMU and altitude sensors are required. In [7], a Kalman filter was adopted to recover the absolute scale. The process model is

$$\begin{aligned} \dot{p}_{z_e} &= v_{z_e} \\ \dot{v}_{z_e} &= a_{z_e\mathrm{m}} + n_{a_{z_e}} + g \\ \dot{\lambda} &= n_{\lambda} \\ \dot{b}_{d_{\mathrm{baro}}} &= n_{b_{d_{\mathrm{baro}}}} \end{aligned}$$

$p_{z_e}, v_{z_e}, \lambda, b_{d_{\mathrm{baro}}}$ are the altitude, velocity in altitude direction, scaling factor and bias of the barometer, respectively, $n_{a_{z_e}}, n_{\lambda}, n_{b_{d_{\mathrm{baro}}}}$ are the corresponding noises.

[7] Achtelik M, Achtelik M, Weiss S, et al. Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments[C]//Robotics and automation (ICRA), 2011 IEEE international conference on. IEEE, 2011: 3056-3063.

2. Position estimation

☐ SLAM-based position estimation

（2） Monocular-vision-based SLAM

A well-performed system called Parallel Tracking And Mapping (PTAM) with monocular system is widely used in micro-UAVs. In addition, a monocular vision syste cannot obtain the absolute scale information. To recover the real motion of the camera, the information from IMU and altitude sensors are required. In [7], a Kalman filter was adopted to recover the absolute scale. The measurement model is

Observability?

$$p_{z\mathrm{SLAM}} = \lambda \cdot p_{z_e} + n_{p_{z\mathrm{SLAM}}} \longleftarrow \text{The altitude by SLAM}$$

$$d_{\mathrm{baro}} = -p_{z_e} + b_{d_{\mathrm{baro}}} + n_{d_{\mathrm{baro}}} \longleftarrow \text{The altitude by barometer}$$

[7] Achtelik M, Achtelik M, Weiss S, et al. Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments[C]//Robotics and automation (ICRA), 2011 IEEE international conference on. IEEE, 2011: 3056-3063.

3. Velocity estimation

☐ Optical-flow-based velocity estimation

（1）Optical flow

Optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between a camera and the scene.

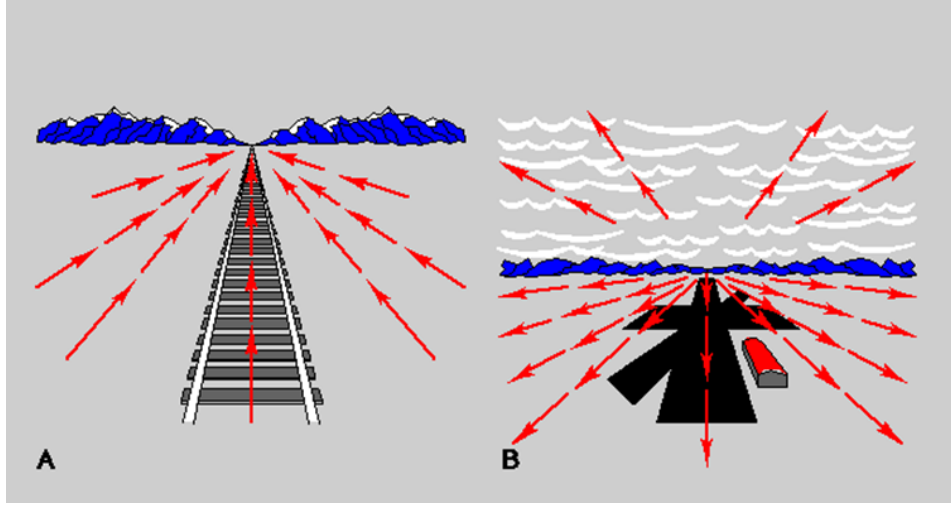


Figure 9.7. Sketch map of optical flow. (photo from the book *The Ecological Approach to Visual Perception* by James Jerome Gibson). Figure A shows optical flow when the camera movement is away from mountains, while Figure B shows optical flow when the camera movement is towards mountains.

3. Velocity estimation

☐ Optical-flow-based velocity estimation

（1） Optical flow

Suppose that the intensity $I(x,y,t)$ denotes the intensity of image points (x,y) at time t. After an interval $\mathrm{d}t$, the image point moves to $(x+\mathrm{d}x, y+\mathrm{d}y)$.The following brightness constancy constraint is given as

$$I(x+\mathrm{d}x, y+\mathrm{d}y, t+\mathrm{d}t) = I(x,y,t)$$

By assuming the movement to be sufficiently small, the image constraint at (x,y,t) with Taylor series is developed to get

$$I(x+\mathrm{d}x, y+\mathrm{d}y, t+\mathrm{d}t) = I(x,y,t) + \frac{\partial I}{\partial x}\mathrm{d}x + \frac{\partial I}{\partial y}\mathrm{d}y + \frac{\partial I}{\partial t}\mathrm{d}t + \varepsilon$$

when $\mathrm{d}t \to 0$, get the optical flow constraint equation

$$\frac{\partial I}{\partial x}\mathrm{d}x + \frac{\partial I}{\partial y}\mathrm{d}y + \frac{\partial I}{\partial t}\mathrm{d}t = 0 \Rightarrow I_x v_x + I_y v_y + I_t = 0$$

Optical flow $v_x = \frac{\mathrm{d}x}{\mathrm{d}t}$, $v_y = \frac{\mathrm{d}y}{\mathrm{d}t}$

the partial derivatives $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $I_t = \frac{\partial I}{\partial t}$

3. Velocity estimation

☐ Optical-flow-based velocity estimation

(1) Optical flow

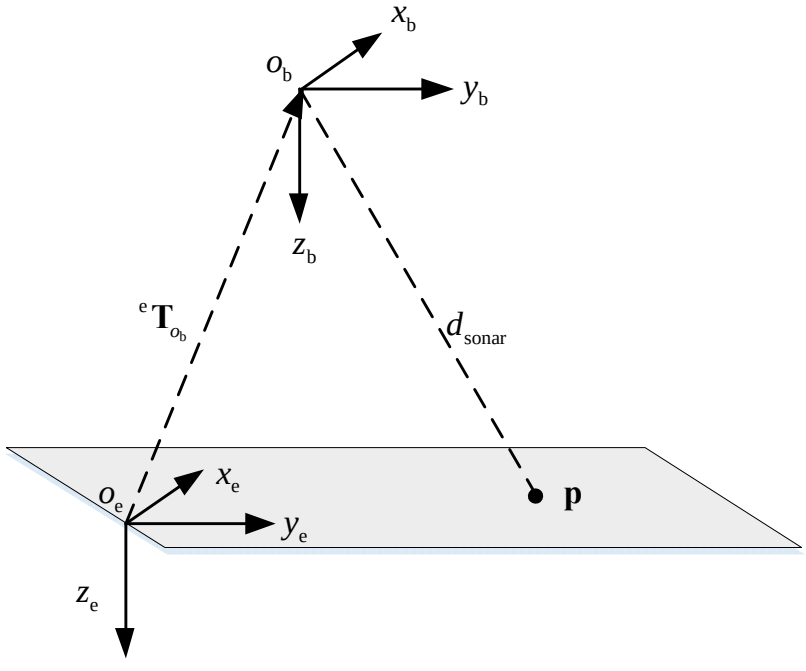
Table 9.3 Some toolboxes of optical flow

Name	Description	Websites
Computer Vision System Toolbox	This toolbox is used in MATLAB R2012a or higher version, optical flow is encapsulated to vision.Optical Flow class.	http://cn.mathworks.com/help/vision/index.html
OpenCV	Open source libraries of computer vision, lots of APIs are provided.	http://opencv.org
Machine Vision Toolbox	Vision toolbox emphasized on computer vision and 3D vision.	http://www.petercorke.com/Machine_Vision_Toolbox.html
VLFeat	Computer vision / image processing open source project,written in C and MATLAB with a large number of computer vision algorithms.	http://www.vlfeat.org/download.html
Peter Kovesi 's Toolbox	Consists of lots of computer vision algorithms written in MATLAB, support Octave	http://www.peterkovesi.com/matlabfns

3. Velocity estimation

☐ Optical-flow-based velocity estimation

（2） Optical-flow-based velocity estimation



Coordinate frames are built as shown in Figure 9.8. The monocular camera is attached to the multicopter in a downward-looking direction. For simplicity, the camera coordinate frame is aligned with the ABCF, denoted as $o_b x_b y_b z_b$, and the ground is a plane, denoted as $p_{z_e} = 0$.

Figure 9.8. A point p in ABCF $o_b x_b y_b z_b$ and EFCF $o_e x_e y_e z_e$, where d_{sonar} denotes the distance between camera center o_b and the image point $\mathbf{p}$, and $x_e o_e y_e$ is on the plane where $\mathbf{p}$ is.

3. Velocity estimation

☐ Optical-flow-based velocity estimation

（2）Optical-flow-based velocity estimation

The normalized image coordinate

$$\bar{\mathbf{p}} = \begin{bmatrix} \bar{p}_x \\ \bar{p}_y \end{bmatrix} = \begin{bmatrix} p_{x_b} / p_{z_b} \\ p_{y_b} / p_{z_b} \end{bmatrix}$$

Optical flow

$$\begin{cases} \dot{\bar{p}}_x = \dfrac{\dot{p}_{x_b} - \bar{p}_x \dot{p}_{z_b}}{p_{z_b}} \\ \dot{\bar{p}}_y = \dfrac{\dot{p}_{y_b} - \bar{p}_y \dot{p}_{z_b}}{p_{z_b}} \end{cases}$$

Substitute

The coordinate of the ground point p satisfies

$$\dot{\mathbf{R}} = \mathbf{R}\left[{}^{b}\boldsymbol{\omega}\right]_{\times}$$

$${}^{b}\mathbf{v} = \mathbf{R}^{T} \cdot {}^{e}\mathbf{v}$$

$${}^{e}\mathbf{p} = \mathbf{R} \cdot {}^{b}\mathbf{p} + {}^{e}\mathbf{T}_{o_b}$$

The ground point p stays still

$$0 \equiv {}^{e}\dot{\mathbf{p}} = \dot{\mathbf{R}} \cdot {}^{b}\mathbf{p} + \mathbf{R} \cdot {}^{b}\dot{\mathbf{p}} + {}^{e}\mathbf{v}$$

$${}^{b}\dot{\mathbf{p}} = -{}^{b}\mathbf{v} - \left[{}^{b}\boldsymbol{\omega}\right]_{\times} {}^{b}\mathbf{p}$$

$$\begin{cases} \dot{p}_{x_b} = -v_{x_b} - \omega_{y_b} p_{z_b} + \omega_{z_b} p_{y_b} \\ \dot{p}_{y_b} = -v_{y_b} - \omega_{z_b} p_{x_b} + \omega_{x_b} p_{z_b} \\ \dot{p}_{z_b} = -v_{z_b} - \omega_{x_b} p_{y_b} + \omega_{y_b} p_{x_b} \end{cases}$$

3. Velocity estimation

☐ Optical-flow-based velocity estimation

（2）Optical-flow-based velocity estimation

$$\underbrace{\begin{bmatrix}\dot{\bar{p}}_x\\ \dot{\bar{p}}_y\end{bmatrix}}_{\dot{\bar{\mathbf{p}}}}=\underbrace{\frac{1}{p_{z_b}}\begin{bmatrix}-1 & 0 & \bar{p}_x\\ 0 & -1 & \bar{p}_y\end{bmatrix}}_{\mathbf{A}(\bar{\mathbf{p}})}{}^{b}\mathbf{v}+\underbrace{\begin{bmatrix}\bar{p}_x\bar{p}_y & -(1+\bar{p}_x{}^2) & \bar{p}_y\\ (1+\bar{p}_y{}^2) & -\bar{p}_x\bar{p}_y & -\bar{p}_x\end{bmatrix}}_{\mathbf{B}(\bar{\mathbf{p}})}{}^{b}\boldsymbol{\omega}$$

For an image point $\bar{\mathbf{p}}$, its optical flow vector is $\dot{\bar{\mathbf{p}}}$ which can be calculated by algorithms shown in Table 9.3. The angular velocity ${}^{b}\boldsymbol{\omega}$ can be obtained directly from a three-axis gyroscope. p_{z_b} can be get by the reading of the sonar.

$$\begin{cases}{}^{e}\mathbf{p}=\mathbf{R}\,{}^{b}\mathbf{p}+{}^{e}\mathbf{T}_{o_b}\\ \mathbf{e}_3^{\mathrm{T}}\,{}^{e}\mathbf{T}_{o_b}=d_{\text{sonar}}\cos\theta\cos\phi\\ p_{z_e}=\mathbf{e}_3^{\mathrm{T}}\,{}^{e}\mathbf{p}=0\end{cases}\Rightarrow\begin{array}{l}\mathbf{e}_3^{\mathrm{T}}\mathbf{R}\,{}^{b}\mathbf{p}=-d_{\text{sonar}}\cos\theta\cos\phi\\ \Rightarrow p_{z_b}\mathbf{e}_3^{\mathrm{T}}\mathbf{R}\begin{bmatrix}\bar{p}_x\\ \bar{p}_y\\ 1\end{bmatrix}=-d_{\text{sonar}}\cos\theta\cos\phi\end{array}\Rightarrow p_{z_b}=-\frac{d_{\text{sonar}}\cos\theta\cos\phi}{\mathbf{e}_3^{\mathrm{T}}\mathbf{R}\left[\bar{p}_x,\bar{p}_y,1\right]^{\mathrm{T}}}$$

3. Velocity estimation

☐ Optical-flow-based velocity estimation

（2）Optical-flow-based velocity estimation

Suppose that M point pairs are detected, then

$$\underbrace{\begin{bmatrix}\dot{\bar{\mathbf{p}}}_1\\ \vdots\\ \dot{\bar{\mathbf{p}}}_M\end{bmatrix}}_{\bar{\mathbf{p}}_a}=\underbrace{\begin{bmatrix}\mathbf{A}(\bar{\mathbf{p}}_1)\\ \vdots\\ \mathbf{A}(\bar{\mathbf{p}}_M)\end{bmatrix}}_{\mathbf{A}_a}\cdot{}^{\mathrm{b}}\mathbf{v}+\underbrace{\begin{bmatrix}\mathbf{B}(\bar{\mathbf{p}}_1)\\ \vdots\\ \mathbf{B}(\bar{\mathbf{p}}_M)\end{bmatrix}}_{\mathbf{B}_a}\cdot{}^{\mathrm{b}}\boldsymbol{\omega}$$

The estimation of the velocity ${}^{\mathrm{b}}\mathbf{v}$ can be obtained by

$${}^{\mathrm{b}}\hat{\mathbf{v}}=\left(\mathbf{A}_a^{\mathrm{T}}\mathbf{A}_a\right)^{-1}\cdot\mathbf{A}_a^{\mathrm{T}}\cdot\left[\bar{\mathbf{p}}_a-\mathbf{B}_a\left({}^{\mathrm{b}}\boldsymbol{\omega}_{\mathrm{m}}-\hat{\mathbf{b}}_g\right)\right]$$

where, ${}^{\mathrm{b}}\boldsymbol{\omega}_{\mathrm{m}}-\hat{\mathbf{b}}_g$ is the de-biased angular velocity.

The following problems are still left to be solved: （1）Since the sampling periods of the camera, height sensor and gyroscope are normally different, time synchronization algorithms are required. （2） Lens distortion should be handled. （3）Rugged ground and moving background are required to be considered. （4）The mismatching of image pairs should be handled.

3. Velocity estimation

☐ Aerodynamic-model-based velocity estimation

According to model in Lesson 6, consider the drag force caused by the blade flapping, the aerodynamic model of multicopters is simplified as

$$\dot{v}_{x_\mathrm{b}} = -g\sin\theta - \frac{k_\mathrm{drag}}{m}v_{x_\mathrm{b}} + n_{a_x}$$

$$\dot{v}_{y_\mathrm{b}} = g\cos\theta\sin\phi - \frac{k_\mathrm{drag}}{m}v_{y_\mathrm{b}} + n_{a_y}$$

where n_{a_x}, n_{a_y} are the corresponding noises and $\dot{v}_{x_\mathrm{b}}, \dot{v}_{y_\mathrm{b}}$ are the accelerations of the multicopter in the ABCF.

3. Velocity estimation

☐ Aerodynamic-model-based velocity estimation

Process model:

Relationship between the attitude rates and the angular velocity

$$\dot{\phi} = \left(\omega_{x_b m} - b_{g_x} - n_{g_x}\right) + \left(\omega_{y_b m} - b_{g_y} - n_{g_y}\right)\tan\theta\sin\phi + \left(\omega_{z_b m} - b_{g_z} - n_{g_z}\right)\tan\theta\cos\phi$$

$$\dot{\theta} = \left(\omega_{y_b m} - b_{g_z} - n_{g_z}\right)\cos\phi - \left(\omega_{z_b m} - b_{g_z} - n_{g_z}\right)\sin\phi$$

$$\dot{b}_{g_x} = n_{b_{g_x}}$$

$$\dot{b}_{g_y} = n_{b_{g_y}}$$

$$\dot{b}_{g_z} = n_{b_{g_z}}$$

$$\dot{v}_{x_b} = -g\sin\theta - \frac{k_{\text{drag}}}{m} v_{x_b} + n_{a_x}$$

$$\dot{v}_{y_b} = g\cos\theta\sin\phi - \frac{k_{\text{drag}}}{m} v_{y_b} + n_{a_y}$$

The angular Velocity replaced by:

$${}^{b}\boldsymbol{\omega} = {}^{b}\boldsymbol{\omega}_{m} - \mathbf{b}_g - \mathbf{n}_g$$

$$\dot{\mathbf{b}}_g = \mathbf{n}_{b_g}$$

where

$$\mathbf{b}_g = \begin{bmatrix} b_{g_x} & b_{g_y} & b_{g_z} \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{n}_g = \begin{bmatrix} n_{g_x} & n_{g_y} & n_{g_z} \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{n}_{b_g} = \begin{bmatrix} n_{b_{g_x}} & n_{b_{g_y}} & n_{b_{g_z}} \end{bmatrix}^{\mathrm{T}}$$

Measurement model:

$$\begin{bmatrix} a_{x_b m} \\ a_{y_b m} \end{bmatrix} = \begin{bmatrix} -\frac{k_{\text{drag}}}{m} v_{x_b} + n_{a_x m} \\ -\frac{k_{\text{drag}}}{m} v_{y_b} + n_{a_y m} \end{bmatrix}$$

The measured specific force by accelerometers

Nonlinear model

Extended Kalman filter

[8]. Abeywardena D, Kodagoda S, Dissanayake G, et al. Improved State Estimation in Quadrotor MAVs: A Novel Drift-Free Velocity Estimator[J]. Robotics & Automation Magazine, IEEE, 2013, 20(4): 32-39.

3. Velocity estimation

☐ Aerodynamic-model-based velocity estimation

Suppose small attitude, $\theta \approx \phi \approx 0, \psi \approx \psi_{\mathrm{d}}, \dot{\theta} \approx \dot{\phi} \approx \dot{\psi} \approx 0$,a simplified linear version is as follows [9]

- Process model:

$$\dot{\theta} = \omega_{x_{\mathrm{b}}\mathrm{m}} - b_{g_x} - n_{g_x}$$

$$\dot{\phi} = \omega_{y_{\mathrm{b}}\mathrm{m}} - b_{g_y} - n_{g_y}$$

$$\dot{b}_{g_x} = n_{b_{g_x}}$$

$$\dot{b}_{g_y} = n_{b_{g_y}}$$

$$\dot{v}_{x_{\mathrm{b}}} = -g\theta - \frac{k_{\mathrm{drag}}}{m} v_{x_{\mathrm{b}}} + n_{a_x}$$

$$\dot{v}_{y_{\mathrm{b}}} = g\phi - \frac{k_{\mathrm{drag}}}{m} v_{y_{\mathrm{b}}} + n_{a_y}$$

Measurement model:

$$\begin{bmatrix} a_{x_{\mathrm{b}}\mathrm{m}} \\ a_{y_{\mathrm{b}}\mathrm{m}} \end{bmatrix} = \begin{bmatrix} -\frac{k_{\mathrm{drag}}}{m} v_{x_{\mathrm{b}}} + n_{a_x\mathrm{m}} \\ -\frac{k_{\mathrm{drag}}}{m} v_{y_{\mathrm{b}}} + n_{a_y\mathrm{m}} \end{bmatrix}$$

Linear model

Classical Kalman filter

[9]. Leishman R C, Macdonald J C, Beard R W, et al. Quadrotors and accelerometers: State estimation with an improved dynamic model[J]. Control Systems, IEEE, 2014, 34(1): 28-41.

4. Obstacle estimation

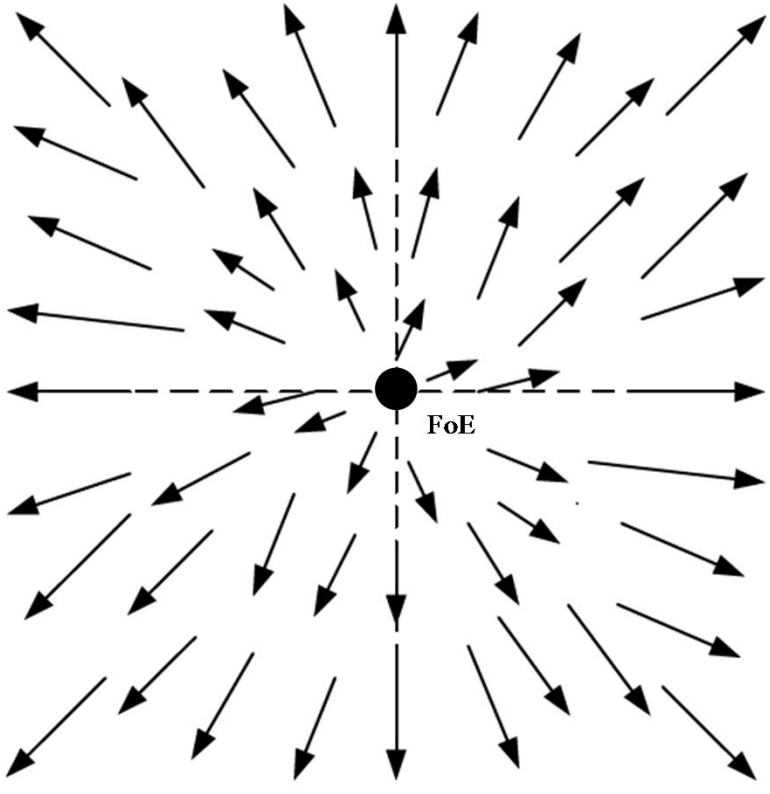


Figure 9.9. FoE point

In this section, an obstacle avoidance method based on optical flow is introduced. This method firstly recovers Time to Contact/Collision (TTC) from optical flow information and then guides multicopters to avoid obstacles. Generally, TTC is calculated from Focus of Expansion (FoE).

4. Obstacle estimation

FOE calculation

Optical flow and the angular velocity satisfies

$$\dot{\bar{\mathbf{p}}} = \mathbf{A}(\bar{\mathbf{p}}) \cdot {}^{\mathrm{b}}\mathbf{v} + \mathbf{B}(\bar{\mathbf{p}}) \cdot {}^{\mathrm{b}}\boldsymbol{\omega}$$

$$\Delta\bar{\mathbf{p}} = \dot{\bar{\mathbf{p}}} - \mathbf{B}(\bar{\mathbf{p}}) \cdot {}^{\mathrm{b}}\boldsymbol{\omega} \Rightarrow \Delta\bar{\mathbf{p}} = \mathbf{A}(\bar{\mathbf{p}})\,{}^{\mathrm{b}}\mathbf{v}$$

Let $\Delta\bar{\mathbf{p}} = \begin{bmatrix} \Delta\bar{p}_x & \Delta\bar{p}_y \end{bmatrix}^{\mathrm{T}}$ and rearrange

$$\frac{\Delta\bar{p}_x}{\Delta\bar{p}_y} = \frac{\bar{p}_x - c_x}{\bar{p}_y - c_y}$$

where $c_x = v_{x_\mathrm{b}} / v_{z_\mathrm{b}}, c_y = v_{y_\mathrm{b}} / v_{z_\mathrm{b}}$. Since optical flow and angular velocity are often subject to noise, more point pairs are required to calculate the FoE. Let $p_i, i = 1, \ldots, N$ be the chosen points. Then

$$\underbrace{\begin{bmatrix} \Delta\bar{p}_{y,1} & -\Delta\bar{p}_{x,1} \\ \Delta\bar{p}_{y,2} & -\Delta\bar{p}_{x,2} \\ \vdots & \vdots \\ \Delta\bar{p}_{y,N} & -\Delta\bar{p}_{x,N} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \underbrace{\begin{bmatrix} \bar{p}_{x,1}\Delta\bar{p}_{y,1} - \bar{p}_{y,1}\Delta\bar{p}_{x,1} \\ \bar{p}_{x,2}\Delta\bar{p}_{y,2} - \bar{p}_{y,2}\Delta\bar{p}_{x,2} \\ \vdots \\ \bar{p}_{x,N}\Delta\bar{p}_{y,N} - \bar{p}_{y,N}\Delta\bar{p}_{x,N} \end{bmatrix}}_{\mathbf{b}} \Rightarrow \begin{bmatrix} \hat{c}_x \\ \hat{c}_y \end{bmatrix} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\left(\mathbf{A}^{\mathrm{T}}\mathbf{b}\right)$$

Least square solution

4. Obstacle estimation

☐ TTC calculation

Generally, the absolute depth cannot be recovered from the image sequence captured by a monocular camera. But the collision time TTC can be estimated from such an image sequence. The definition of TTC is

$$t_{\mathrm{TTC}} = \frac{p_{z_\mathrm{b}}}{v_{z_\mathrm{b}}}$$

On the other hand,

$$\Delta\bar{\mathbf{p}} = \mathbf{A}\left(\bar{\mathbf{p}}\right)^{\mathrm{b}}\mathbf{v}$$

$$\Delta\bar{p}_x = \frac{1}{p_{z_\mathrm{b}}}\left(-v_{x_\mathrm{b}} + \bar{p}_x v_{z_\mathrm{b}}\right)$$

$$\Delta\bar{p}_y = \frac{1}{p_{z_\mathrm{b}}}\left(-v_{y_\mathrm{b}} + \bar{p}_y v_{z_\mathrm{b}}\right)$$

$$\Delta\bar{p}_x t_{\mathrm{TTC}} = \bar{p}_x - c_x$$

$$\Delta\bar{p}_y t_{\mathrm{TTC}} = \bar{p}_y - c_y$$

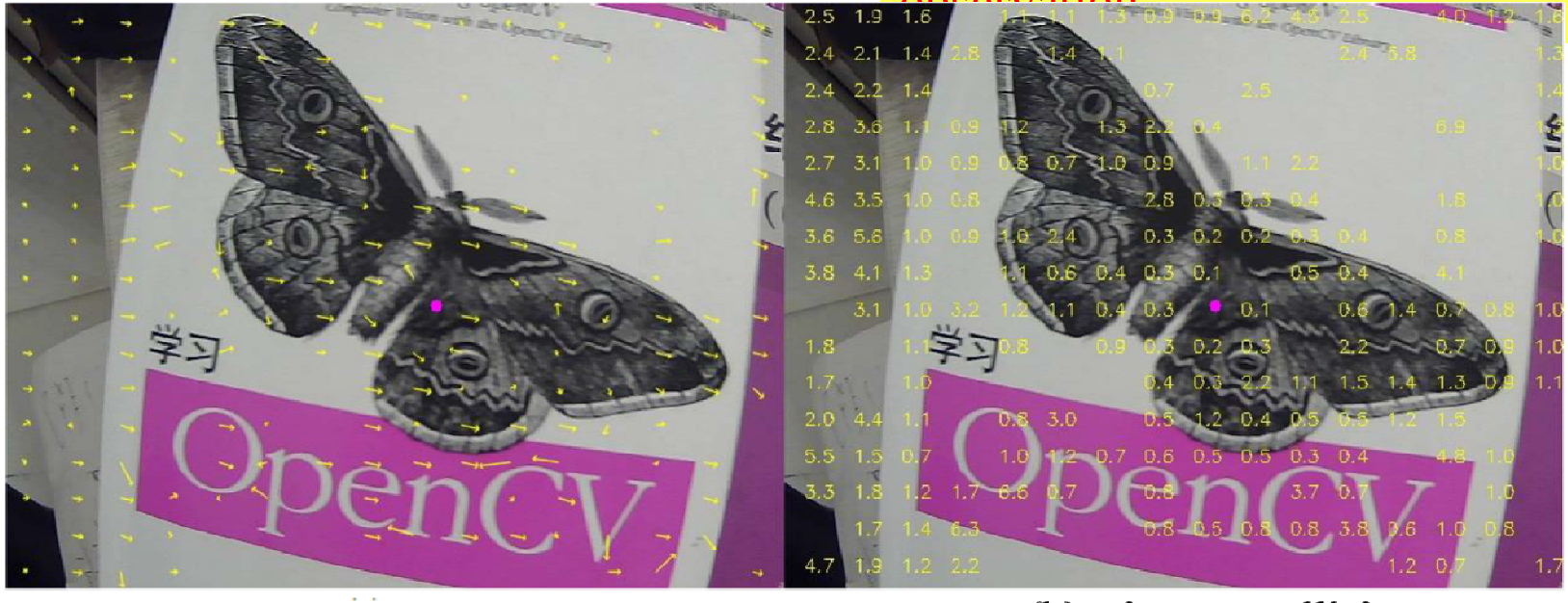
Least square solution

$$t_{\mathrm{TTC}} = \sqrt{\frac{\left(\bar{p}_x - \hat{c}_x\right)^2 + \left(\bar{p}_y - \hat{c}_y\right)^2}{\Delta\bar{p}_x^2 + \Delta\bar{p}_y^2}}$$

4. Obstacle estimation

☐ TTC calculation

As shown, an optical flow vector with a large length corresponds to a small collision time. This is consistent with the observation.



(a) Optical flow (b) Time to collision

Figure 9.10. Optical flow and collision time

5. Conclusion

1. State estimation is the basis of control and decision-making. Actually, it is much more complicated and therefore deserves further research.
2. At present, the studies on multicopters are focused on the state estimation and environment sensing, especially based on vision, where SLAM and the optical flow methods mentioned above are the main research areas.
3. In a practical state estimation system, more problems need to be considered [10]:

（1） Computation performance. Since the computing ability of processors and the allowable calculation time are both limited, computing resources should be used efficiently.

（2） Abnormal data. In APM, about 90% of the code aims to detect and deal with abnormal data. (Corner Case)

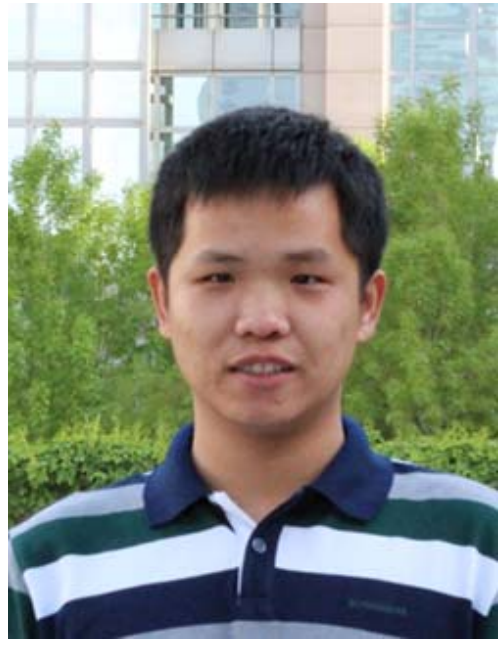
（3） Measurement latency. In a real system, the phenomenon naturally exists that sensors are with different sampling rates and suffer measurement delay.

[10]. Paul Riseborough. Application of Data Fusion to Aerial Robotics. March 24, 2015, available online at http://thirty5tech.com/vid/watch/Z3Qpi1Rx6HM.

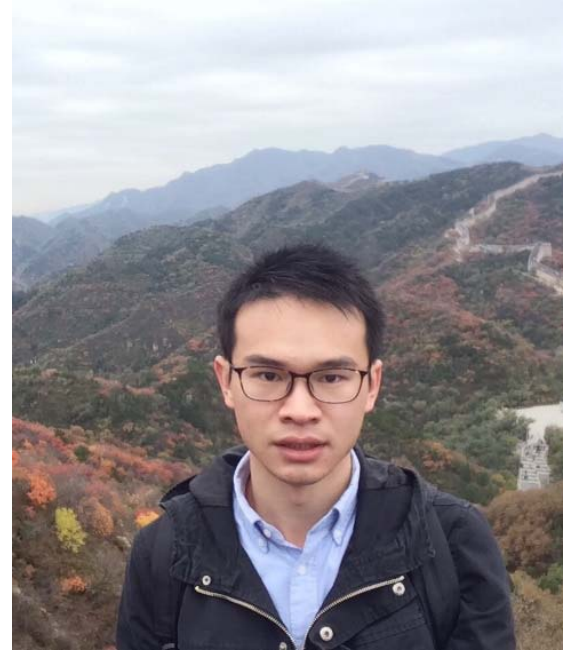
Acknowledgement

Deep thanks go to

Heng Deng



Hongxin Dong



Xunhua Dai

for material preparation.

Thank you!

All course PPTs and resources can be downloaded at
http://rfly.buaa.edu.cn/course

For more detailed content, please refer to the textbook:

Quan, Quan. Introduction to Multicopter Design and Control. Springer, 2017. ISBN: 978-981-10-3382-7.

It is available now, please visit http://www.springer.com/us/book/9789811033810