



*Open-Source  
Projects on  
Unmanned  
Aerial Vehicles*

# Build Your Own Quadrotor

**T**his article presents a survey on publicly available open-source projects (OSPs) on quadrotor unmanned aerial vehicles (UAVs).

Recently, there has been increasing interest in quadrotor UAVs. Exciting videos have been published on the Internet by many research groups and have attracted much attention from the public [1]–[7]. Relatively simple structures of quadrotors has promoted interest from academia, UAV industries, and radio-control (RC) hobbyists alike. Unlike conventional helicopters, swashplates, which are prone to failure without constant maintenance, are not required. Furthermore, the diameter of individual rotors can be reduced as a result of the presence of four actuators [8].

Many research groups or institutions have constructed their own quadrotors to suit specific purposes. Successes have been reported from academia, such as the X4-flyer [9], OS4 [10], STARMAC [11], and Pixhawk [12] to mention a few. To the commercial market, the Draganflyer X4, Asctec Hummingbird, Gaiu Quad flyer, Parrot ARDrone, and DJI Wookong have been introduced. At the same time, a number of OSPs for quadrotors have emerged as shown in Figure 1, with contributions from RC hobbyists, universities [12], [13], and corporations.

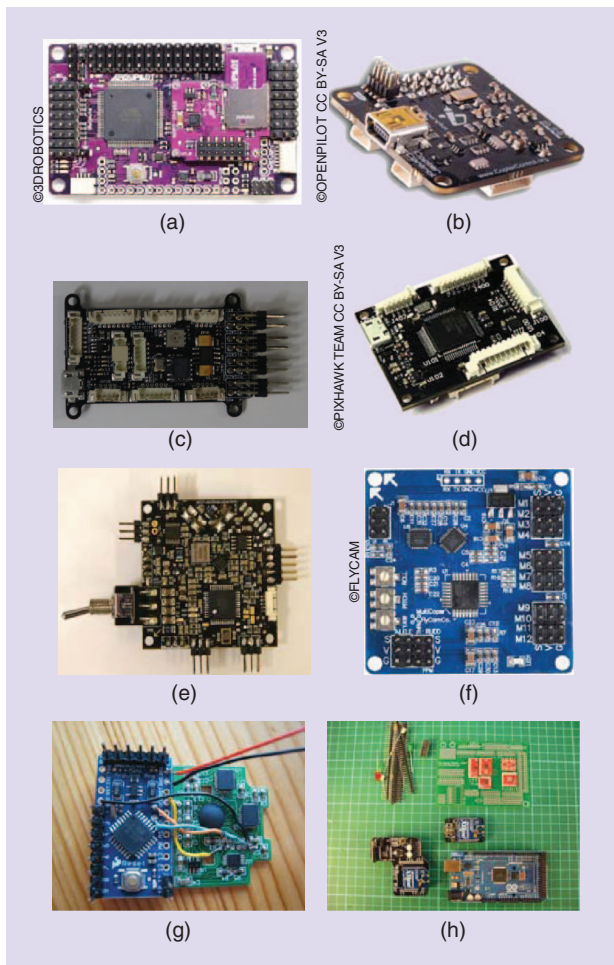
---

By Hyon Lim, Jaemann Park,  
Daewon Lee, and H.J. Kim

---

Digital Object Identifier 10.1109/MRA.2012.2205629

Date of publication: 10 September 2012



**Figure 1.** Open-source quadrotor autopilots. (a) Arducopter, (b) Openpilot, (c) Paparazzi, (d) Pixhawk, (e) Mikrokopter (photo courtesy of Inkyu-Sa), (f) KKmulticopter, (g) Multiwii (photo courtesy of Alexandre Dubus), and (h) Aeroquad (photo courtesy of Ian Johnston).

Quadrotor OSPs use community-hosting sites (e.g., Google code and Github) to create code, blueprints, or schematics, which are freely available under open-source licenses such as the general public license (GPL) [14]. These tools help talented independent developers to join

**Table 1. OSPs on quadrotor autopilot.**

Project Name	Web site URL
Arducopter	<a href="http://code.google.com/p/arducopter">http://code.google.com/p/arducopter</a>
Openpilot	<a href="http://www.openpilot.org">http://www.openpilot.org</a>
Paparazzi	<a href="http://paparazzi.enac.fr">http://paparazzi.enac.fr</a>
Pixhawk	<a href="http://pixhawk.ethz.ch">http://pixhawk.ethz.ch</a>
Mikrokopter	<a href="http://www.mikrokopter.de">http://www.mikrokopter.de</a>
KKmulticopter	<a href="http://www.kkmulticopter.com">http://www.kkmulticopter.com</a>
Multiwii	<a href="http://www.multiwii.com">http://www.multiwii.com</a>
Aeroquad	<a href="http://www.aeroquad.com">http://www.aeroquad.com</a>

OSP freely from all over the world. This setting allows very fast development processes because new features can be tested not only by the developer but also by other people in the community. Feedback is given in real time from various conditions and configurations, which make open-source software more robust in a relatively short period of time.

OSP have been successful in many disciplines and remain competitive with commercial alternatives with Linux being the most famous operating system. In the robotics area, more than 2,000 projects have been established based on the robot operating system [15] with its well-organized framework that encourages open-source development.

In the case of quadrotor OSPs, one of the main reasons to use them is flexibility in both hardware and software, which makes modification easier to meet the specific requirements of a user. In addition, OSPs allow researchers to replicate and extend the results of others and provide a baseline for comparison among various approaches.

In this article, we introduce eight quadrotor OSPs and compare them in terms of hardware and software to provide a compact overview for use in a variety of areas as well as in academic research.

### Open-Source Projects for Quadrotor UAVs

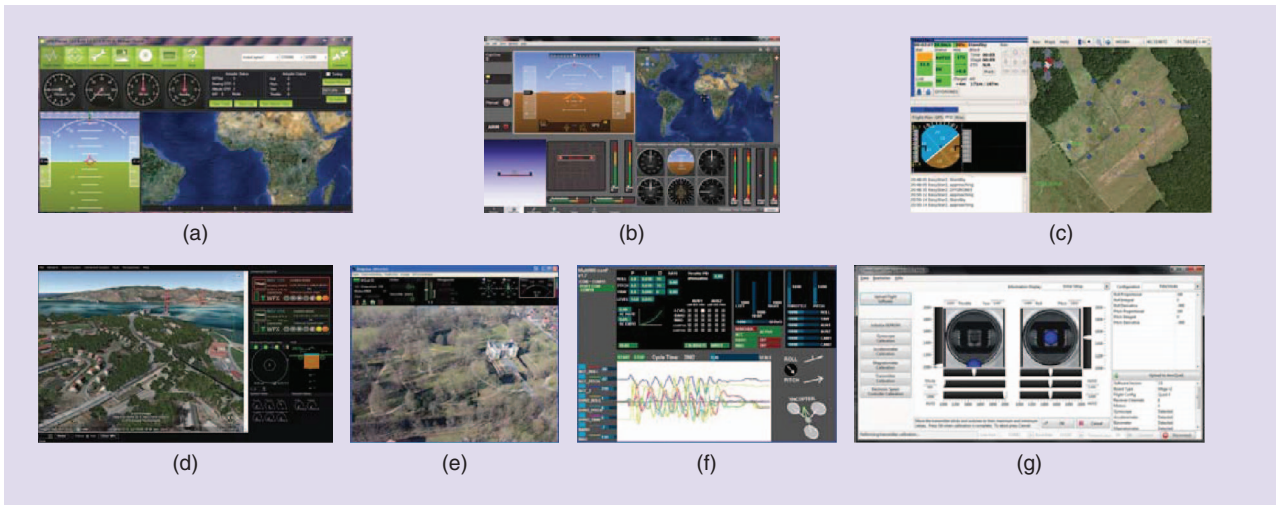
In this section, we introduce the quadrotor OSPs that are listed in Table 1. These projects have been selected based on the user volume, activity, and project maintenance status. All of these projects are still in development. Therefore, readers should note that the information described in this article is as of May 2012. We use the term OSP to refer to code, electronics, or auxiliary software such as the ground-control software (GCS), depending on the context.

#### Arducopter

Arducopter is a quadrotor autopilot project based on the Arduino framework developed by individual engineers worldwide, which is described earlier [Figure 1(a)]. A graphical-user-interface (GUI)-based software GCS is provided to tune control gains and display flight information [Figure 2(a)]. This project shares the same avionics platform with Ardupilot, which is a fixed-wing aircraft autopilot OSP. A helicopter autopilot is also supported. There are more than 30 contributors on the project Web site and it uses the GNU Lesser GPL (LGPL) [16].

#### Openpilot

Openpilot is an OSP led by RC hobbyists [Figure 1(b)] using GPL [14]. This project features a real-time operating system modified from FreeRTOS, which is an open-source operating system. Openpilot supports fixed-wing aircraft and helicopters with the same autopilot avionics. A GUI-based GCS is provided to tune



**Figure 2.** Screenshots of open-source GCS. (a) Arducopter, (b) Openpilot, (c) Paparazzi, (d) Pixhawk, (e) Mikrokoetter, (f) Multiwii, and (g) Aeroquad.

gains and receive flight data [Figure 2(b)]. On the Web site of this project, various videos are available describing gain tuning, assembly processes, and flight principles to help users.

### Paparazzi

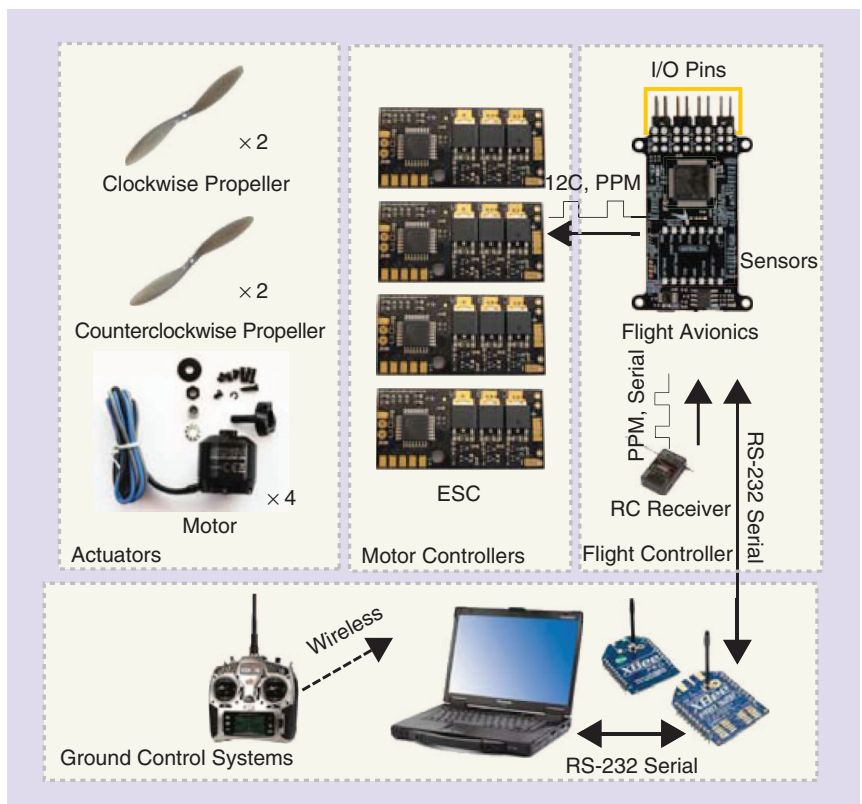
Paparazzi is an autopilot system oriented toward inexpensive autonomous aircraft of all types [Figure 1(c)]. It has been in development since 2003 [13]. Originally a fixed-wing autopilot, it now supports quadrotor configurations by modifying the control mixing rule. Nine different autopilot hardware systems are developed under the lead of the Paparazzi team at ENAC University. Paparazzi provides GUI-based GCS with flight-path scripting that makes mission planning in outdoors convenient [Figure 2(c)]. This project uses GPL for hardware and software.

### Pixhawk

Pixhawk [12] uses onboard computer-vision algorithms developed by ETHZ Computer Vision Group [Figure 1(e)]. Among the projects introduced here, only the Pixhawk project has computer vision equipment which has been used in several papers [17]–[19]. It also provides GUI-based GCS [Figure 2(d)] called *Qgroundcontrol*, which is a separate OSP in collaboration with the MAVlink protocol project. The Pixhawk project is available under the GPL.

### Mikrokoetter

Mikrokoetter is a quadrotor autopilot system developed by a subsidiary of HiSystems GmbH in 2006 [Figure 1(e)]. GUI-based software for gain tuning and health monitoring is provided as shown in Figure 2(e). Mikrokoetter is operated in well-organized Internet shops for their autopilot boards. In 2010, the University of Tasmania and the Australian Antarctic Division made use of Mikrokoetter to monitor moss beds in Antarctica.



**Figure 3.** Configuration of a typical quadrotor UAV system.

**Table 2. Main components of the OSP quadrotors.**

Description	Arducopter	Openpilot	Paparazzi (Lisa/M)	Pixhawk	Mikrokopter	KKmulticopter	Multiwii	Aeroquad
Dimension (mm)	66 × 40.5	36 × 36	51 × 25	40 × 30.2	44.6 × 50	49 × 49	N/A <sup>a</sup>	N/A <sup>a</sup>
Weight (g)	23	8.5	10.8	8	35	11.7	N/A <sup>a</sup>	N/A <sup>a</sup>
Processor	ATmega2560	STM32F103CB	STM32F105RCT6	LPC2148	ATmega644	ATmega168	Arduino <sup>b</sup>	Arduino <sup>b</sup>
Processor frequency (MHz)	16	72	60	60	20	20	8–20	8–20
Gyroscope	Motion processor unit-6000 (MPU-6000)	ISZ/IDC-500	MPU-6000	ISZ/IDC-500	ADXRS610	ENC-03	ISZ/IDC-650	ITG3200
Accelerometer	MPU-6000	ADX330	MPU-6000	SCA3100-D04	LIS344ALH	—	LIS3L02AL	ADXL345
Magnetometer	HMC5843	HMC5843	HMC5843	HMC5843	KMZ51	—	HMC5883L	HMC5883L
Barometer	MS5611	BMP085	MS5611	BMP085	MPX4115A	—	BMP085	BMP085

<sup>a</sup>Not available. Because the Multiwii and Aeroquad support dynamic hardware configuration, the size depends on configuration.

<sup>b</sup>The project is based on the Arduino board, so the actual processor varies.

The source code is available for noncommercial purposes only.

### KKmulticopter

KKmulticopter is contributed by 20 people around the world [Figure 1(f)]. This project has targeted hobbyists who want to capture aerial photographs using quadrotors. The autopilot hardware of this project is the most basic among the projects described in this article. It is equipped only with a triaxis gyroscope for inertial measurement and an 8-b microcontroller for control. No GCS is provided and gains are tuned by variable resistors on board.

### Multiwii

Multiwii is a quadrotor autopilot system developed by RC hobbyists [Figure 1(g)]. This project uses an Arduino board as a main processor while the sensor system can vary. This project aims to make the fabrication of electronics easy. It uses gyroscopes and accelerometers of the commercial off-the-shelf Wii motion controller from Nintendo, which needs less soldering. GUI-based GCS is provided as shown in Figure 2(f). GPL is used for this project.

### Aeroquad

Aeroquad is a quadrotor autopilot based on Arduino [Figure 1(h)]. Similar to Multiwii, it uses a standard Arduino board instead of making its own fully fledged single board. Aeroquad also provides GUI-based GCS software as shown in Figure 2(g). Arducopter was separated from this project in May 2010. GPL is used by Aeroquad.

### Arduino Platform

Although the Arduino platform is not a quadrotor autopilot, we introduce it here because the Arducopter, Multiwii, and Aeroquad projects all use it. Arduino is the name of both the open-source single-board microcontroller circuit and the integrated development environment (IDE). Arduino has a well-organized device driver library for different sensors and actuators. It is frequently used for rapid prototyping because of the following advantages:

- *Simple setup:* Arduino's IDE is easy to install, and firmware can be easily downloaded via USB or RS-232 without an expensive JTAG interface.
- *Rich device drivers:* There are more than 100 libraries related to hardware peripherals and signal analysis on the Arduino platform.
- *Operating systems supported:* The Arduino development IDE is ported to Mac OS X, Windows, and Linux.

### Components of Open-Source Projects for Quadrotor UAVs

Figure 3 shows overall configuration of a typical quadrotor UAV, which consists of flight avionics, sensor systems, radio transmitters, receivers, and communication systems.

**Table 3. Specifications of accelerometers.**

Chip Name	Outputs	Axis	Supply (V)	Power (mA)	Measurement Range (g)	Bandwidth (kHz)	Nonlinearity (%)	Dimension (mm)
ADXL330	Voltage	Three axes	1.8–3.6	0.32	±3	1.6 (XY), 0.55 (Z)	±0.3	4.0 × 4.0 × 1.45
SCA3100-D04	SPI	Three axes	3.0–3.6	3	±2	2	±2	7.6 × 3.3 × 8.6
LIS344ALH	Voltage	Three axes	2.4–3.6	0.68	±6	1.8	±0.5	4.0 × 4.0 × 1.5
MPU-6000	I2C	Three axes	2.375–3.46	0.5	±16	1	±0.5	4.0 × 4.0 × 0.9
LIS3LO2AL	Voltage	Three axes	3.4–3.6	0.85	±2	1.5	±0.3 (XY), ±0.5 (Z)	5.0 × 5.0 × 1.52

**Table 4. Specifications of gyroscopes.**

Chip Name	Outputs	Axis	Supply (V)	Power (mA)	Range (degree/s)	Response (Hz max.)	Dimension (mm)
ENC-03	Voltage	One axis	2.7–5.5	5	±300	50	15.5 × 8.0 × 4.3
IDG-500	Voltage	Two axes (XY)	2.7–3.3	7	±500	140	4.5 × 5.0 × 1.2
ISG-500	Voltage	One axis (Z)	2.7–3.3	4.5	±500	140	4.0 × 5.0 × 1.15
IDC-650	Voltage	Two axes (XY)	2.7–3.3	7	±2,000	140	4.0 × 5.0 × 1.15
ISG-650	Voltage	One axis (Z)	2.7–3.3	4.5	±2,000	140	4.0 × 5.0 × 1.15
ADXRS610	Voltage	One axis	4.75–5.25	3.5	±300	2,500	7.0 × 7.0 × 3.0
MPU-6000	I2C	Three axes	2.375–3.46	3.6	±2,000	256	5.0 × 5.0 × 1.52

### Flight Avionics

Flight avionics for the various projects mentioned in the section “Open-Source Projects for Quadrotor UAVs” are shown in Figure 1. Most of the introduced projects provide electronic schematics for self-production. Typically, flight avionics consists of a processor, input/output (I/O) pins, and sensors. The I/O pins connect an off-the-shelf

electronic speed controller (ESC) and RC receiver to the flight controller. The sensor suite consists of a gyroscope, accelerometer, barometer, magnetometer, and global positioning system (GPS).

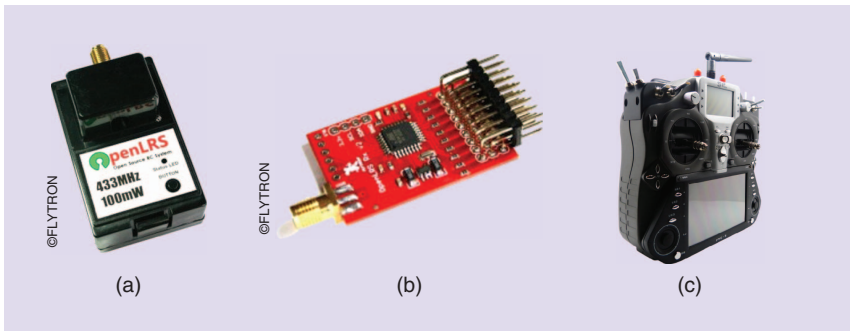
Table 2 describes flight avionics composition. Most flight avionics are full-fledged with six degrees of freedom (6DoF) inertial measurement unit (IMU), magnetometer,

**Table 5. Specifications of magnetometers.**

Chip Name	Outputs	Axis	Supply (V)	Power (mA)	Range (G)	Rate (Hz)	Dimension (mm)
HMC5843	I2C	Three axes	2.5–3.3	0.8	±4	116	4.0 × 4.0 × 1.3
HMC5883	I2C	Three axes	1.6–3.3	0.64	±8	116	3.0 × 3.0 × 0.9
KMZ51	Voltage	One axis	5.0–8.0	—	±2.5	—	5.0 × 4.0 × 1.75

**Table 6. Specifications of barometers.**

Chip Name	Interface	Supply Voltage	Power Consumption	Range	Response Time (ms)	Dimension (mm)
MPX4115A	Analog (voltage)	4.85–5.35	0.1 mA (max)	150–1150 hPa	1	11.38 × 10.54 × 12.7
BMP085	I <sup>2</sup> C	1.8–3.6	5 μA	300–1100 hPa	7.5	5.0 × 5.0 × 1.2
MS5611	I <sup>2</sup> C	1.8–3.6	0.9–12.5 μA	450–1100 mBar	0.5–8.22	3.0 × 5.0 × 1.7



**Figure 4.** Open-source RC transmitters and receivers. (a) OpenLRS transmitter, (b) openLRS receiver, and (c) OSRC transmitter.

and barometer. However, KKmulticopter has only three gyroscopes, because it is devoted to manual flight. KKmulticopter implements a stability augmented system (SAS), which will be discussed in the section “Open-Source Projects Internals.” Most flight controllers implement proportional-integral-derivative (PID) control for stabilization of the quadrotor, although the structure of the PID controllers between the projects varies slightly. This will also be discussed in detail in the section “Open-Source Projects Internals.”

### Sensors

Detailed specification of the sensors used in the OSPs is given in Tables 3–6. For the accelerometer, five different chips are used in the OSPs, which are shown in Table 3. For the gyroscope, there are seven different chips used as listed in Table 4. Magnetometers are used to correct attitude information and estimate drift of gyroscopes. There are three different magnetometers used in the OSPs as shown in Table 5. Three types of barometers used to measure altitude are shown in Table 6.

### Radio Transmitters and Receivers

Recently, some groups have modified off-the-shelf RC transmitters to fit their requirement such as complex control mixing or curve shaping of a stick. As a result, custom firmwares for a few RC transmitters have been released as open source. In addition, open-source RC transmitters and receivers have been emerging [Figure 4(a) and (b)]. The OpenLRS project was initiated for open-source RC radio transmitter and receiver development. The OSRC project has developed not only a radio part but also controller hardware as shown in Figure 4(c). These projects are useful when a flight-avionics package needs to be more compact without additional hardware such as an RC receiver.

### Communication Systems

XBee is a popular communication system because of its simple setup, low cost, and reasonable communication range when compared with its size. All the projects addressed here use Xbee.

Arducopter and Pixhawk implement the MAVLink protocol for ground control. One advantage of the MAVLink protocol is that one can use Qgroundcontrol without a need to develop separate GCS.

## Open-Source Projects Internals

### Attitude Estimation

Because a sensor suite is typically composed of a three-axis gyroscope and a three-axis accelerometer, which

provide linear accelerations and angular rates only, a proper attitude estimation algorithm should be employed.

### Extended Kalman Filter

The Openpilot and Pixhawk projects have designed an attitude estimation algorithm based on the extended Kalman filter (EKF). Here, we provide only an overview of the EKF-based attitude estimation of the Openpilot project, and the complete EKF algorithm can be found in [20].

Let  $p$  and  $v$  be three-dimensional (3-D) position and velocity in earth-fixed frame,  $q$  the quaternion, and  $b$  the gyro bias. Let  $\mathbf{R}_{eb}(q)$  and  $\mathbf{\Omega}(q)$  be rotation matrix that converts body-fixed frame to earth-fixed frame and quaternion rates matrix, respectively, as a function of the unit quaternion. Let  $a$  denotes linear acceleration in body-fixed frame and  $\omega$  the angular velocity in body-fixed frame. Then, the state equation in discrete time can be written as

$$x_k = \begin{bmatrix} p_k \\ v_k \\ q_k \\ b_k \end{bmatrix} = \begin{bmatrix} v_{k-1} \\ \mathbf{R}_{eb}(q_{k-1}) \cdot a_{k-1} \\ \frac{1}{2} \mathbf{\Omega}(q_{k-1}) \cdot \omega_{k-1} \\ w_{b,k-1} \end{bmatrix}. \quad (1)$$

In (1), the gyro bias  $b$  is modeled with noise  $w_b$ . The system input  $u$  consists of measurements of angular velocity  $\omega_m$  and linear acceleration  $a_m$ :

$$u_k = \begin{bmatrix} \omega_{m,k} \\ a_{m,k} \end{bmatrix} = \begin{bmatrix} \omega_k - w_{\omega,k} + b_k \\ a_k - w_{a,k} - \mathbf{R}_{eb}^T(q_k) [0 \ 0 \ g]^T \end{bmatrix}, \quad (2)$$

where  $w_\omega$  and  $w_a$  represent noise and  $g$  is gravitational acceleration. Substitution of (2) into (1) yields the following nonlinear model:

$$x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} = \begin{bmatrix} v_{k-1} \\ \mathbf{R}_{eb}(q_{k-1})(a_{m,k-1} + w_{a,k-1}) + [0 \ 0 \ g]^T \\ \frac{1}{2} \mathbf{\Omega}(q_{k-1})(\omega_{m,k-1} + w_{\omega,k-1} - b_{k-1}) \\ w_{b,k-1} \end{bmatrix}, \quad (3)$$

where  $w_k = [w_{\omega,k}, w_{a,k}, w_{b,k}]^T$  is process noise. The nonlinear measurement model is (we omit time index  $k$  for notational simplicity)

$$z_k = h(x_k) + v_k = \begin{bmatrix} p \\ v \\ m_b \\ h_b \end{bmatrix} = \begin{bmatrix} p \\ v \\ \mathbf{R}_{eb}^T(q)m_e \\ -P_z \end{bmatrix}, \quad (4)$$

where  $m_b$  is the measurement of the magnetic field of the earth  $m_e$  in body frame,  $h_b$  is the height measured by the barometric sensor reading  $P_z$ , and  $v_k$  is the measurement noise.

The states are estimated by the standard EKF algorithm and measurements from accelerometers, gyroscopes, magnetometers, GPS, and barometer are fused to estimate the states.

### Linear Complementary Filter

The Mikrokopter project implements the linear complementary filter (LCF) and is shown in Figure 5 on each axis of the accelerometer and gyroscope. It is designed to fuse multiple independent noisy measurements of the same signal that have complementary spectral characteristics. The details of complementary filters can be found in [21] and [22].

Let  $y_u$  be the rate measurement of the angle  $\theta$  and  $y_x$  the angle measured by accelerometer. The complementary filter to estimate the angle  $\theta$  is given by

$$\dot{\hat{\theta}} = y_u + k_p(y_x - \hat{\theta}), \quad (5)$$

where  $\hat{\theta}$  denotes the estimate of  $\theta$  and  $k_p$  is a gain that determines crossover frequency. The complementary filter described in (5) assumes that there is no steady-state estimation error. However, in practice, the gyro bias varies over time. To compensate for this, an integrator [Figure 5(b)] is added to obtain the following:

$$\dot{\hat{\theta}} = y_u - \hat{b} + k_p(y_x - \hat{\theta}), \quad (6)$$

$$\dot{\hat{b}} = -k_I(y_x - \hat{\theta}). \quad (7)$$

### Nonlinear Complementary Filters on the SO(3) Group

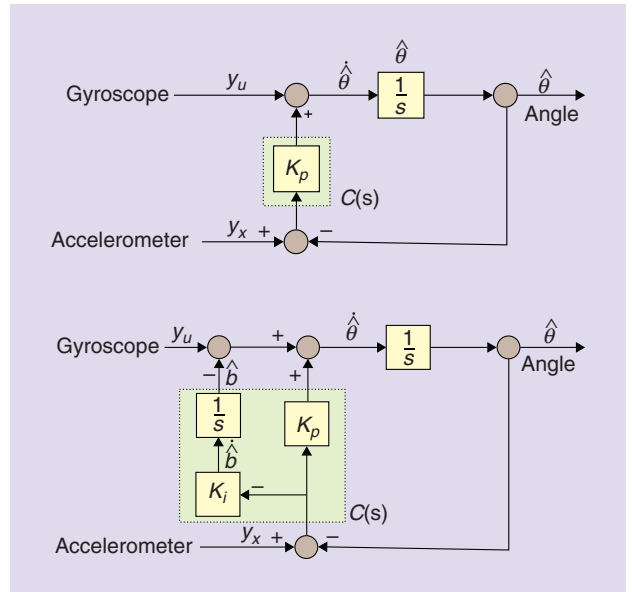
An LCF is extended to the nonlinear SO(3) group [22] (Figure 6). The final form of the filter with bias estimate is given by

$$\dot{\hat{\mathbf{R}}} = \hat{\mathbf{R}}(\Omega_y - \hat{b} + \lambda)_\times \quad (8)$$

$$\dot{\hat{b}} = -k_I \lambda, \quad \hat{b}(0) = \hat{b}_0 \quad (9)$$

$$\lambda = \text{vex}(\pi_a(\tilde{\mathbf{R}})), \quad \tilde{\mathbf{R}} = \hat{\mathbf{R}}^T \mathbf{R}_y, \quad (10)$$

where  $\pi_a(\tilde{\mathbf{R}}) = 1/2(\tilde{\mathbf{R}} - \tilde{\mathbf{R}}^T)$  and  $\hat{\mathbf{R}}, \tilde{\mathbf{R}} \in \text{SO}(3)$  are attitude estimate and estimate error, respectively, and the vex operator is the inverse operation of a skew-symmetric matrix.



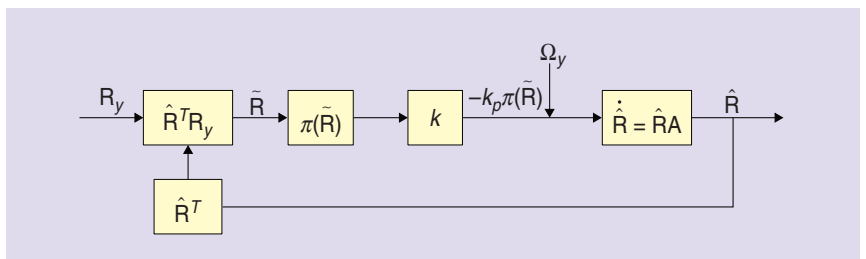
**Figure 5.** Complementary filter blends two different sensors that have different frequency responses [21]. (a) Complementary filter without bias compensation and (b) complementary filter with bias compensation.

$\mathbf{R}_y$  is the rotation matrix reconstructed using roll and pitch measured from the accelerometer.  $\Omega_y$  is the measurement from the three-axis gyroscope. Because  $\mathbf{R}$  has to satisfy the constraint  $\mathbf{R}^T \mathbf{R} = \mathbf{I}$ , the computation load becomes an issue in implementing this on an embedded system. For this, in [22], the filter based on quaternion is provided. Arducopter, Multiwii, and Aeroquad have implemented this algorithm with the rotation matrix representation, and the Paparazzi project provides both rotation matrix and quaternion representations.

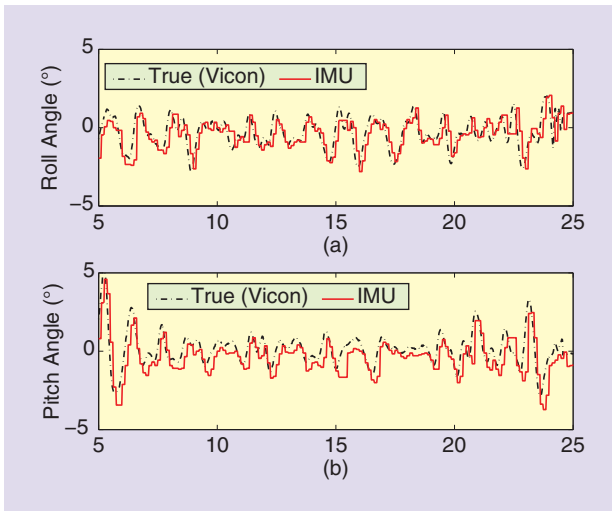
We evaluated the nonlinear complementary filters (NCFs) on the SO(3) group using a Vicon motion capture system, which gives accurate ground-truth measurements. The attitude computed by the flight controller was sent to the GCS by an XBee 2.4 GHz transceiver. Attitude estimates of the quadrotor are shown in Figure 7, which suggests that the attitude computed by the algorithm is accurate and reliable.

### Controllers

It is well known that the open-loop rotational dynamics of a quadrotor are unstable as studied in [23]. The identified model reveals that poles are located in the right-half plane



**Figure 6.** NCF on SO(3) group [22].



**Figure 7.** Attitude estimates obtained from the Vicon system plotted in black dashed curves, and the NCF on the SO(3) group plotted in red solid curves.

of the real-imaginary axis and damping ratio is negative. Therefore, it needs to be stabilized by a feedback control algorithm, for example, SASs [24].

SAS, which makes the aircraft stable via the rate measurement in the feedback loop, is popular in aircraft control [25]. SAS is shown in Figure 8 with a dotted-line box. It consists of rate feedback with gain. If SAS is applied to a quadrotor, damping is increased. As a result, the quadrotor becomes controllable by a user.

The KKmulticopter project implements its SAS exactly as shown in Figure 9(f). Consequently, it has only three gyroscopes. Because the SAS only provides rate regulation, an autopilot is required to maintain the attitude of a quadrotor. We describe different autopilot structures. We begin with a proportional-derivative controller to illustrate how different architectures produce different characteristics.

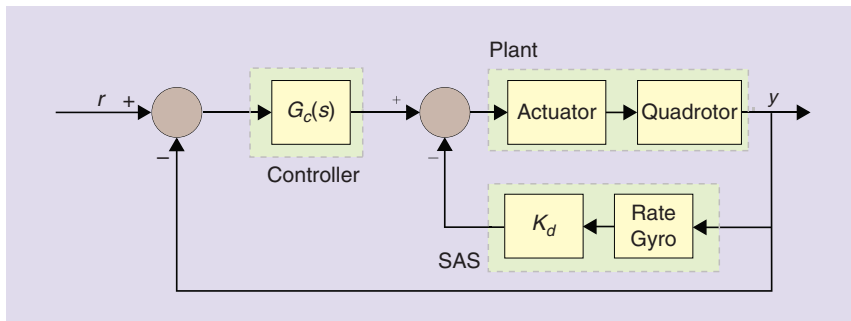
The Pixhawk project implements a single-feedback loop as shown in Figure 10(a). In this case, the controller is

$$G_c(s) = K_P + K_D s. \quad (11)$$

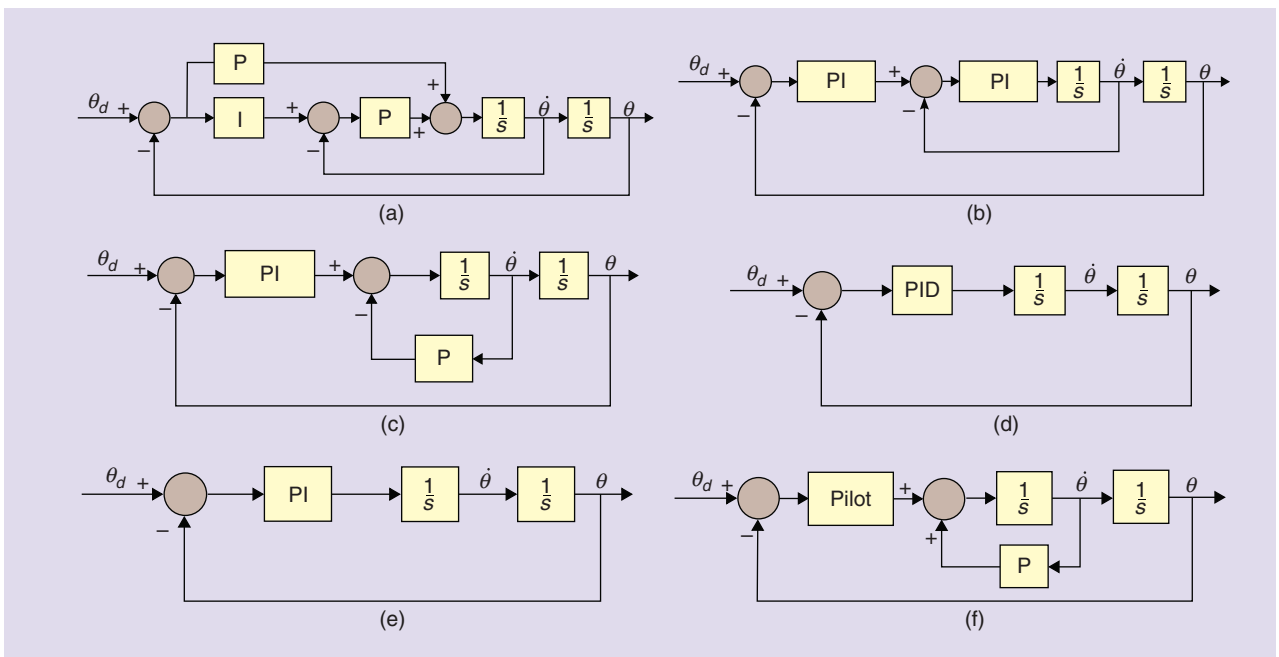
Then, the unity-feedback closed-loop transfer function is

$$\frac{Y}{R} = \frac{(K_P + K_D s)G_P}{1 + (K_P + K_D s)G_P}. \quad (12)$$

Desirable closed-loop poles can be achieved by adjusting  $K_P$  and  $K_D$ . In addition to the noise problem due to differentiation, there is now a



**Figure 8.** General attitude autopilot configuration [24]. Vehicle stability is enhanced by the SAS and the vehicle attitude is controlled by the outer loop with an integrator.



**Figure 9.** Various PID control structures of the OSPs. (a) Arducopter, (b) Openpilot, (c) Paparazzi, Multiwii (d) Pixhawk, Aeroquad, (e) Mikrokopter, and (f) KKMulticopter.



closed-loop zero at  $s = -K_P/K_D$ , and this zero can cause a large overshoot in case of a step disturbance unless the plant poles are heavily damped.

To resolve the above problem, consider the inner-loop configuration of Figure 10(b),

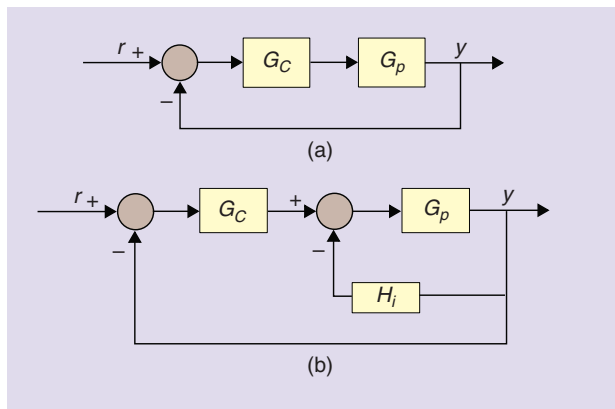
$$G_c = K_P, H_i = K_D s. \quad (13)$$

Then, the close-loop transfer function is

$$\frac{Y}{R} = \frac{K_P G_P}{1 + (K_P + K_D s) G_P}. \quad (14)$$

Therefore, with rate feedback, we can achieve the same closed-loop poles, but without an undesirable zero. Gyroscopes directly provide the rate information of the aircraft attitude, so this particular compensator structure is useful. All the other OSPs employ inner-loop configurations, while Pixhawk uses the single-loop configuration.

The Arducopter project implements a controller based on an inner-/outer-loop structure as shown in Figure 9(a). To mitigate steady-state error, an integrator is added in the forward path. The controller of the Openpilot project [Figure 9(b)] closely resembles Figure 8, which is proposed in [24]. The Paparazzi project also has the same structure, but the derivative term in the forward path is removed as shown in Figure 9(c). Therefore, Paparazzi controller is exactly the same as Figure 8. The Multiwii project has the same control configuration as Paparazzi. The Pixhawk project implemented a standard PID controller as shown in Figure 9(d). The rate measurement is not used in this controller. Only the derivative of the error signal is used to provide control inputs to the plant. As mentioned earlier, this controller may not be suitable for dynamic maneuvering, because a step change in the reference input will cause an undesirable initial spike in the control signal. However, Pixhawk is designed to support indoor navigation with relatively slow motion,



**Figure 10.** Feedback control with two configurations.  $G_p$  denotes the plant and  $H_i$  is the inner-loop controller. (a) Feedback control with a single-loop configuration. (b) Feedback control with an inner-loop configuration.

**Table 7. Summarized control structure and the number of gains to be tuned.**

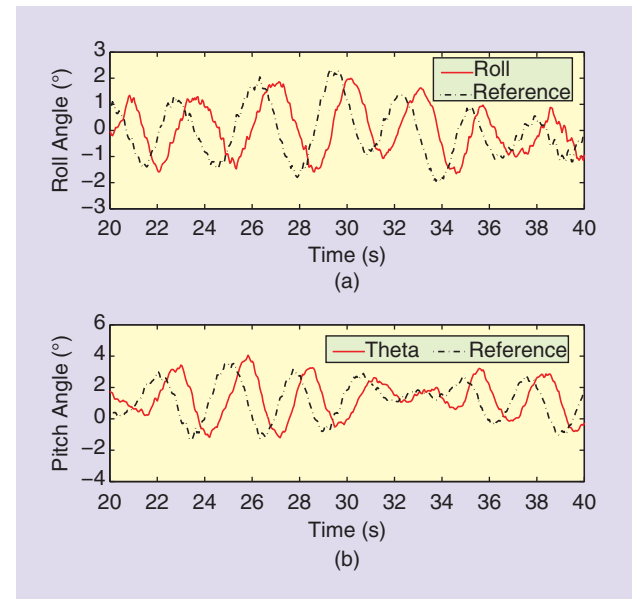
Project Name	Number of Gains on Each Axis	Controller Configuration
Arducopter	3 + 1 (antiwindup)	PI + P
Openpilot	4 + 2 (antiwindup)	PI + PI
Paparazzi	3 + 1 (antiwindup)	PI + P
Pixhawk	3 + 1 (antiwindup)	PID
Mikrokopter	2 + 1 (antiwindup)	PI
KKmulticopter	1	P
Multiwii	3 + 1 (antiwindup)	PI + P
Aeroquad	3 + 1 (antiwindup)	PID

so it is suitable for this purpose. Mikrokopter has only a proportional-integral (PI) controller in the forward path as shown in Figure 9(e).

As mentioned earlier, the KKmulticopter project only implements SAS that controls the body rate as shown in Figure 9(f). This controller is particularly useful for system identification experiments. This controller only depends on the gyroscope output:  $u = -Ky$ . In this case, we can perform closed-loop system identification to obtain an open-loop system model. Consider the following linear system:

$$\dot{x} = Ax + Bu, \quad y = Cx, \quad u = -Ky. \quad (15)$$

Then, the closed-loop system becomes  $\dot{x} = A_{cl}x$ , where  $A_{cl} = A - BKC$ . Therefore, once we perform closed-loop identification to obtain  $A_{cl}$ , the open-loop A matrix can also be obtained. This is only possible when there is no



**Figure 11.** Attitude tracking result of the quadrotor. Arducopter autopilot is tested with ground truth. The delay between reference and roll angle is due to communication delay.

**Table 8. Comparison of functionality between the OSPs.**

	Arducopter	Openpilot	Paparazzi	Pixhawk	Mikrokopter	KKmulti-copter	Multiwii	Aeroquad
Attitude estimation algorithm	NCF	EKF	NCF	EKF	LCF	—	LCF	NCF
GPS-based waypoint navigation	✓	Δ	✓	—	Δ	Δ	Δ <sup>a</sup>	—
Altitude hold	✓	Δ	✓	✓	✓	—	—	✓
Hardware in the loop simulation	✓	✓	✓	—	—	—	—	—
Support of other multirotor airframes	✓	✓	✓	—	✓	✓	—	✓
Support of computer vision	—	—	—	✓	—	—	—	—
GCS provided	✓	✓	✓	✓	✓	—	✓	✓
Airframe design provided <sup>b</sup>	✓	✓	—	—	—	—	—	✓
Camera stabilization	✓	✓	✓	—	Δ	—	✓	✓
Availability <sup>c</sup>	✓	—	✓	—	✓	✓	—	✓
Open-source license	LGPL	GPL	GPL	GPL	Δ <sup>d</sup>	—	GPL	GPL
Used by	[26], [27]	—	—	[28], [12]	[29], [30]	—	—	—

✓: supported, Δ: partially supported (e.g., additional navigation electronics), —: not supported.

<sup>a</sup>Only GPS-based homing is supported.

<sup>b</sup>The project provides a quadrotor airframe design in computer-aided design files.

<sup>c</sup>The project avionics on sale.

<sup>d</sup>Only noncommercial purposes.

integrator involved and  $K$  is fully known, which is not always the case for typical quadrotors in the market.

### Controller Parameters

Control structure and the number of gains to be tuned in each project are shown in Table 7. KKmulticopter is the simplest one, which has only one gain for tuning. Among many controller configurations, PI+P is dominant. P is for the inner loop (rate feedback), and PI is for the forward attitude error compensation.

### Controller Evaluation

We have constructed quadrotors using five different autopilots among mentioned OSPs: the Arducopter, Paparazzi, Mikrokopter, KKmulticopter, and Multiwii. Among these

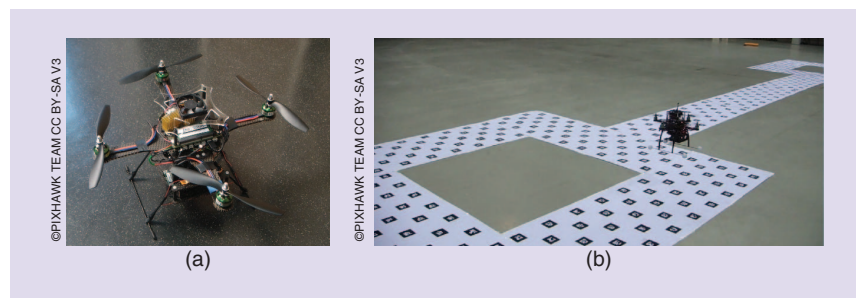
projects, Arducopter, Paparazzi, and Multiwii share the same controller composition as shown in Table 7. For qualitative evaluation, we mount markers on a quadrotor to acquire ground-truth data from the Vicon system. The desired angle is transmitted to the Arducopter-based quadrotor while quadrotor attitude from the Vicon and the transmitted commands are recorded simultaneously. The satisfactory attitude tracking result is shown in Figure 11. The delay is due to RC signal processing.

### Selection Guidelines

We have analyzed eight OSPs with attitude estimation algorithm, control configuration, electronic components, and features. A comparison of features between OSPs is given in Table 8.

### Availability of Flight Avionics

All the projects we described provide electronic schematic and bill of materials to reproduce their flight avionics. However, it takes high initial cost to manufacture electronics individually. Only five projects among them are available for purchase now: Arducopter, Paparazzi, Mikrokopter, KKmulticopter, and Aeroquad. It is recommended to start with these projects if a reader prefers to avoid electronics fabrication.



**Figure 12.** Pixhawk-based quadrotor platform with a camera and onboard computer [31]. (a) Pixhawk quadrotor platform and (b) flight environment of the Pixhawk quadrotor with ARToolkit markerboard on the floor.

### Attitude Estimation Algorithm Development

For attitude estimation tests, Arducopter and Paparazzi will be a good choice. The other projects are equipped with two or more gyro chips, which are hard to be calibrated for alignment. Only Arducopter and Paparazzi are equipped with 6-DoF IMU in a single chip: MPU 6000. The dynamic range is the best among the accelerometers and gyroscopes as described in Tables 3–6.

### Minimalistic Configuration

As studied in [23], an open-loop model can be easily identified when control input is fully known and no integrators exist in a controller as mentioned in the section “Attitude Estimation Algorithm Development.” Because SAS is implemented in [23] to identify the open-loop dynamics, KKmulticopter is a good choice to this end. The system is simple to understand and modify because a source code for attitude control is less than 500 lines in C.

### GPS-Based Navigation

For GPS-based outdoor missions (e.g., waypoint navigation and hovering): Arducopter, Openpilot, Paparazzi, or Mikrokopter will be a good choice. Only these projects support GPS-based navigation. Although Multiwii has GPS, it only supports a homing capability to move a quadrotor back to the initial position.

### Vision-Based Navigation

Only the Pixhawk project supports vision-based navigation capability. It can synchronize an IMU and a camera in hardware level, which allows tight integration of IMU measurements into the computer vision pipeline.

## Open-Source Projects in Research

### Vision-Based Navigation

The Pixhawk UAV is designed to be a research platform for computer-vision-based autonomous flight [28]. The Pixhawk team has constructed a localization test setup using augmented reality ToolKit+ (ARToolKit+). They successfully performed waypoint navigation using a camera on the localization test bed as shown in Figure 12.

In [26], adaptive image-based visual serving (IBVS) was integrated with adaptive sliding mode control based on Arducopter. Figure 13 shows the experiment in process where the inset picture is the image obtained from the onboard camera. The fiducial marker and its tracking result are shown.

Real-time vision-based localization was performed on a quadrotor system based on Arducopter [Figure 14(a)]

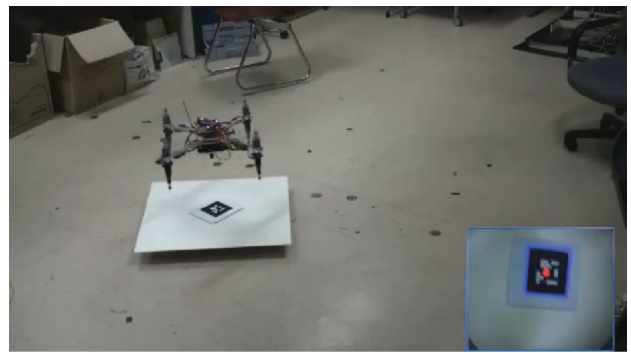


Figure 13. Arduino-based quadrotor platform with a camera, designed to perform IBVS [26].

[27]. This quadrotor is equipped with a frontal-view gray-scale USB2.0 camera with  $640 \times 480$  pixel resolution. Image data from the camera are transferred to a single-board computer and processed in a real time [Figure 14(b)] to obtain the vehicle location based on a map created in advance.

### Indoor Flight

A Mikrokopter-based quadrotor flew autonomously using a laser range finder (LRF) [29]. Equipped with LRF, Gumstix, and external IMU, it successfully performed autonomous indoor navigation without external localization sensors. Indoor position control based on an onboard LRF was performed on the Mikrokopter-based quadrotor platform shown in Figure 15 [32]. An autoregressive moving average with exogenous terms model of the stabilized Mikrokopter was identified in [30]. Recently, the quadrotor platform with shared autonomy was investigated for infrastructure inspection [33].

Multiagent-related research can be easily performed on the indoor quadrotor flight system. Especially, as the communication topology between agents can be user-defined within the GCS, various settings and algorithms can be exploited. Figure 16 shows three quadrotors in flight where an auction algorithm is being tested for online task

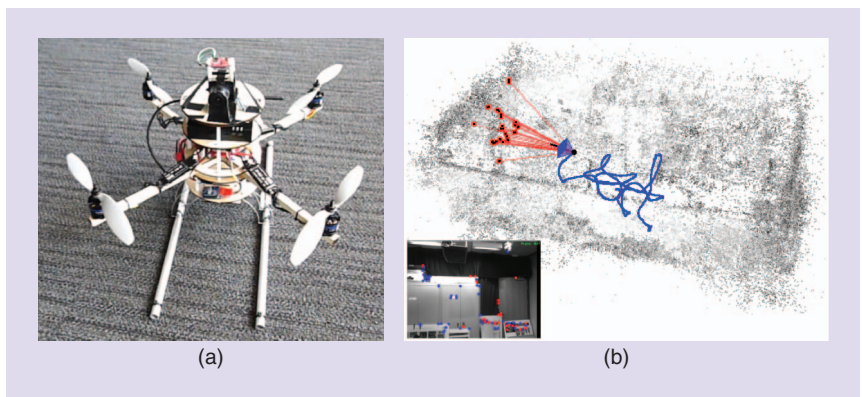


Figure 14. Arduino-based quadrotor equipped with a camera, single-board computer, and external IMU synchronized with a camera [27]. (a) Experimental quadrotor and (b) real-time vision-based localization is running on the single-board computer on a quadrotor.



**Figure 15.** Mikrokopter-based quadrotor equipped with Gumstix and LRF [32].



**Figure 16.** Three quadrotors in an experiment for an online task assignment algorithm.

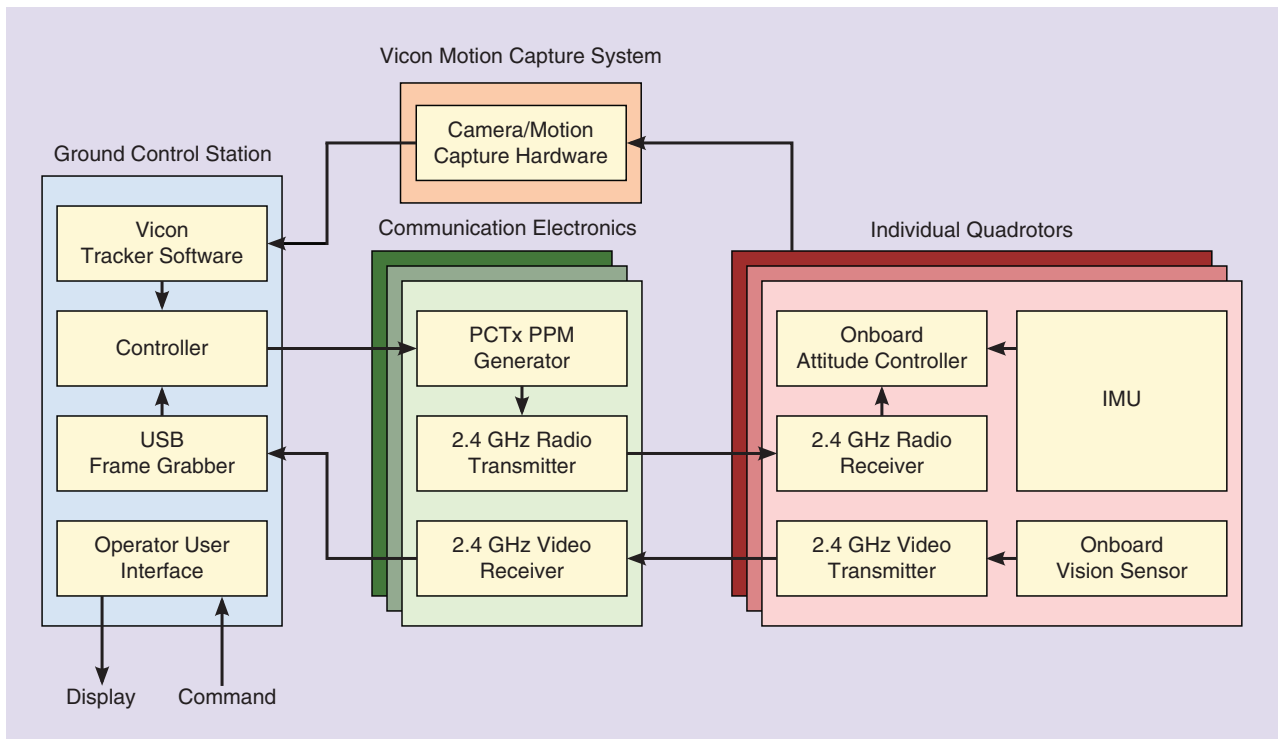
assignment. As described in Figure 17, each quadrotor is equipped with an onboard controller to track input commands sent by the GCS that collects position and/or attitude data of the quadrotors from the Vicon motion capture system. Data from the onboard vision sensors are sent to the GCS using a dedicated communication link.

### Conclusions

This article has presented eight quadrotor OSPs with descriptions of their avionics, sensor composition, analysis of attitude estimation and control algorithms, and comparison of additional features. Several research projects that use OSPs as a main flight controller are described.

Among the eight OSPs summarized in this article, we have implemented five by utilizing the benefit of OSPs that allow to build own systems at a low cost with less effort. To bring out continued improvements based on communities' work, objective evaluations of OSPs remain an important open problem.

The meaning of OSP had been more about software, but it is expanding to hardware and even products. There is already a project that has open hardware blueprints and a 3-D model of the quadrotor airframe that can be ordered from 3-D printing services. Sharing the same platform will become easier with such services. We expect that more OSPs for UAV will be initiated in the future.



**Figure 17.** Overall hardware architecture of the indoor quadrotor flight system for Figure 16. Multiple layers indicate one for each quadrotor.

## Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (nos. 20120000921 and 2012014219), and by the New and Renewable Energy Program of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Knowledge Economy (no. 20104010100490).

## References

- [1] M. Muller, S. Lupashin, and R. D'Andrea, "Quadcopter ball juggling," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Sept. 2011, pp. 5113–5120.
- [2] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 763–770.
- [3] V. Kumar. (2012, Feb.). Robots that fly and cooperate, in *Proc. TED Conf.* [Online]. Available: [http://www.ted.com/talks/lang/en/vijay\\_kumar\\_robots\\_that\\_fly\\_and](http://www.ted.com/talks/lang/en/vijay_kumar_robots_that_fly_and)
- [4] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," in *Proc. Int. Symp. Experimental Robotics*, Dec. 2010.
- [5] D. Mellinger, M. Shomin, and V. Kumar, "Control of quadrotors for robust perching and landing," in *Proc. Int. Powered Lift Conf.*, Oct. 2010.
- [6] S. Lupashin and R. D'Andrea, "Adaptive open-loop aerobatic maneuvers for quadcopters," in *Proc. Int. Federation of Automatic Control World Cong. (IFAC)*, 2011, pp. 2600–2606.
- [7] Q. Lindsey, D. Mellinger, and V. Kumar, "Construction of cubic structures with quadrotor teams," in *Proc. Robotics: Science and Systems*, June 2011.
- [8] P. E. I. Pounds, "Design, construction and control of a large quadrotor micro air vehicle," *Ph.D. dissertation*, Australian National Univ., 2007.
- [9] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 331–340, Apr. 2008.
- [10] S. Bouabdallah and R. Siegwart, "Towards intelligent miniature flying robots," in *Proc. Field and Service Robotics*, 2006, pp. 429–440.
- [11] G. Hoffmann, D. Rajnarayan, S. Waslander, D. Dostal, J. Jang, and C. Tomlin, "The stanford testbed of autonomous rotorcraft for multi agent control (STARMAC)," in *Proc. Digital Avionics Systems Conf.*, 2004, vol. 2, pp. 12.E.4–121.10.
- [12] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," in *Proc. ICRA*, May 2011, pp. 2992–2997.
- [13] P. Brisset, A. Drouin, M. Gorraz, P. Huard, and J. Tyler, "The paparazzi solution," in *Proc. Micro Aerial Vehicle*, Sandestin, Florida, 2006.
- [14] Free Software Foundation, Inc. (2007, June 29). GNU general public license [Online]. Available: <http://www.gnu.org/copyleft/gpl.html>.
- [15] S. Cousins, B. Gerkey, K. Conley, and W. Garage, "Sharing software with ROS," *IEEE Robot. Automat. Mag.*, vol. 17, no. 2, pp. 12–14, June 2010.
- [16] Free Software Foundation, Inc. (2007, June 29). GNU lesser GPL [Online]. Available: <http://www.gnu.org/licenses/lgpl.html>.
- [17] L. Heng, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "Autonomous obstacle avoidance and maneuvering on a vision-guided MAV using on-board processing," in *Proc. ICRA*, May 2011, pp. 2472–2477.
- [18] G. Lee, M. Achtelik, F. Fraundorfer, M. Pollefeys, and R. Siegwart, "Benchmarking tool for mav visual pose estimation," in *Proc. Int. Conf. Control, Automation, Robotics and Vision*, 2010, pp. 1541–1546.
- [19] G. H. Lee, F. Fraundorfer, and M. Pollefeys, "MAV visual SLAM with plane constraint," in *Proc. ICRA*, May 2011, pp. 3139–3144.
- [20] D. Simon, *Optimal State Estimation: Kalman, H [infinity] and Non-linear Approaches*. Hoboken, NJ: Wiley, 2006.
- [21] A. Pascoal, I. Kaminer, and P. Oliveira, "Navigation system design using time-varying complementary filters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 4, pp. 1099–1114, 2000.
- [22] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Trans. Automat. Contr.*, vol. 53, no. 5, pp. 1203–1218, June 2008.
- [23] D. S. Miller, "Open loop system identification of a micro quadrotor helicopter from closed loop data," Master's thesis, Univ. Maryland, College Park, 2011.
- [24] F. Lewis and B. Stevens, *Aircraft Control and Simulation*. Hoboken, NJ: Wiley, 2003.
- [25] R. Nelson, *Flight Stability and Automatic Control*. New York: McGraw-Hill, 1989.
- [26] D. Lee, H. Lim, H. J. Kim, and Y. Kim, "Adaptive image-based visual servoing for an under-actuated quadrotor system," *AIAA J. Guid. Control Dyn.*, vol. 35, no. 4, pp. 1335–1353.
- [27] H. Lim, S. N. Sinha, M. Cohen, and M. Uyttendaele, "Real-time image-based 6-dof localization in large-scale environments," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 1043–1050.
- [28] L. Meier, P. Tanskanen, L. Heng, G. Lee, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Auton. Robots*, vol. 33, no. 1, pp. 21–39, 2012.
- [29] S. Grzonka, G. Grisetti, and W. Burgard, "A fully autonomous indoor quadrotor," *IEEE Trans. Robot.*, vol. 28, no. 99, pp. 1–11, 2012.
- [30] I. Sa and P. Corke, "System identification, estimation and control for a cost effective open-source quadcopter," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2012, pp. 2202–2209.
- [31] ETHZ Pixhawk MAV [Online]. Available: <https://pixhawk.ethz.ch>.
- [32] I. Sa and P. Corke, "Estimation and control for an open-source quadcopter," in *Proc. Australian Conf. Robotics and Automation*, 2011.
- [33] I. Sa and P. Corke, "Vertical infrastructure inspection using a quadcopter and shared autonomy control," in *Proc. Int. Conf. Field and Service Robotics*, 2012.

**Hyon Lim**, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: hyonlim@snu.ac.kr.

**Jaemann Park**, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: pjm7981@snu.ac.kr.

**Daewon Lee**, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: dwsh001@snu.ac.kr.

**H.J. Kim**, Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, South Korea. E-mail: hjinkim@snu.ac.kr.

