

11

Mechatronic Design of Unmanned Aircraft Systems

Feng Lin, Fei Wang, Xiangxu Dong, Kemao Peng, and Ben M. Chen

CONTENTS

11.1	Introduction.....	404
11.2	Unmanned System Hardware	406
11.2.1	Sensors and Measurement Systems	406
11.2.1.1	Inertial Sensors.....	406
11.2.1.2	GPS/DGPS	410
11.2.1.3	Magnetometer.....	410
11.2.1.4	Lidar.....	411
11.2.1.5	Vision Sensor	412
11.2.1.6	RGB-D Camera	413
11.2.2	Computers.....	415
11.2.3	Actuator Management	415
11.2.4	Communication Unit.....	416
11.2.5	Hardware Integration.....	417
11.3	Unmanned System Software.....	421
11.3.1	Onboard Real-Time Software System	421
11.3.2	Ground Control Software System	424
11.4	Case I: Design of a Coaxial Rotorcraft System	424
11.4.1	Hardware System.....	424
11.4.1.1	Navigation Sensors	427
11.4.1.2	Computers.....	427
11.4.1.3	Servo Controller	428
11.4.1.4	Communication.....	429
11.4.1.5	Control Hub	429
11.4.1.6	Hardware Integration.....	429
11.4.2	Software System.....	432
11.4.2.1	Onboard Real-Time Software System	432
11.4.2.2	Ground Control Software System.....	433
11.4.3	Experimental Results	434
11.5	Case II: Design of a UAV Cargo Transportation System	436
11.5.1	Hardware System.....	436
11.5.1.1	Grabbing Mechanism.....	438
11.5.1.2	Sensors and Measurement Systems	440
11.5.1.3	Computers.....	441
11.5.2	Software System.....	442
11.5.3	Experimental Results	445
11.6	Conclusion	448
	References.....	448

SUMMARY This chapter presents the mechatronics design of unmanned aircraft systems to deliver autonomous capabilities in various missions. The modular design approach and computer-aided hardware integration have been addressed to achieve cost-effective development of the hardware system. The multitask embedded software design has been presented in the software system development to tackle complex missions. The proposed design approaches are feasible and effective, which have been verified in two case studies including a coaxial rotorcraft GremLion developed for the DARPA UAVForge challenge and a cargo transportation system used in UAVGP 2013. Such design approaches can also be extended to different unmanned systems for autonomous missions.

11.1 Introduction

Mechatronics is the synergistic combination of mechanical engineering, electrical engineering, control engineering, and computing engineering and also systems thinking in the design and development of products and manufacturing processes. An unmanned aerial vehicle (UAV) is a typical mechatronics system (see Figure 11.1), which is equipped with necessary data processing units, advanced sensors, an actuation system, automatic

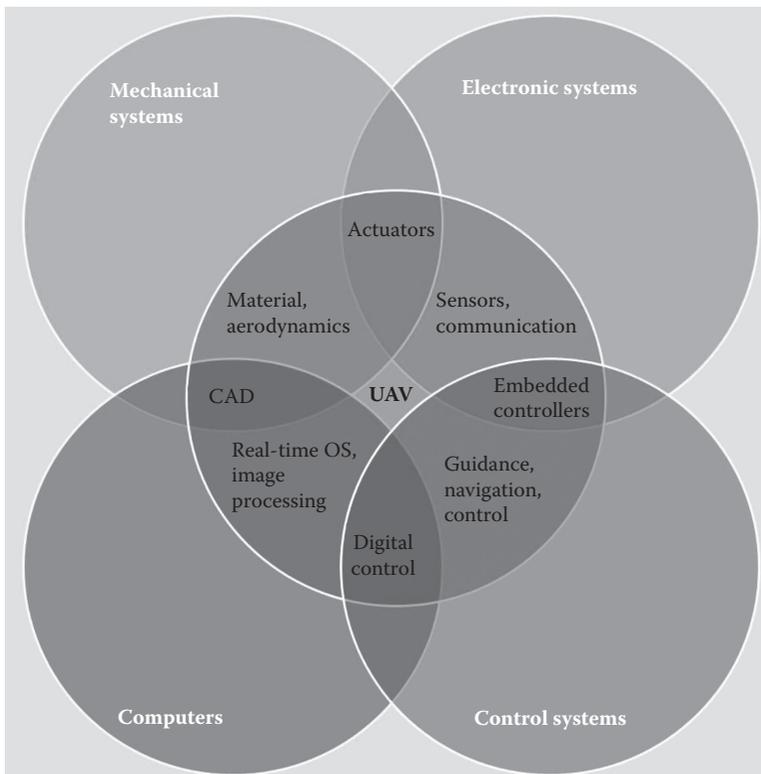


FIGURE 11.1
Mechatronics integration of UAVs.

control, and communication systems in order to perform autonomous flight missions without an onboard crew [1]. To fit its systematic nature, a UAV system is designed and developed in an integrated, cross-disciplinary manner.

In the last two decades, UAVs have made explosive growth in the civil and industrial markets, ranging from surveillance, reconnaissance, and agriculture to wildlife conservation [2–7]. It can be foreseen that the demand for UAVs will continue to grow, and they will become a valuable tool for researchers, rescuers, and other users in various working areas [8–15].

For instance, UAVs have been adopted as useful platforms to verify advanced flight algorithms. In [16], the researchers have proposed a hierarchical nonlinear flight controller that combines nonlinear feedback control for the inner loop and dynamic inversion for the outer loop. The proposed controller has been verified in a real flight test using a Raptor 90 helicopter equipped with the customized avionics to achieve automatic takeoff and landing, hovering, slithering, pirouetting, vertical turning, and spiral turning. In [17], a bumpless hybrid supervisory control algorithm has been applied to the formation control of UAVs, based on polar abstraction of the motion space and the use of properties of multi-affine functions over the partitioned space. Several actual flights have been conducted to evaluate the proposed algorithms using unmanned helicopters.

Another interesting research direction is to enhance navigation and localization capabilities of UAVs in extreme conditions, such as GPS-denied unknown environments. Simultaneous localization and mapping (SLAM) have been studied extensively by fusing the measurements of onboard sensors only [18–21]. Especially, vision-aided SLAM solutions have been tested successfully in [22,23], which are extremely useful for ultralight UAVs, such as insect-size flapping-wing UAVs.

In many practical applications, low-level autonomy has already been achieved successfully, such as obstacle avoidance, path planning, and trajectory tracking. Due to limitation in advanced perception and artificial intelligence, research in high-level autonomy of single or multiple UAVs in unsupervised situations is still in the preliminary stage and definitely needs great efforts to make a breakthrough. For instance, a formal framework and architecture using delegation has been proposed in [24] to support the specification, generation, and execution of a collaborative mission for multiple UAVs and associated operators. The cooperative search and exploration using multiple UAVs has been addressed in [25]. Moreover, there is a trend to combine UAVs with other unmanned systems, such as unmanned ground vehicles and unmanned surface vehicles, to form a large-scale and effective integrated unmanned system [26].

To satisfy the requirements of the aforementioned exciting applications, UAV systems need to be designed and developed in a systematic way. A typical UAV system consists of the following four parts: an aircraft, avionics, a manual control system, and a ground supporting system.

1. The aircraft is a platform to carry avionics and other payloads, which consist of the structure and the propulsion system.
2. Avionics are the onboard electronic systems that have the functions of sensing, stabilizing, navigating, perceiving, analyzing, communicating, planning, decision-making, and acting or executing. Avionics are composed of sensing and measurement systems, computers, communication units, embedded software systems, and associated equipment.
3. The ground supporting system provides a user-friendly interface for the display and management of UAVs. It includes a ground wireless transceiver and a

computer, which is able to monitor the states of the UAV in real time and send the command to the UAV through this interface.

4. The manual control system is a backup system for UAVs, which normally consists of a pilot and a wireless joystick. It is very useful in unpredicted events, such as emergency landing or mechanical or electrical failures of UAVs.

In this chapter, we aim to address systematic design approaches of the avionics and ground control system from the perspective of mechatronics, including hardware and software systems. We will further illustrate the design approaches by using a couple of UAV systems developed by the Unmanned System Research Group in the National University of Singapore (shown in Figure 11.2). Due to word limit, the platform design and the manual control system are not covered in this chapter. Reader please refer to [27,28] for more details. We emphasize that the UAV systems discussed in this chapter belong to the category of mini UAVs [28] and are mainly used for academic and civilian applications.

The remainder of this chapter is organized as follows: The hardware system design and integration is presented in Section 11.2. The software system development is presented in Section 11.3. Several case studies are explored in Sections 11.4 and 11.5. Finally, we draw the conclusion in Section 11.6.

11.2 Unmanned System Hardware

Avionics are composed of various sensors, computers, actuators, and communication units as illustrated in Figure 11.3, and they can be divided into multiple self-contained functional modules and implemented using the modular design approach. Each module needs to be developed or purchased separately with industrial standard interfaces. Such modularity offers benefits such as a reduction in cost and flexibility augmentation. The functions and features of those essential modules and the hardware integration issues will be addressed in the following sections.

11.2.1 Sensors and Measurement Systems

Unlike a radio controlled “drone,” a UAV is an autonomous system that is able to sense, make decisions, and react without human interaction. The onboard sensors are combined together to obtain the best-achievable estimation, including 6 DoF rigid body motion and the perception of the surrounding environment. The characteristics of those key sensors are addressed in this section, and the implementation will be presented in Sections 11.4 and 11.5.

11.2.1.1 Inertial Sensors

An inertial navigation system (INS) is a self-contained and core navigation solution, which can dead reckon the attitude, velocity, and position without external navigation references. The basic element of INS is an inertial measurement unit (IMU) composed of a three-axis accelerometer measuring the linear acceleration in the inertial reference frame resolved in the carrier’s body coordinate, and a three-axis gyroscope providing the angular rates in the inertial reference coordinate. A couple of examples of INS have been



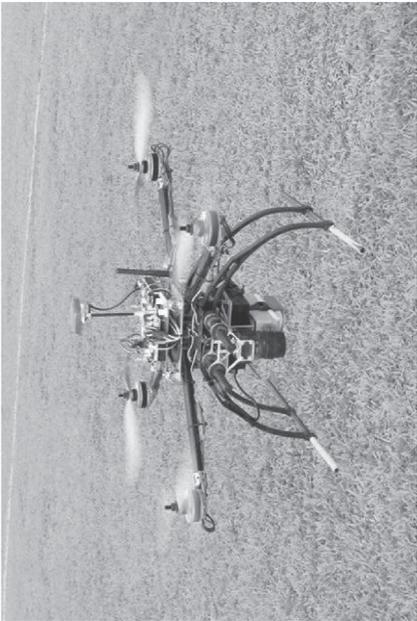
(b)



(d)



(a)



(c)

FIGURE 11.2 UAVs developed by the Unmanned System Research Group in NUS: (a) Helion, (b) GremLion, (c) Q-Lion, and (d) K-Lion.

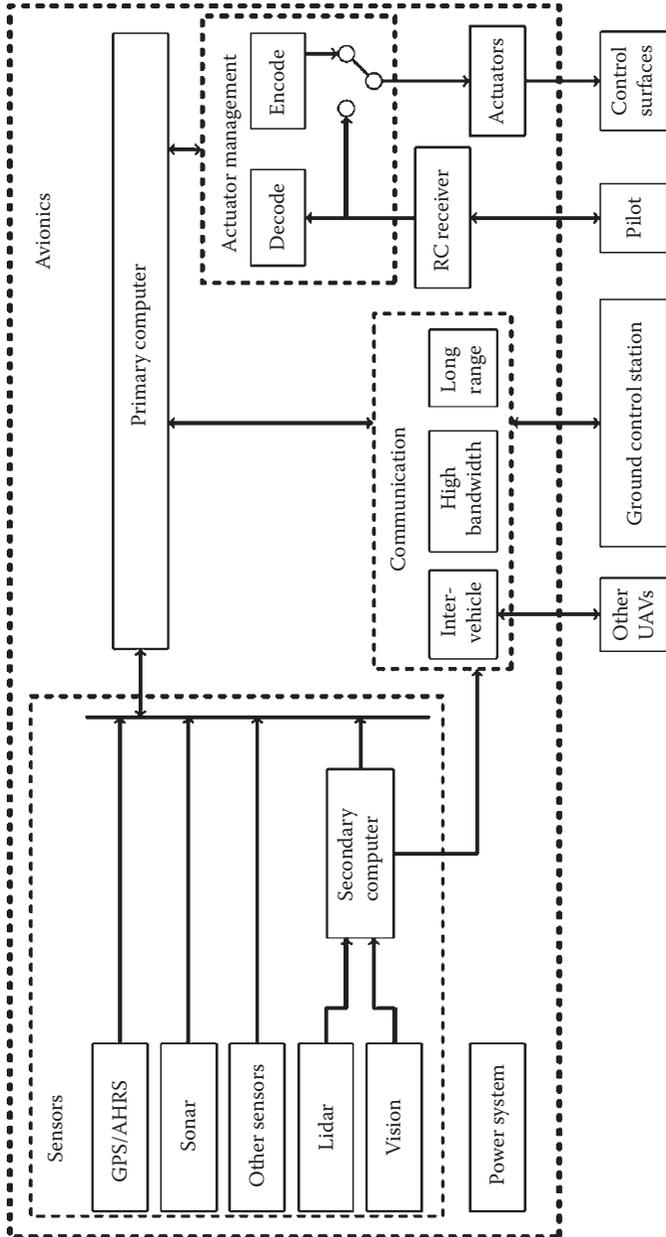


FIGURE 11.3 Hardware configuration of a UAV.

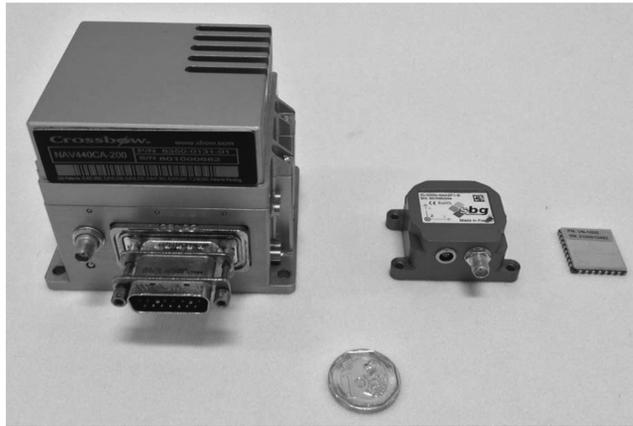


FIGURE 11.4
Inertial navigation systems (left to right): NAV440, IG500N, and Vectornav VN-100.

shown in Figure 11.4. A comprehensive list of commercial available small IMUs and INSs can be found in [29].

The navigation accuracy of the INS is significantly affected by the IMU used, and IMUs can be roughly divided into four performance categories: marine and aviation grade, tactical grade, industrial grade, and automotive and consumer grade [30]. The specification differences of the four performance categories are given in Table 11.1 (see detailed description in [31]), based on the commercial available IMU and INS systems. The bias and noise level of the sensors eventually determine the grade of IMU.

To improve the navigation performance of INSs, especially industrial- or tactical-grade ones, the inertial sensors need to be integrated with a three-axis magnetic sensor to form an attitude and heading reference system (AHRS) that is able to provide filtered attitude and heading solutions with an onboard processing system in the AHRS. In addition to the magnetic sensor, it is common and useful to employ a GPS receiver together with the AHRS to obtain bounded position and velocity estimation. The whole navigation system is referred to as the GPS-aided AHRS or the GPS-aided INS that is able to provide complete navigation information. Moreover, to improve altitude estimation, a barometer is

TABLE 11.1
IMU Categories

IMU Grade	Marine and Aviation	Tactical	Industrial	Automotive
Cost (USD)	100 k–1 M	10–30 k	0.5–3 k	<500
Gyro type	Laser/fiber optic	Fiber optic/MEMS	MEMS	MEMS
Gyro bias (°/h)	0.001–0.01	0.01–1	1–100	>100
Gyro random walk (°/√h)	<0.005	0.005–0.5	0.5–5	>5
Accelerometer bias (mg)	0.01–0.1	0.1–1	1–10	>10
Example	Honeywell HG9900 Honeywell HG9848	Honeywell HG1900 Spatial FOG	Crossbow IMU440 Microstrain GX2 SBG IG500A Vectornav VN-100	ArduIMU RasorIMU

integrated into the navigation system. In indoor applications, the external reference system may come from a motion-capture system, providing position and velocity measurements.

There are a large number of small IMU/INS devices available on the market nowadays. To select a suitable one for a UAV system, we are in accordance with the following concerns:

1. If the extreme or acrobatic flight conditions are not taken into consideration, the industrial-grade IMU/INS devices are strongly recommended, which have acceptable performance, affordable prices, and low maintenance cost. For unconventional or fast maneuverable aircrafts, the acceleration, angular rate, and magnetics may change dramatically during flight. As the conventional expression is mathematically singular in one of the flight modes, the quaternion-based motion estimation is compulsory in such situations. In addition, a high sampling rate and large measurement range are also required in such applications.
2. On the basis of meeting all of abovementioned requirements, the size, weight, and power consumption of the adopted IMU/INS device should be minimized.
3. Pure inertial navigation suffers from the integration drift [30]. In practice, inertial navigation needs to be aided by external references, such as GPS, magnetometer, altimeter (barometric, lidar, radar), and stereo vision to realize drift-free state estimation.

11.2.1.2 GPS/DGPS

GPS is widely used nowadays in numerous civil and military applications. The GPS can be used to correct the bias and error of the inertial navigation system. The majority of GPS receivers that are commercially available work at L1-band (1.57542 GHz). Information provided by the GPS receivers commonly includes position in the geodetic coordinate system with about 3 m CEP (3 m circular error probable [32]), velocity in the vehicle-carried NED frame (with the accuracy about 0.5 m/s), and time when the information package is sent.

The accuracy of position and velocity can be further improved to the meter or even centimeter level by employing advanced positioning methodologies, such as differential GPS (DGPS) and real-time kinematic (RTK) navigation methods. However, a stationary GPS base station is required in order to achieve high accuracy measurement. GPS measurement is drift-free and thus commonly blended into the estimation algorithm as the periodical reference signal. Two major deficiencies of GPS receivers are the vulnerability in a poor-visibility environment and low update rate (commonly 1 to 4 Hz).

11.2.1.3 Magnetometer

A magnetometer is to measure the magnetic fields of the Earth. A three-axis magnetometer can be used for estimating heading and gyro bias. In UAV systems, the magnetometer can effectively provide initial reference and periodical correction for the heading angle via measuring the strength of the magnetic field. Most of the commercial magnetometers are MEMS-based and with a sufficient resolution (milligauss level) and sampling rate (up to 10 Hz). For specific implementation in miniature UAVs, special attention should be paid to electromagnetic interference (EMI) shielding and hard- and soft-iron calibrations for each particular platform.

11.2.1.4 Lidar

Relative sensing in unknown environments is a challenging and exciting research topic for autonomy of unmanned systems. The most reliable and promising solutions for small-scale UAVs are lidars and stereo vision. We will introduce lidars first, and the detailed description of stereo vision will be addressed later.

The lidar is a laser-based sensing technology measuring distance by computing time of flight of the laser beam. The lidar technology has shown its efficiency, accuracy, and reliability in many applications, such as surveying, mapping, and autonomous navigation. As an active sensing technique, lidar systems can work under a wide range of environmental conditions, such as day, night, dusk, and dawn. Lidar systems have been used for several decades, but the previous lidar systems were so heavy and bulky that they could only be used in manned aircrafts [33]. With the rapid growth of small-size unmanned systems, there are a variety of ultralight lidar systems available now (for example, see Figure 11.5). According to the type of the scanned area, lidar systems can be categorized into two types: 2-D and 3-D lidars.

A typical 2-D lidar is composed of a laser range finder and a spinning mirror that can directly provide accurate displacement measurements in the scanning plan. Compared to 3-D lidars, 2-D lidars are lightweight and have low power consumption and relatively low cost, which makes them suited for unmanned systems navigating in structured environments [19]. A comparison of commercial off-the-shelf 2-D lidar products is given in Table 11.2. However, the quantity of information encapsulated in a 2-D lidar is not sufficient for crucial and demanding tasks such as obstacle detection in cluttered environments where complete 3-D information is required.

A 3-D scanning lidar obviously can provide much more information about the environment in just one scan. One straightforward way to generate 3-D measurement is to install a 2-D lidar unit in a sweeping or spinning mechanism. Another sophisticated approach is to use the configuration of an array of individual laser sensors, and the entire system

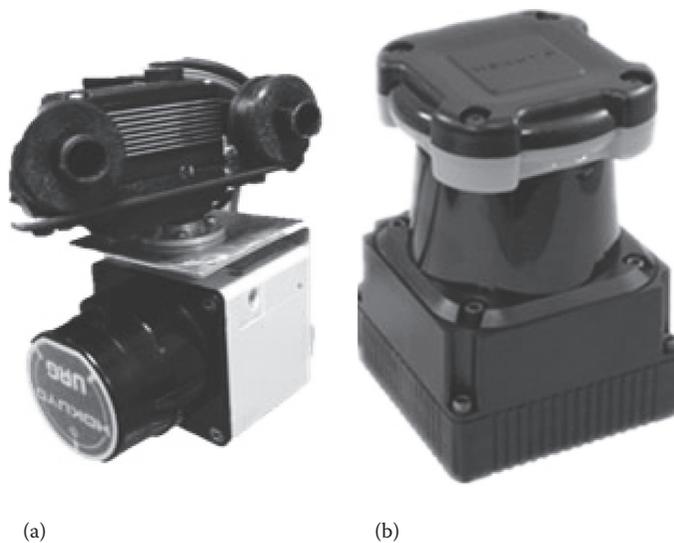


FIGURE 11.5
2-D lidars: (a) URG-04LX, (b) UTM-30LX.

TABLE 11.2

2-D Lidar Categories

2-D lidar	Hokuyo URG-04LX	Hokuyo UTM-30LX	SICK LMS511	RIEGL VQ-180
Detection range (m)	4	30	80	150
Accuracy (mm)	10	30	50	15
Scan angle (°)	240	270	190	100
Angular resolution (°)	0.36	0.25	0.25	0.001
Scan speed (ms)	100	25	13	100
Power consumption (W)	3	8	22	50
Weight (kg)	0.16	0.21	3.7	9

TABLE 11.3

3-D Lidar Categories

3-D Lidar	Velodyne HDL-32	Fraunhofer 3DLS	Riegl LMS-Z420i	Ocular Robotics RE05
Detection range (m)	70	30	1000	30
Scan angle, vertical (°)	40	124	124	70
Scan angle, horizontal (°)	360	180	360	360
Angular resolution, vertical (°)	0.16	0.25	0.002	0.01
Angular resolution, horizontal (°)	1.33	0.25	0.0025	0.002
Scan speed, vertical (sec)	0.1	26.64	0.05	0.05
Scan speed, horizontal (sec)	0.1	26.64	24	0.05
Data rate (points/sec)	7×10^5	7.6×10^3	1.1×10^4	3×10^4
Power consumption (W)	12	20	78	70
Weight (kg)	2	7.4	16	2.5

spins. The most popular and successful 3-D lidar is the Velodyne lidar, which has been widely used in the DARPA urban challenge. Recently, a Velodyne lidar has been used on a quadrotor for 3-D mapping and surveillance [34]. A comparison of commercial off-the-shelf 3-D lidars is given in Table 11.3. A more detailed comparison and discussion of 3-D lidars is given in [35], which also provides the 3-D coverage of different lidar systems.

The main drawback of laser sensing is the weight and high power consumption due to the nature of active sensing. Although a 3-D lidar is able to provide one more dimensional measurement than a 2-D lidar, it is hard for it to be carried by a small-scale UAV with limited payload and power supply. Therefore, in localization and navigation applications, the combination of two 2-D lidars is a feasible solution, balancing the quality of 3-D measurement against total weight. In addition, fusing a lidar with other sensors, such as a camera, can also provide instant 3-D information [36]. An application of using a lidar system on a small-scale UAV will be presented in Section 11.5. In practice, measurements of the lidar are affected by the attitude and velocity of the carrying vehicles, and it is necessary to do motion compensation for a moving platform.

11.2.1.5 Vision Sensor

There is a boom in vision sensing in academic research and industrial applications because it is able to provide human-like perception, such as geometry of the scene, photometry

of objects, and dynamics of the environment. By integrating vision sensors with other avionic sensors, unmanned systems can autonomously perform a variety of applications, such as vision-based stabilization [37,38], air-to-air tacking [39], navigation in complex environments [40], vision-based pose estimation, and autonomous landing [41,42] as well as localization and mapping [43,44]. Generally speaking, those applications can be roughly divided into two categories, depending on how to use the extracted vision information:

1. Vision-aided target acquisition: Vision sensing is used to search and identify obstacles or targets in the surrounding environment and estimate their relative distance and orientation to the UAV. The estimated information is used to guide the UAV to avoid the obstacles or keep tracking certain targets. In such applications, the motion of the UAV is normally known, and the main challenges are the vision-based object detection and robust tracking algorithms.
2. Vision-aided motion estimation: In contrast to target acquisition, vision sensing is used to estimate the motion of the UAV itself, such as position, velocity, and heading, by fusing vision measurement with the aforementioned navigation sensors and a predefined map. In mapless situations, vision-aided SLAM algorithms need to be utilized to generate the map at the same time. Feature extraction methods in near homogenous environments and data fusion algorithms among different sensors still need to be further investigated.

A single camera is able to handle applications of vision-aided target identification and tracking. When it comes to situations in which relative distance measurement is required, the recommended reliable solution is to utilize stereo vision that is able to provide complete 3-D information. However, stereo vision requires extensive computation and also needs a large baseline between cameras. To solve this problem, we can still use the monocular camera solution for 3-D sensing, which requires relatively lower complexity in terms of hardware and computation than stereo vision as long as the geometry information of the targets or environments is known or partially known.

The performance of vision sensors is greatly affected by environmental conditions, especially lighting conditions. According to operating conditions, vision sensors can be divided into three categories: daylight cameras, night cameras, and thermal cameras. Daylight and night cameras work on the same principle of generating an image based on reflected visible light energy. The major difference is that night cameras greatly magnify small amounts of visible light in dark environments. Unlike daylight or night cameras, thermal cameras produce an image by detecting thermal radiation, and they are able to tell differences in heat as small as 0.01°C. Therefore, they can work in various weather or lighting conditions, such as night, light fog, rain, and smoke.

11.2.1.6 RGB-D Camera

Despite the common usage of monocular or stereo cameras on UAV platforms, the depth camera is a more advanced vision sensor, which can also be used to solve UAV navigation and environmental mapping problems. A depth camera is also called a range camera, flash lidar, or RGB-D camera. Different from traditional cameras, which can only provide 2-D color information, the depth camera also contains per-pixel depth information. There are two main types of depth cameras, namely the time-of-flight (ToF) cameras and the more recently invented projected-light cameras.

Most ToF cameras are very expensive, costing around US\$10,000 each. Hence, they may not be suitable for small-scale robotics projects with limited budgets. In contrast, the new generation of consumer depth cameras based on the projected-light technology is much cheaper. Some of them cost only around US\$150. Typical examples include the Microsoft Kinect and the Asus Xtion PRO LIVE. The per-pixel depth sensor technology used in these projected-light cameras was first developed by PrimeSense. The technology is patented (United States Patent US7433024). These sensors are able to project a known infrared speckle pattern, which can then be captured by an infrared camera and compared part-by-part to the reference patterns, captured at known depths and stored in the device previously. An algorithm runs in the device to estimate the per-pixel depth based on the reference pattern that the projected pattern matches best. The depth data is then associated with a precalibrated RGB camera and yields a point cloud in 3-D space, with which each point has also color information associated. In addition, approximated surface normals are sometimes calculated and stored with each point.

Commercially available depth cameras have their respective advantages and disadvantages in aspects such as range, resolution, frame rate, field of view, weight, dimension, and so on. Table 11.4 lists the key specifications of four popular depth cameras, which can possibly be used on UAV onboard systems. In spite of their specification differences, they have one common shortcoming, which is the drastic drop in performance when used in outdoor environments. Due to sunlight interfering with the infrared light source, most depth cameras simply fail in outdoor environments. This is expected as these products were originally developed for indoor electronic gaming purposes. Nevertheless, companies such as PMDtec are doing investigations on ambient light suppression technology to ensure more robust performance, which can possibly solve the problem of outdoor usage.

Depth cameras have been utilized in various robotics and UAV-related projects. In [45], the consumer RGB-D cameras and their applications are reviewed. It is mentioned that consumer RGB-D cameras have been used for applications including RGB-D mapping [46,47], interactive 3-D modeling of indoor environments [48], UAV autonomous flight [49,50], and real-time visual and point cloud SLAM [51]. SLAM based on a depth camera has become popular in recent years. In [52], an RGB-D SLAM system is proposed and evaluated via a large set of sequences of different indoor scenes with varying camera speeds and illumination conditions.

To work with consumer RGB-D cameras, several software libraries can be used. They include the Point Cloud Library (PCL), the Microsoft Kinect SDK, and OpenCV. PCL handles data structure, filters, segmentation, registration, visualization of 3-D point clouds,

TABLE 11.4

Comparison Between Different Depth Camera Products

	PMD [vision] CamCube 3.0	Mesa Imaging SR4000	Microsoft Kinect	Asus Xtion PRO LIVE
Type	ToF	ToF	Projected-light	Projected-light
Range (m)	7.5	0.1–10	0.8–4	0.8–3.5
Resolution	200 × 200 or 160 × 120	176 × 144	640 × 480	640 × 480 or 320 × 240
Frame rate	40 or 80	50	30	30 or 60
FOV (°)	40 × 40	43.6 × 34.6	57 × 43	58 × 45
Weight (g)	1438	470	440	228
Dimension (cm)	Not published	65 × 65 × 68	24.9 × 6.6 × 6.7	18 × 3.5 × 5

and input–output interface with Microsoft Kinect, Asus Xtion, and other RGB-D sensors based on the PrimeSense technology. The Microsoft Kinect SDK only works for Kinect, and it is focused more toward human body action identification. OpenCV has abundant useful functions to deal with 2-D image processing and some 3-D data interpretation.

11.2.2 Computers

The primary functions of onboard computers include analyzing and processing various data delivered by onboard sensors, executing missions, communicating with the ground station and other UAVs, and logging flight data for post-flight analysis.

To select suitable onboard computers, special attention should be paid to size, weight, input/output (I/O) port configuration, expendability, antivibration property, and power consumption.

A single board computer (SBC) is still the first choice for UAV systems as they have a compact size and the complete features of a fully functional computer, including microprocessor(s), memory, input/output, storage, and so on. Especially, PC-104–based SBCs are strongly recommended because they are designed for harsh environments from the ground up. For instance, a UAV system using a PC-104–based flight control computer has been reported in [53].

In order to implement sophisticated vision algorithms, powerful and advanced SBCs have been widely used [53–55], including the popular PC/104(-plus)–based SBCs and other powerful SBCs, such as the Mastermind computer from Ascending Technologies and fit-PC2i from fit PC. For example, a vision system using a Lippert CoreExpress 1.6 GHz Intel Atom board with a WiFi link has been proposed in [56]. Feature detection and frame-to-frame motion estimation algorithms were implemented to realize autonomous navigation of a quadrotor UAV in indoor environments. On the other hand, micro UAVs expect ultralight and small-size onboard computers, such as computer-on-modules (COMs). The authors in [57] presented a vision system they developed based on a Gumstix Overo Fire with 600 MHz and a webcam to realize vision-aided indoor navigation. A novel and efficient vision algorithm was proposed to realize the robust landmark tracking and path generation onboard. The flight tests verified the robustness and efficiency of the proposed system. In addition to SBCs, it is also useful to implement vision algorithms in special embedded systems such as digital signal processors (DSPs) and field-programmable gate arrays (FPGAs) to speed up the processing, but these technologies require long development time and special development technique compared with the general purpose processors.

11.2.3 Actuator Management

Actuator management is to realize smooth switching between the manual control mode and the automatic control mode. The requirements for the actuator management are listed as follows:

1. Reliable switching function: The switching between automatic control and manual control should be fast and smooth.
2. Sufficient input/output channels: For most aircrafts, five onboard servos are equipped to drive the aircraft. Adding an extra channel for switching function and some necessary redundancy, the channel number must not be less than seven.

3. Capability of recording the servo actuator's input signal: This function is important in initial manual flight tests. The recorded input data are essential for deriving the dynamic model of the UAV and for evaluating control performance.
4. High resolution: Substantially, the input-recording and servo-driving functions are the A/D and D/A procedure. The resolution should be sufficiently high to ensure the data consistency and accuracy.

11.2.4 Communication Unit

The communication units in the UAV system framework are deployed as interfaces between the UAV entity itself and external entities. The external entity can be the GCS for the ground operator or another UAV entity for information exchange. With UAV-to-GCS communications, the operator can remotely control and monitor the UAVs in operation. With inter-UAV communications, the UAV team can multiply their capability and effectiveness in cooperative tasks. Generic communication architecture is illustrated in Figure 11.6.

Although various types of communication devices are available on the market, they can be categorized into two categories based on the device protocols. One is the commonly used UART protocol-based devices. The other one is the TCP/IP protocol-based devices.

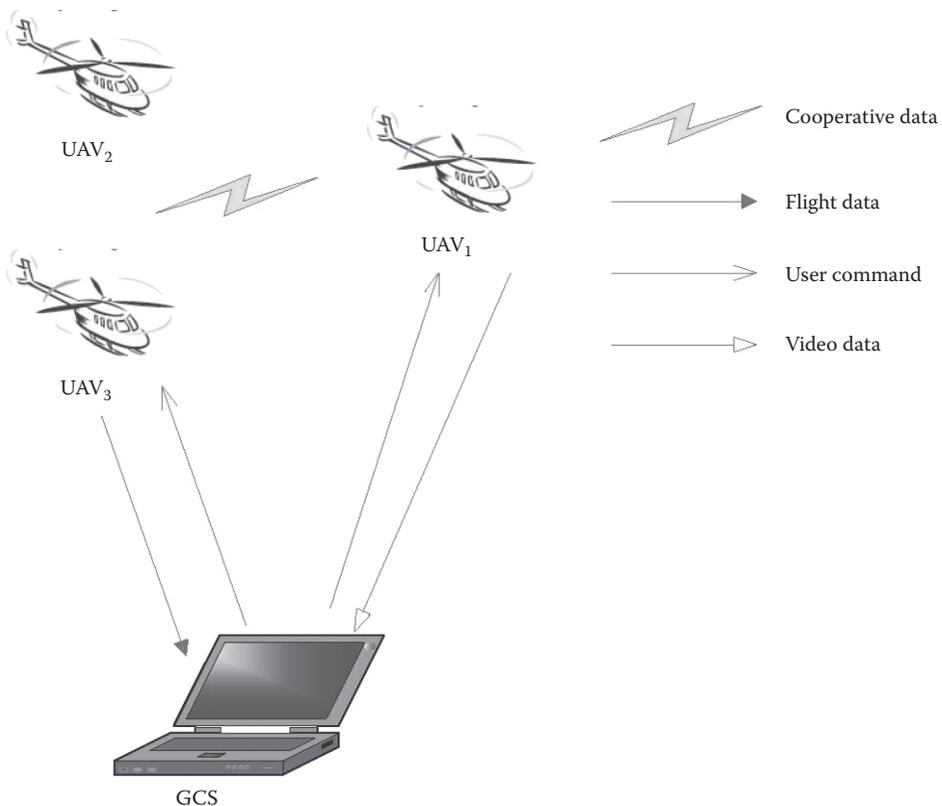


FIGURE 11.6
Communication architecture in a multiple-UAV system.

TABLE 11.5
Communication Device Specifications

Communication Module	Data Bandwidth	Range	Protocol
UART interface	115,200 bps	32 km	UART
802.11g	54 Mbps	250 m	TCP/IP
3G modem	300 kbps	100 km	TCP/IP
4G modem	5 Mbps	100 km	TCP/IP

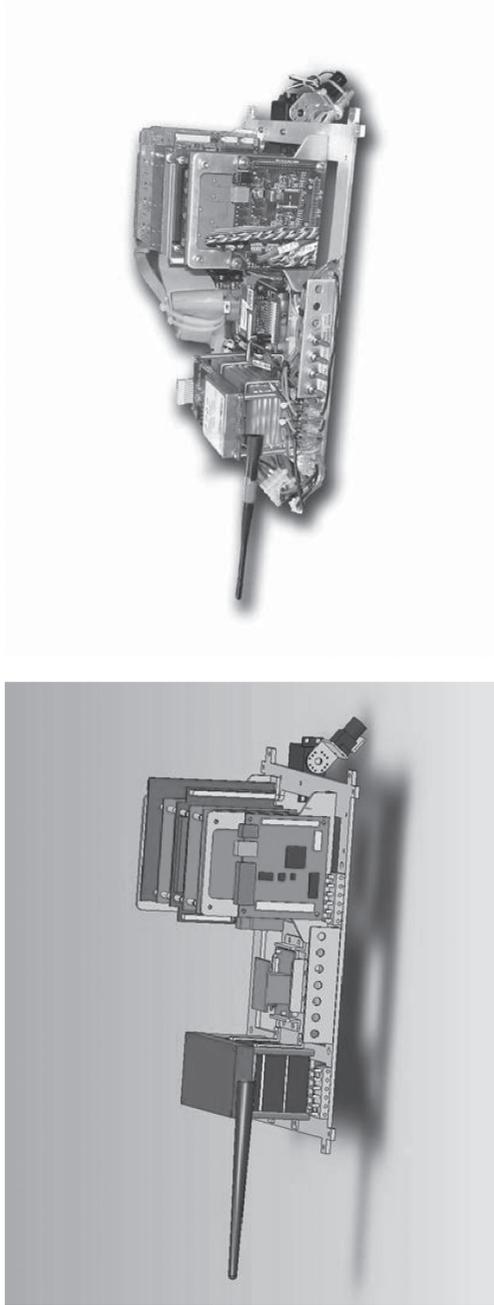
The UART devices are mostly used for long range and low data bandwidth communication. TCP/IP enabled wireless devices are utilized for high data bandwidth communication, such as 802.11 a/b/g/n modules and telecommunication modules with 3G/4G functions. Equipped with such a high bandwidth communication unit, both status data and onboard video can be transmitted in real time. The typical values of data bandwidth and communication range of those communication devices are summarized in Table 11.5. We observe that the 4G devices can deliver both high data bandwidth and long distance coverage at the same time but at the cost of a relatively high subscription fee to local telecommunication service providers. On the other hand, the 802.11-based devices can work in ad hoc mode without any communication infrastructures compared with the 3G/4G devices.

However, once a UAV flies at low altitude or into cluttered environments, such as an urban canyon, the communication range and bandwidth will decrease dramatically due to no line of sight (LOS) between the transmitting antenna and the receiving antenna. A possible solution is to use a second high-fly UAV as a relay, which can extend the range substantially. Of course, the relay UAV introduces additional complexity, such as self-jamming, antenna placement, and the dynamic mission planning for UAVs to serve as communication relays [7].

11.2.5 Hardware Integration

Based on the hardware components selected, we now proceed to carry out systematic integration of those components to achieve reliable performance.

1. Computer-aided virtual design: The lack of a powerful 3-D design environment causes great difficulty in layout design and the integration of hardware components. As a result, the design and integration procedure has to be iterated quite a number of times, which prolongs the total construction time. To avoid such problems, a powerful virtual design environment should be adopted, such as SolidWorks. In the virtual design environment, the virtual counterpart can be modeled to be identical to the real hardware component, both in shape and material properties. When the 3-D design is finished, the corresponding 2-D views will be generated at the same time for the convenience of mechanical manufacturing. Layout design for the avionic system can be realized in the virtual design environment to achieve near optimal CG balance and space usage. One example of virtually designed avionics and its real counterpart is illustrated in Figure 11.7. In addition to the layout design, the simulation environment is also used for the mechanical structure design and modeling. For example, landing gear for a UAV helicopter has been analyzed in SolidWorks to simulate its strength and stiffness (shown in Figure 11.8) to reduce the need for costly prototypes or design changes later on.



(b)

(a)

FIGURE 11.7 An example of virtually designed avionics and real avionics.



FIGURE 11.8
Structure analysis of landing gear.

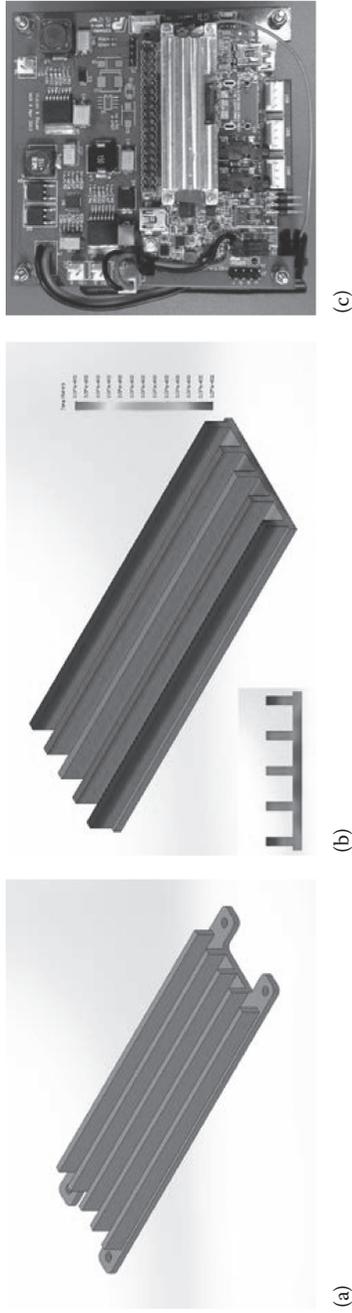


FIGURE 11.9 Heat sink designed for an ARM-based processor: (a) virtual design, (b) temptation distribution, and (c) installed.

2. Antivibration design: For any small-scale UAVs, there are several main vibration sources that should be taken into account: the rotation of the rotors or propellers and the power engine. It has the potential to introduce bias to measurement data and to cause loss of connection of mechanical components. For this reason, an antivibration design is necessary to ensure the overall onboard system working properly and reliably.
3. Power supply design: The main consideration in the power supply design is to meet the overall experimental requirements and system safety. For example, although a single battery is sufficient to power the avionic devices, two batteries have been chosen instead to enhance the overall safety of the system. Thus, the system can still run smoothly and guarantee manual maneuvering even if one of the batteries is out of power.
4. EMI shielding design: Electromagnetic interference (EMI) is a serious issue for small-scaled UAVs as all of the highly integrated electronic components are required to be mounted in a very limited space. The main problems aroused by EMI include reducing the effective range of RC manual control, generating errors in INS/GPS measurements, and causing data losses in wireless communications. These problems have to be eliminated or reduced to a minimum before conducting actual flight tests. It is suggested to use aluminum boxes and foil to isolate the necessary electronic components.
5. Thermal analysis: All the electronic components have a temperature-dependent property. It is a necessary consideration to design a guard over the component that can easily overheat, such as processors and communication units. For instance, we have designed a heat sink to protect an ARM-based computer system operating in outdoor high temperature environments, shown in Figure 11.9. This design can effectively reduce the temperature of the processor and ensure the continuous operation of the computer system in high-temperature environments.

11.3 Unmanned System Software

A generic framework of unmanned system software is presented in Figure 11.10. In this framework, the functions of all necessary UAV modules, including the onboard system and ground control system are clearly presented. The logical data flows among different modules facilitate the design of UAV systems.

11.3.1 Onboard Real-Time Software System

The main functions of the onboard real-time software system are to (a) collect sensor data, (b) process and fuse them, (c) feed the control law, and (d) send the servo driving signals to realize desired automatic and intelligent operations. Meanwhile, the UAV status data is transmitted back to the GCS and exchanged among team members. The UAV onboard software system has the functions as follows.

1. Sensing and measurement: The sensing data may come from different sources, such as the IMU, GPS, ultrasonic, laser scanner, or vision system, depending on the configuration of the avionics. All the sensing data are fed into the data selection

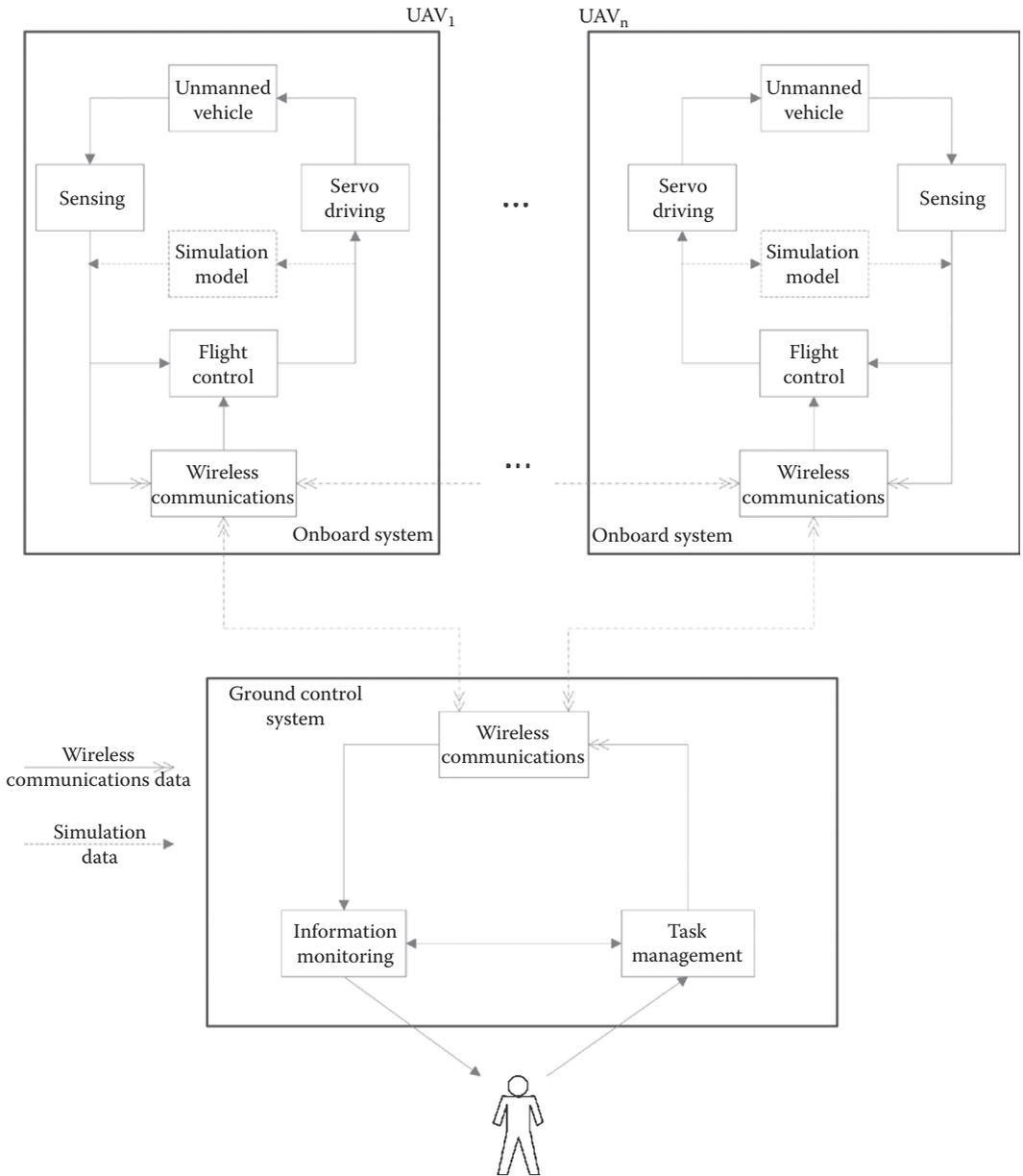


FIGURE 11.10
Framework of UAV systems.

and the processing unit, which will be selected and combined for the flight control and other mission algorithms.

2. Flight control system: The flight control module consists of three main units: task scheduling, outer-loop control, and inner-loop control. The task scheduling is to generate the outer-loop references given current status data and user commands from wireless communications. Based on the reference signals, the outer-loop realizes position, velocity, and heading control by generating references for the inner loop. The role of the inner loop is to stabilize the attitude of the UAV.
3. Servo driving: During the control procedure, the outputs are sent to the servo-driving module to drive the actuators on the UAV.
4. Communication: The communication block is to transfer the status data and commands between the vehicle and the GCS or between vehicles. The status data incorporates the flight information, such as position, velocity, and attitude. The ground operator can issue predefined commands, such as takeoff, hover, cruise flight, and landing to UAVs to perform specific tasks. The onboard video stream is also needed to feed back to the GCS. In addition, the inter-vehicle communication is applied for cooperative data exchange to achieve UAV team cooperative control, such as formation flight control.
5. Data logging: The key data, such as sensor measurements and control signals, in the flight will be logged for further analysis. The data logging is usually designed as a background task to minimize the interference to other critical tasks.
6. Emergency function: The emergency function is learned from a couple of crash incidents of our UAV helicopters. There are many causes for UAV failures, such as drastic changes in environment or hardware and software failure. To handle such emergency situations, the control task thread checks all sending data at every cycle before applying control actions. Once an abnormality is detected, the emergency control function will be activated immediately to send an alert signal to the GCS to inform the pilot to take over the control authority, drive and maintain all the control outputs to their trimmed values in the hovering condition, and slow down the engine or motor speed if the control authority is still at the automatic side after a predefined alert time. The flight data will also be recorded, which is important for fault analysis.
7. Simulation model: The simulation model block is an offline module used to conduct hardware-in-the-loop simulation before flight tests, which can generate the dynamics of the UAV based on a mathematic model. The UAV model can be formulated as an order ordinary differential equation (ODE) as below:

$$\dot{x} = f(t, u, v, x), \quad (11.1)$$

where x represents the UAV model output with multiple states, t is the current system time, u is the current control signal input with multiple channels, and v is the wind disturbance in three directions. For the ODE implementation, the classical Runge-Kutta approximation method is applied in the software.

8. Main function: To manage all above tasks and functions with predefined priorities and properly assigned time slots.

Although it is possible to implement a software system using multiple nested loops to handle the above tasks in real time without an operating system, such a software system is not scalable and reliable for complex missions. The development of avionic software systems is preferred to be carried out in a real-time operating system (RTOS) environment, which can guarantee the final system is executed in deterministic behavior, based on certain scheduling, intertask communications, resource sharing, interrupt handling, and memory allocation algorithms [58]. Currently, the three most popular real-time operating systems adopted in the UAV development are the QNX Neutrino [59], VxWorks [60], and RTLinux [61]. In our current software system, the QNX is adopted for the mission-critical control purposes to meet strict timing deadlines, and the Linux is adopted for computation intensive mission planning tasks.

11.3.2 Ground Control Software System

The main responsibility of the ground control software system is to establish effective communications between the avionic system and the ground users and pilots. To fulfill this aim, the ground station is generally required to have the following fundamental capabilities: (a) displaying and monitoring the inflight status; (b) displaying images captured by the onboard system; (c) generating and updating flight trajectories; (d) sending control commands to the avionic system; (e) facilitating the ground piloted control or automatic control, especially in sudden occurrences of urgent events, such as emergency landing and cruise; and (f) logging inflight data. Other features, such as displaying the reconstruction of the actual flight status in a 3-D virtual environment, can be very helpful to the ground users when the UAV is flying out of sight (see, for example, [62]).

Compared with its avionic counterpart, the real-time feature for the ground station software system is preferable but not strictly compulsory. As such, many ground station software systems, particularly for scientific research and commercial purposes, are not developed under a RTOS environment. Instead, other powerful programming environments with rich interface capacities, such as Windows-based Visual C++ [63], are commonly adopted.

11.4 Case I: Design of a Coaxial Rotorcraft System

In the following section, we will detail the mechatronics design of unmanned aircraft systems using a couple of examples. The first example is a fully functional unmanned rotorcraft system: GremLion [64]. GremLion is a new small-scale UAV concept using two contra-rotating rotors and one cyclic swashplate. It can fit within a rucksack and be carried easily by a single person. GremLion is developed with all necessary avionics and a ground control station. It has been employed to participate in the 2012 UAVForge competition.

11.4.1 Hardware System

GremLion, shown in Figure 11.11, features a coaxial design driven by two contra-rotating rotors that can compensate the torque due to aerodynamic drag. Such a design allows for a more stable, more maneuverable, quieter, and safer helicopter due to the inclusion of a coaxial main rotor and exclusion of a tail rotor, which results in a smaller footprint.



FIGURE 11.11
The GremLion UAV.

Coaxial helicopters also provide a better thrust-to-weight ratio than traditional helicopters, produce greater lift, and are also much more efficient [65]. Therefore, this platform is suited for the size requirement of the competition, which is required to be kept in a rucksack. The key specifications of the platform are listed in Table 11.6.

To reduce the complexity of the actuation system of the conventional coaxial design, a novel actuation system has been employed in GremLion, which is shown in Figure 11.12. The operating principle of this actuation system is presented as follows:

1. Unlike conventional single-rotor helicopters that utilize the collective pitch of their rotor blades to adjust the lift force, GremLion’s rotor pitches are fixed, and the thrust variation is accomplished by changing the rotor spinning speed simultaneously. Hence, the vertical motion is controlled by the pulse width modulation (PWM) signals fed to the motors attached to the top and bottom rotors. As illustrated in Figure 11.12, the throttle input δ_{col} can adjust the speed of the upper rotor and the lower rotor simultaneously.
2. The helicopter yaw motion (head turning) is produced by the difference of spinning speed between the top and bottom rotors. When one rotor spins, other than the lifting force it creates, it also generates a rotational torque on the fuselage of the helicopter in the direction opposite to the rotor spinning direction. Note that the top and bottom rotors always spin in opposite directions so that their torques

TABLE 11.6
Specifications of the GremLion UAV

Specifications	GremLion
Upper rotor span	798 mm
Lower rotor span	895 mm
Upper rotor speed	1900 rpm
Lower rotor speed	1700 rpm
No-load weight	2.4 kg
Maximum takeoff weight	5.1 kg
Power source	LiPo battery
Flight endurance	15 min

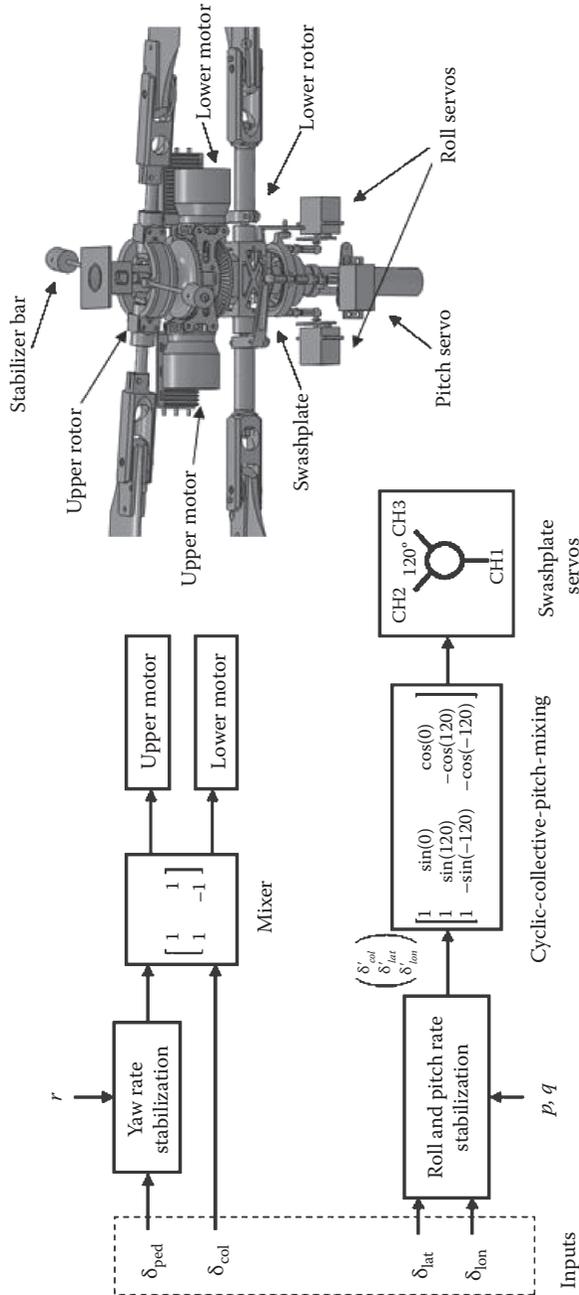


FIGURE 11.12 Operating principle of Gremlion.

cancel each other. In order to make the heading of the helicopter stable, a hardware rate gyro is installed to finely adjust the spinning speed of the two rotors so that yaw dynamics become much more damped. As shown in Figure 11.12, the rudder input δ_{ped} for control of the yaw of the vehicle differentiates the spinning speeds of the two rotors.

3. In order to have lateral and longitudinal motions, the bottom rotor cyclic pitch is actively controlled by three servos. This is done through a swashplate mechanism, which acts as a link between the servos and the bottom rotor cyclic pitch. As shown in Figure 11.12, the aileron input, δ_{lat} , controls the leftward and rightward tilting motion of the swashplate. Such a movement changes the cyclic pitch angle of the lower rotor blades and results in both a rolling motion and lateral translation. The elevator input, δ_{lon} , is responsible for the forward and backward tilting motion of the swashplate. This tilting also changes the cyclic pitch angle of the lower rotor blades but results in pitching motion and longitudinal translation. The aileron and elevator inputs cooperate with the roll and pitch rate feedback controller to stabilize the angular rate of roll and pitch motions. Such a rate feedback controller is used to allow a human pilot to control the oversensitive dynamics of the bare platform.
4. The upper rotor is equipped with a stabilizer bar to further increase the stability of the vehicle. The top rotor is not actively linked to any servos, but it is passively controlled via a mechanical stabilizer bar. With the presence of this stabilizer bar, the top rotor always has a cyclic pitch (with respect to the body frame) countering the inclination of the fuselage at any single moment. This slows down the whole platform's response to the rapid changes in the cyclic pitch of the bottom rotor [66]. In this way, the helicopter stability is increased, but the maneuverability is decreased.

11.4.1.1 Navigation Sensors

IG-500N (see Figure 11.13) is one of the world's smallest GPS-enhanced AHRS embedded with an extended Kalman filter (EKF). It includes a MEMS-based IMU, a GPS receiver, and a pressure sensor. It is able to provide precise and drift-free 3-D orientation and position even during aggressive maneuvers, updated at 100 Hz. Its key specifications are summarized in Table 11.7.

11.4.1.2 Computers

The onboard processor is the brain of the whole avionic system. It collects measurement data from various sensors, performs sensor filtering and fusion, executes flight control law, and outputs control signals to carry out the desired control actions. In addition, it is also responsible for communicating with the GCS for real-time inspection and command issuing as well as logging in-flight data for post-flight analysis. Hence, selecting suitable COTS processors is crucial to ensure successful implementation of the UAV system. We have chosen two Gumstix Overo Fire embedded computers for flight control and navigation purposes, respectively (see Figure 11.13). This embedded computer system has a main processor running at 720 MHz and a DSP coprocessor. The main processor is an OMAP3530 ARM processor from Texas Instruments, and it is one of the fastest low-power embedded processors as of this writing. Moreover, it has WiFi functionality despite its tiny size

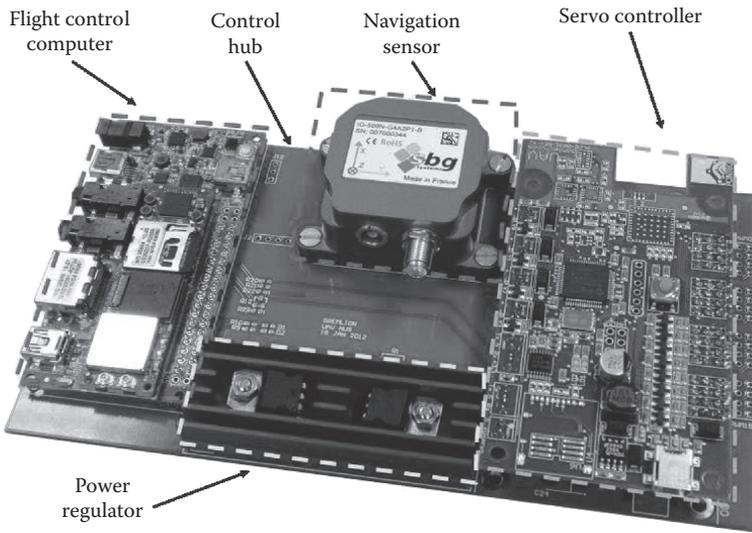


FIGURE 11.13
Control hub with all the modules.

TABLE 11.7

Main Specifications of IG-500N

Specifications	IG-500N
Attitude range	360° in three axes
Attitude accuracy	±0.5 (pitch, roll), ±1 (heading)
Accelerometer range	±5 g
Gyroscope range	±300
Magnetometer range	±1.2 Gauss
GPS accuracy in CEP	2.5 m (horizontal), 5 m (vertical)
Output rate (Hz)	{1, 25, 50, 75, 100} selectable
Dimensions	36 × 49 × 22 mm
Weight	46 g (with aluminum enclosure)
Power consumption	550 mW @ 5.0 V

and light weight. In order to improve its real-time performance, the original Linux operating system provided by the manufacturer is replaced by the QNX Neutrino. Custom-built autopilot software developed by the NUS UAV Research Group is used to realize the desired autonomous flight control.

11.4.1.3 Servo Controller

An eight-channel PWM servo controller, UAV100 (see Figure 11.13), is adopted to allow servo outputs to be controlled by an onboard computer or control command from the RC receiver, depending on the state of a switching signal from the RC transmitter. Although GremLion maneuvers autonomously in the air, it is desirable to have a failsafe feature to allow the ground pilot to take over control during emergencies. This servo controller

allows the pilot control to take over the control of a UAV at the flick of a switch of the transmitter to prevent a catastrophic incident from a malfunction in the flight computer. This function gives us the ability to test flight control software early without fear of damaging a test vehicle.

11.4.1.4 Communication

The communication unit includes a pair of Microhard wireless data transceivers. This pair of transceivers establishes communication between the onboard system and the ground station. They are configured to operate in a point-to-point mode and work in 2.400 to 2.4835 GHz. The transceiver used in the onboard system is set as a point-to-point slave and connected to the flight control computer board. The transceiver in the ground station is set as a point-to-point master and connected to a laptop.

In addition to the RF transmission, another data transmission mechanism is based on the 3G network that possesses continental distance (around 10 s of km) and high data bandwidth (100~300 kpbs). To facilitate the image transmission, the images are compressed via a JPEG compression method provided by OpenCV libraries. The size of the compressed image size is about 3K bytes that can be reasonably accommodated by the 3G network bandwidth. The update frequency is achieved as 2 Hz. The 3G module adopted is the UC864-E from Telit Company, which is well compatible with the Gumstix board. The communication protocol for image transmission is selected as TCP that can guarantee reliable image transmission although some delays can be expected for the handshaking and image retransmission.

11.4.1.5 Control Hub

The control hub, shown in Figure 11.13, is a motherboard designed to host subsystems for control purposes. It has the following features:

1. Module connection. Aforementioned modules, such as the Gumstix board, the IG-500N, and the UAV100 servo control board, are installed on the slots on the control hub and connected to the onboard power regulator and other essential components through the control hub. In addition to the mounting slots, extra mounting holes on the control hub have been used to lock the installed modules to resist the vibration and shock in flight and landing. Manual wire wrap has been minimized to improve the reliability and quality of the system.
2. Level shifter. An onboard level shifter, MAX3232, has been built in the control hub to convert the serial signal from RS-232 level to TTL level, which has been used to make the output of IG-500N compatible with the Gumstix board.
3. Power regulation. To power up all the avionics, linear regulators are built in the control hub to convert a power input from a three-cell LiPo battery into a 5 V output with 10 A capacity and a 2–8 V adjustable output with 10 A capacity. The 5 V output powers the Gumstix board, the rate gyro, and the electronic mixer. The adjustable output powers the servos.

11.4.1.6 Hardware Integration

The final integrated platform is shown in Figure 11.11. In addition to the essential mechanical parts, all the related avionic components have been assembled onto the system. This

platform has been extensively used in test flights for model identification and verification. Layout design for onboard computer systems is a challenging issue for small-scale UAVs. In what follows, we propose a simple and unified layout design approach, which is independent of the hardware components used and can be easily adopted to construct any small-scale UAVs.

1. Location of the navigation sensor. The essential rule of this step is to mount the navigation sensor IG-500N with the control hub as close as possible to the center of gravity (CG) of the UAV to minimize the so-called lever effect, which can cause bias on the measured accelerations when the UAV performs rotatory motions. Based on the experiences we gained from the construction of our earlier version UAVs [53], we find that it is easier to control the UAV when the onboard system is mounted underneath the bare vehicle. For such a layout, the general guideline is to line up the CGs of the INS/GPS, the onboard computer system and the basic helicopter along the z-axis of the body frame. Because the CG location of the bare vehicle is fully known using the pendulum test introduced in [67], the mounting location of the navigation sensor in the x-y plane of the body frame can be determined. The offset between the CG of the UAV and that of the navigation sensor is only in the z-axis and is unavoidable. However, it can be minimized by carefully considering the height of the onboard system and adding necessary space between the bare helicopter and the onboard system for bumping avoidance. In addition, the GPS antenna of IG-500N has been located in the top of the main shaft of GremLion to have a good view of the sky in order to obtain a stable signal lock.
2. CG balancing. The locations of the following four components, that is, the vision hub, the onboard camera, the wireless modem, and the battery packs, have to be carefully selected. In general, the camera and the wireless modem are to be mounted at the front part for the convenience of observation and wireless communication. The vision hub is placed on the back to balance the overall CG of the onboard system. The battery packs are placed beneath the fuselage and along the z-axis of the body frame. Furthermore, we also guarantee that the CG of the onboard system coincides with the CG of the INS/GPS, and the onboard system is symmetrical in both longitudinal and lateral directions.

Antivibration for the platform is a key issue that affects performance of the system significantly. The main vibration sources in GremLion come from the two main rotors with the frequency of 33.3 Hz. This frequency is calculated based on the designed main rotor speed at 2000 rpm, which was also verified using a handheld tachometer.

Several solutions have been employed for antivibration purposes: (a) use four wire-ropes isolators mounted symmetrically around the CG of the avionic system; (b) employ a redesigned landing skid that has better connections to the platform; (c) replace wooden blades with carbon blades, which have the same airfoil profile and size but with a smooth surface; and (d) configure the cutoff frequencies of the built-in low-pass filters of the IMU at 10 Hz. These proposed antivibration solutions have been demonstrated in flight. The comparison of the vibrations before and after employing the antivibration is shown in Figure 11.14.

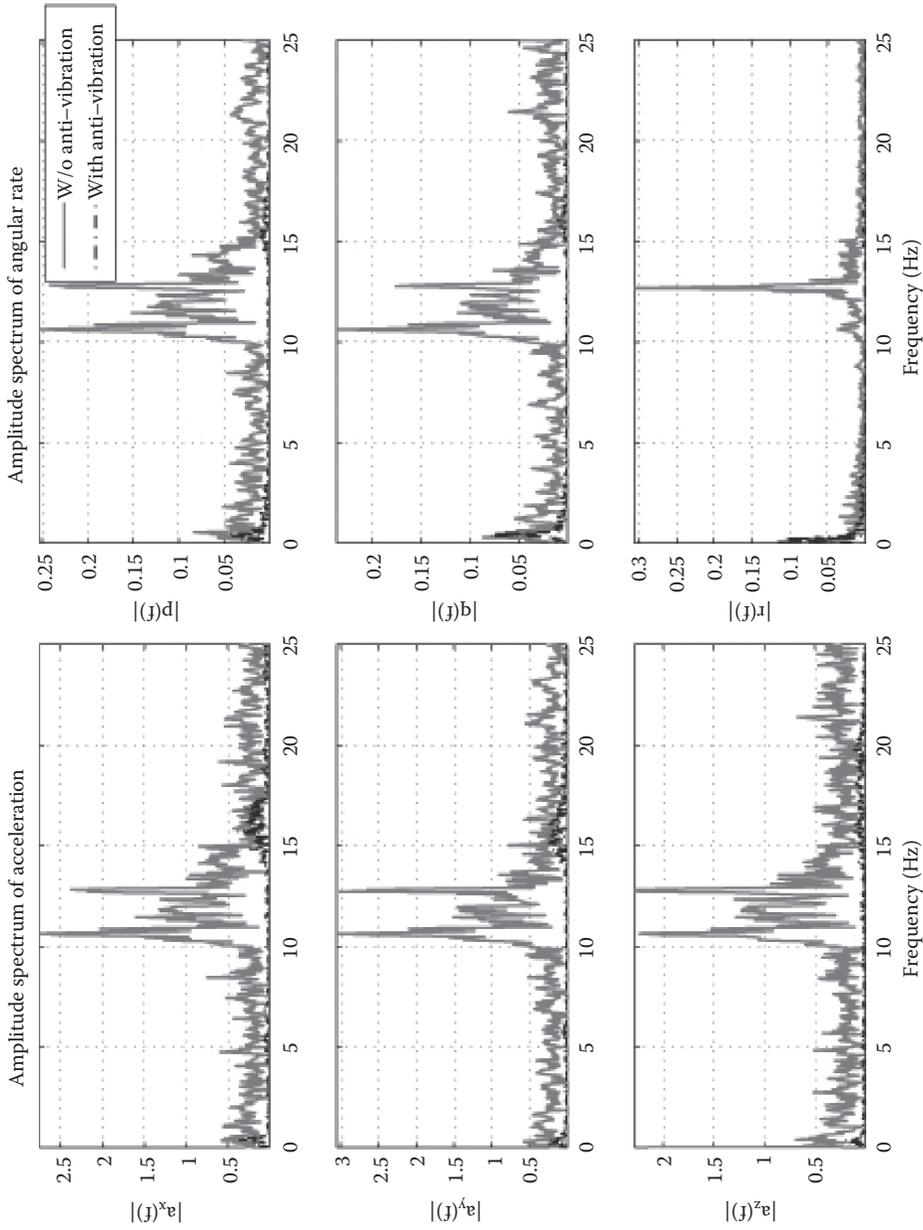


FIGURE 11.14 Comparison of frequency response of the acceleration and angular rate with antivibration solutions.

11.4.2 Software System

11.4.2.1 Onboard Real-Time Software System

The control law implementation is realized via two hierarchical blocks, the inner loop and outer loop as shown in Figure 11.15. The outer loop is to generate references for the inner loop as the input. The flight-scheduling module is to divide a whole flight mission into several specified tasks such as takeoff, path tracking, and landing. The flight mission can be as simple as conducting an automatic hover operation or can be as complicated as surveillance of a group of UAVs. With this hierarchical approach, various high-level missions can be transferred into logical representation and practical implementations. Specifically, for different UAV platforms, the corresponding blocks of the outer loop and the inner loop are activated. In addition, given different control behaviors, such as landing and takeoff, specific control blocks are also developed.

In some critical applications, the reference paths for a UAV must be generated online. This is commonly needed in dynamic environments, such as in the cases of a lost link or waypoint updates from the ground pilot. There are basically two parts: one is the path creation given a certain task, and the other is the outer-loop reference generation from the generated path. In the path creation, the task needs to provide the destination waypoint, which when given the current position and heading, can automatically generate the new path with a specified tracking velocity. The destination waypoint can be uploaded from the Google map with the GPS information or by a user-defined relative distance to the launch position.

The servo-driving block is to control the deflections of actuators according to the control outputs. The servo-driving block has two inputs. One is the manual control signal from the ground pilot, and another input is the automatic control signal generated by the flight control block. One of the two inputs can be selected to output in terms of the switch signal from the manual input. If the manual control is enabled, then the manual input signal will be translated and stored in the data store. Otherwise, the automatic signal will be translated, stored, and finally sent to the servos to realize desired deflection positions. The data logging block is to record the key avionics data in flights, including UAV status, manual and automatic servo signals, user commands, and so on. The recorded data can reflect the working status and property of the whole system and are useful for off-line analysis. Considering the CPU load increase when writing data to the onboard CF card, the data logging thread is executed every 50 cycles, that is, once per second. The above discussed subsystem is represented as an individual task in the whole onboard software system. Based on the theory of systems and control, the above tasks should be executed in

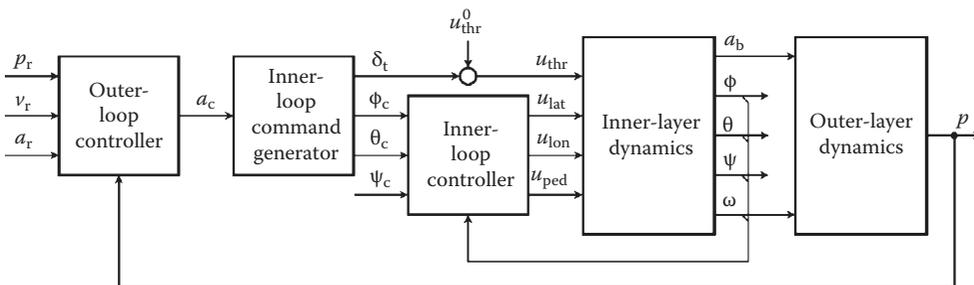


FIGURE 11.15 Dual-loop structure of flight control system.

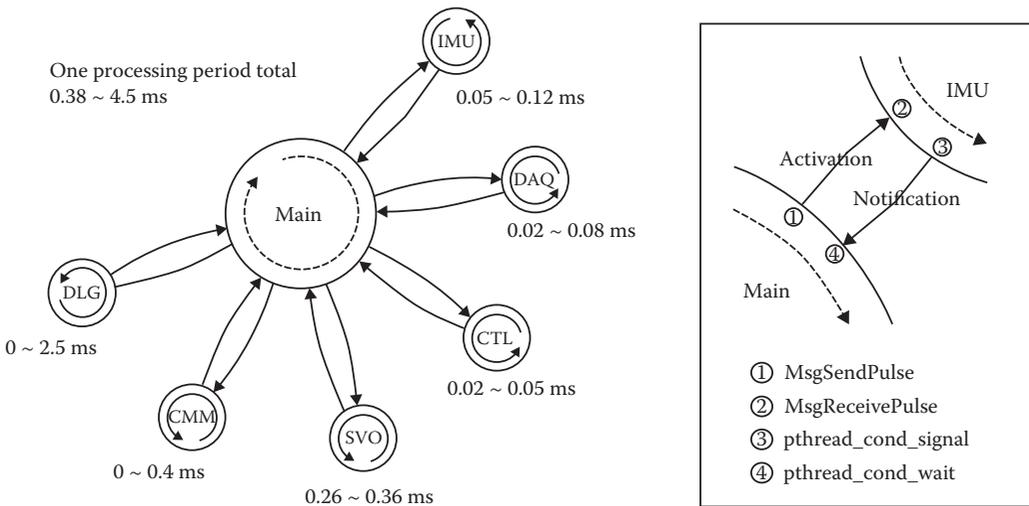


FIGURE 11.16
Task management illustration diagram.

a reasonable order to fulfill the automatic flight control purpose. Considering the context-switching cost, the onboard avionics application is designed in a multithread fashion. Therefore, the above identified task can be accomplished within each working thread. The QNX provides the kernel level mechanism to support the message passing and synchronization of multithread software architecture.

To realize the predefined task execution and synchronization, a task management module is carefully designed, which is shown in Figure 11.16. Note that the tasks involved here are the active tasks that should be executed within every control loop. To achieve fair allocation of processor running time, the round robin scheduling policy is adopted in the multithread design. On the other hand, the background tasks, such as communication receiving (UART-based and TCP/IP-based) are activated once data arrive and if there are still CPU running slices left and the activation mechanism is determined by the scheduling policy in QNX RTOS.

As shown in Figure 11.16, the main program is responsible for activating the next task once the notification message is received from the current running task.

The time deadline is the most important property in real-time systems. As the main control period of the onboard system is set to 20 ms, the timing intervals between each running loop should be examined to test the robustness and efficiency of the avionics system framework. The total processing time for each loop is summarized in Figure 11.17. It can be observed that the timing interval between each loop is separated by strictly following the 20 ms configuration, which provides the fundamental support for correct and stable control law implementation.

11.4.2.2 Ground Control Software System

The GCS is composed of background tasks and foreground tasks. The background layer has mainly two tasks, receiving flight status from and sending commands to UAVs, both of which interact with the UAV onboard CMM task module. The receiving thread

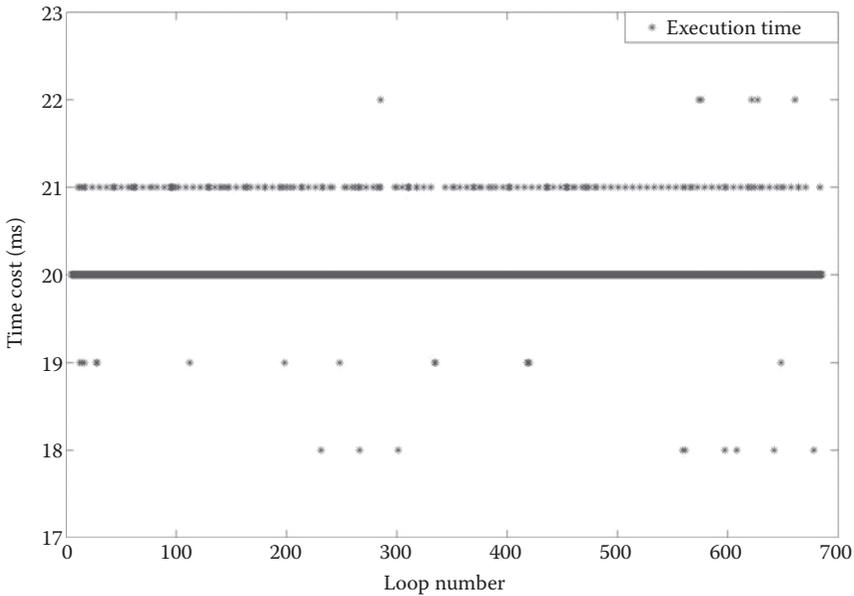


FIGURE 11.17
Time intervals between each loop.

accepts all the data from the fleet of UAVs, and identifies each status data via the telegraph packet header. Consequently, the corresponding multiple display is executed, and the cooperative waypoints of the paths are demonstrated. Similarly, the upload link can broadcast the commands to all UAVs or alternatively send commands to a specific UAV, both via the sending task. The global status data from UAVs are dynamically updated from the background layer. The foreground task is composed of information monitoring and task management, and the information monitoring module consists of various user-friendly views.

A document class implementation in MFC [68] is deployed to realize the communication between the background tasks and foreground tasks. The document class performs the flight data store (up to 2000 updates), data processing (rotation computation in 3-D view), command interpreting and packaging, etc.

Five kinds of views are developed on GCS, including the map view, the curve view, the text view, the command view, and 3-D view, which are all shown in Figure 11.18. To facilitate navigation and better demonstrate the flight trajectories of UAVs, we downloaded from the Internet the map from the Google map server and used the map tiles offline on the flight field conveniently without bothering the Internet service [69]. In the flight test, the GPS data from the onboard system will be updated on the global shared data, and the flight trajectories are drawn correspondingly on the Google map view.

11.4.3 Experimental Results

To verify the developed UAV systems, automatic flight tests have been conducted. Figure 11.19 shows clearly the reference generation onboard with a total of four trapezoids, which represent the four waypoints uploaded by the GCS operator on the Google map. Between every two waypoints, the longitudinal and lateral velocity reference signals increase and



FIGURE 11.18
Snapshot of GCS layout.

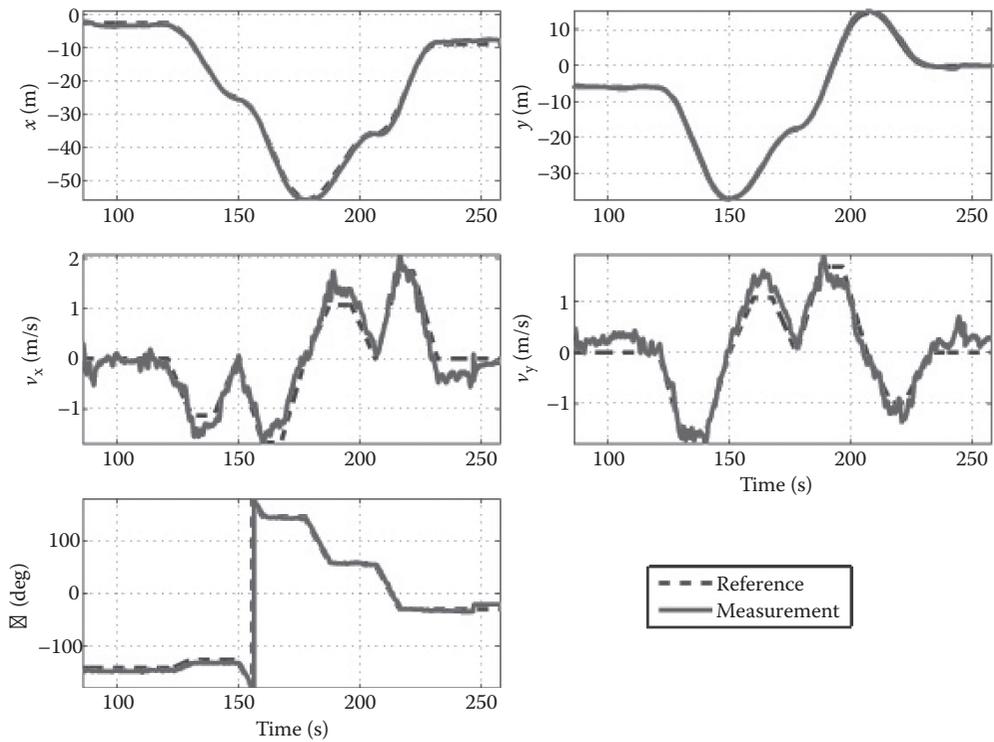


FIGURE 11.19
Experimental results of the automatic flight.

decrease smoothly and the corresponding position reference is also generated correctly. In the flight test, the UAV can automatically track the reference very well.

11.5 Case II: Design of a UAV Cargo Transportation System

In this part, we present a comprehensive UAV cargo transportation system, which involves an innovative cargo grabbing system, a set of UAV autonomous navigation and control algorithms, and a vision-based intelligent cargo searching and guidance method. The proposed UAV system, codenamed NUS²T-Lion (see Figure 11.20) was implemented and had taken part in the second AVIC Cup–International UAV Innovation Grand Prix (UAVGP), which was held in Beijing in September 2013. In this competition, NUS²T-Lion demonstrated its ability of searching and locating cargo on one moving platform and automatically grabbing and transporting the cargo to another moving platform. The whole process from taking off to landing was done without any decision making from the ground operator.

11.5.1 Hardware System

The hardware configuration of NUS²T-Lion follows the universal rotorcraft UAV structure proposed in [1]. As illustrated in Figure 11.21, in which each block represents an individual hardware device, the whole system is constituted by four main parts, namely a bare rotorcraft platform, an onboard avionic system, a manual control system, and a ground control system.

Although the manual control system and the GCS are rather standard for all kinds of UAV systems, the choice of the bare rotorcraft platform and its onboard avionic system is usually application-dependent. For this case, they should be properly chosen and integrated for the UAV cargo transportation task. It is believed that by designing the hardware configuration creatively, difficulties for the later software algorithm development can be minimized.

To realize fully autonomous flight, an onboard avionic system with sensors, processors, and other electronic boards has to be designed. All components used on the NUS²T-Lion



FIGURE 11.20
NUS²T-Lion with cargo grabbing capability.

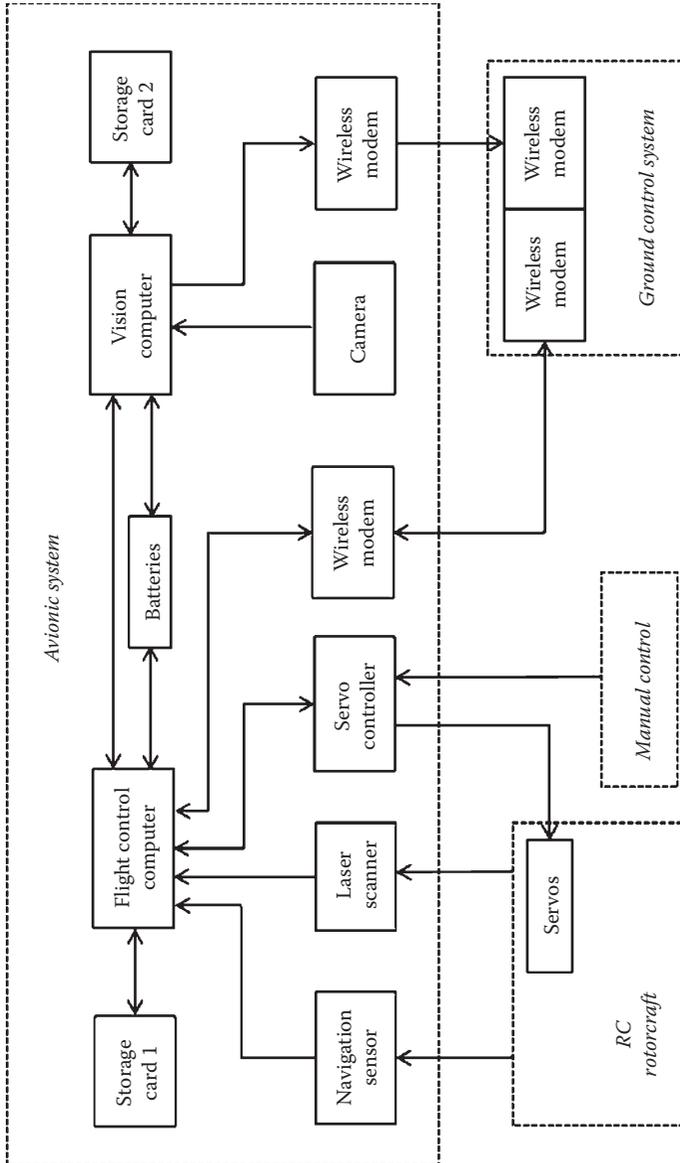


FIGURE 11.21 Hardware configuration of NUS-T-Lion rotorcraft system.

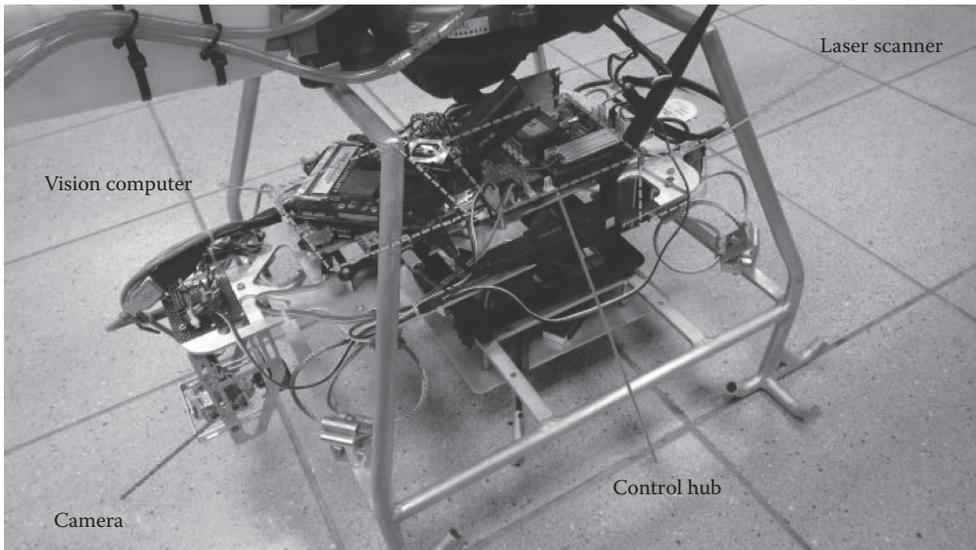


FIGURE 11.22

Onboard avionics system of NUS²T-Lion.

are carefully chosen COTS products that are up to date. Figure 11.22 gives a complete view of the onboard system with the key components indicated. The details and usage of these components are explained as follows.

11.5.1.1 Grabbing Mechanism

For the bare rotorcraft platform, the Thunder Tiger Raptor 90 SE Nitro RC helicopter is adopted in this work. It is a hobby-level single rotor helicopter originally designed for acrobatic flight. As compared with other COTS RC rotorcrafts, such as Turbulence D3 and Observer Twin, Raptor 90 SE provides a reliable structural design and equivalent flight performance at approximately half the price.

However, with the original Raptor 90's nitro engine and nitro fuel tank, the endurance of the UAV can barely reach 8 min with full load avionics. This is not sufficient for practical applications. To overcome this limitation, the original nitro engine is replaced by a gasoline counterpart, which is a product from Zenoah with model number G270RC. With the more efficient gasoline engine, a full-tank Raptor 90 can fly up to 30 min. This greatly widens the range of potential applications for this UAV, and it is especially beneficial to the cargo transportation task.

Unfortunately, this endurance improvement comes with two tradeoffs. First, the vibration of the whole platform intensifies due to the gasoline engine. Second, the ignition magnet inside Zenoah G270RC is so large that its magnetic field can badly affect the onboard sensors. To overcome the vibration issue, wire rope isolators are used to protect the onboard avionics and filter out unwanted high-frequency noises. For the problem of magnetic interference, the final solution boils down to replacing the electromagnetic ignition system inside the engine with a pure electric ignition system. With this modification, the onboard sensors, especially the AHRS, all work as they should.

To cope with the cargo transportation task, there must be a loading mechanism installed on the helicopter. By comparing the solution of a rigid claw-like grabbing mechanism and

a long flexible rope hooking mechanism, the former is more precise in picking up the cargo, and the latter can avoid descending the UAV too low to the ship surface where the aerodynamic ground effect becomes an issue. In this work, an innovative design incorporating advantages from both sides has been proposed. The solution is a claw-like grabbing mechanism with very long arms (see Figure 11.23). With this design, the UAV can keep a safe distance from the ship's surface and, at the same time, grab and release the cargo in a precise and reliable way. Another highlight of this design is its omnidirectional feature, meaning no matter in which direction the cargo handle is oriented, it is not necessary for the UAV to adjust its heading to align accordingly. This saves time and minimizes unnecessary maneuvers, which may induce higher risks in autonomous flights. In addition, this design features a self-locking mechanism commonly used in landing gears of hobby-grade fixed-wing planes. The mechanism is enclosed in the rectangular boxes as shown in Figure 11.23 with each box supporting one arm and powered by one servo motor. When the claw fully opens or closes, there is a slider inside the box to lock the position of the servo motor. In this way, the servo motors consume zero power while carrying heavy cargo as the cargo weight is fully supported by the locking mechanism.

A load-sensing mechanism that can differentiate a successful cargo loading from a failure is also installed. This mechanism acts as a safeguard in cases in which the UAV fails to grab the cargo. By knowing that the cargo is not successfully loaded, the UAV can be commanded to descend and grab the cargo again. The detailed design is shown in Figure 11.24, in which four limit switches, which send out electrical signals when pushed down, are installed on the customized landing skid. The baseplate of the claw is rigidly attached to a hollow rectangular plate on its top. The rectangular plate is then resting on the cross-over beams of the landing skid via four springs. When the claw is loaded, the rectangular plate compresses the spring and triggers one or more of the limit switches. When the claw is unloaded, the springs push up the rectangular plate to release the limit switches.

In order to retain the UAV x- and y-axis CG balancing, the claw needs to be installed precisely under the UAV CG. In this way, the UAV roll and pitch dynamics will not change

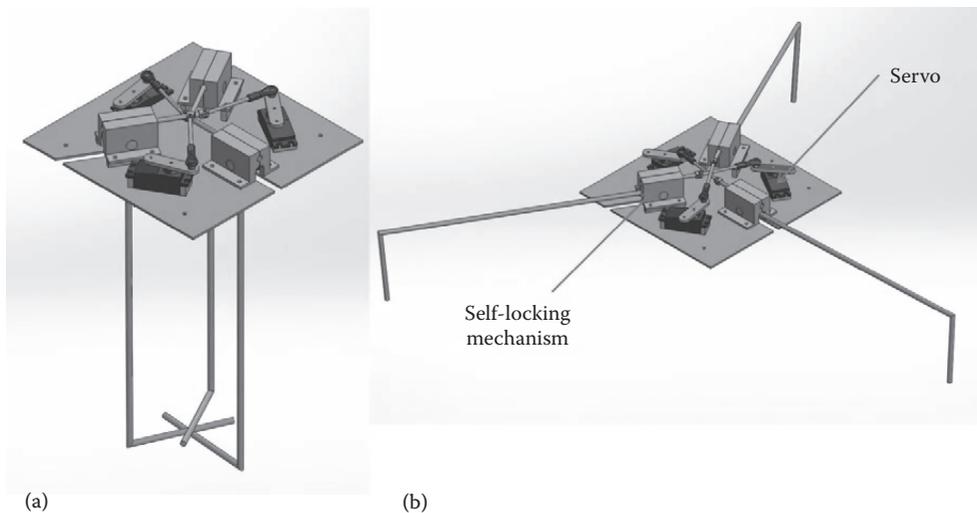


FIGURE 11.23 Grabbing mechanism in closed (a) and open (b) configurations.

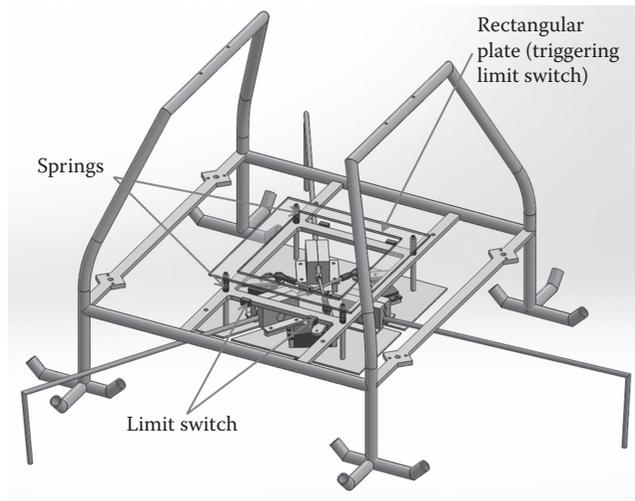


FIGURE 11.24
Grabbing mechanism with load-sensing function.

too much after cargo loading; thus the same set of control laws can be used. It also makes sure that controlling the UAV CG to the correct planar position is equivalent to controlling the claw to the correct position, which makes the problem easier.

11.5.1.2 Sensors and Measurement Systems

In addition to the fundamental navigation sensor: IG-500N, another main sensor used onboard the NUS²T-Lion is the mvBlueFOX camera from Matrix Vision. It is a compact industrial CMOS camera, compatible with any computers with USB ports. A superior image quality makes it suitable for both indoor and outdoor applications. Its high-speed USB interface guarantees easy integration without any additional interface board. In this specific cargo transportation application, it is the main guidance sensor for locating the cargo and their unloading points.

By considering the fact that the UAV usually flies forward to search for targets and hovers right above the cargo for loading and unloading, the best position to place the camera is at the nose of the helicopter. In addition, a controlled pan-tilt gimbal (see Figure 11.25) is designed to host the camera sensor so that it always looks vertically downward despite the UAV's rolling and pitching motions. Taking advantage of the camera's wide viewing angle, even when the UAV descends to the lowest altitude for cargo grabbing, the camera can still see the cargo without any problem.

For this cargo transportation application, height measurement from GPS/INS or the barometer is not accurate enough for the UAV to pick up or drop the cargo appropriately. The UAV may even crash on the surface of the cargo platform because of inaccurate height measurement, resulting in catastrophic consequences. Although a vision sensor or 1-D laser range finder may accomplish the task, the former can only be relied on when the visual target is within the field of view, and the latter cannot handle ground surfaces with scattered obstacles. To make the height measurement accurate and consistent, a scanning laser range finder is the ideal choice. The laser scanner code named URG-30LX from Hokuyo is installed in the system. It has a maximum range of 30 m with fine resolution of 50 mm, and it can scan its frontal 270° fan-shaped area with a resolution of 0.25°.

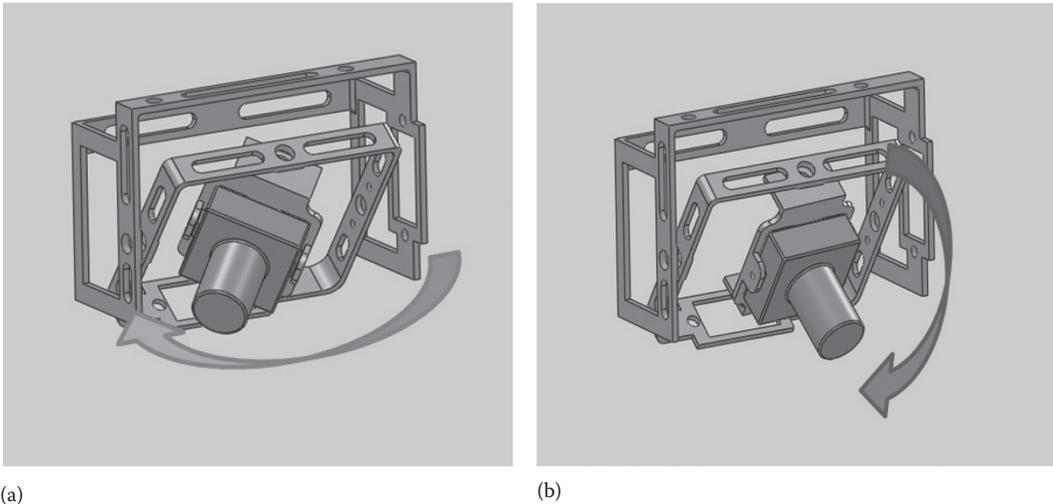


FIGURE 11.25
Pan-tilt mechanism of the camera: (a) camera pan, (b) camera tilt.

11.5.1.3 Computers

There are two onboard computers in the avionic system: one for the implementation of navigation and control algorithms and the other, more powerful, one dedicated for vision processing. With this dual-computer structure, the vision algorithm can be implemented and tested separately at the development stage, and it is very convenient to upgrade to a more powerful vision computer in the future without modifying the control hardware and software system. It also improves the reliability of the overall system because this structure ensures control stability even when the vision computer malfunctions or encounters run-time errors (it happens more frequently on the vision computer compared to the control counterpart because the vision algorithm usually involves more sophisticated calculations and logics). If it ever happens, the UAV should still fly safely with the control computer alone, and there will be enough time for the human pilot to take over and land the UAV safely.

For the onboard control computer, it collects measurement data from various sensors, performs sensor filtering and fusion, executes flight control law, and outputs control signals to carry out the desired control actions. In addition, it is also responsible for communicating with the GCS as well as data logging. To select a lightweight yet powerful embedded computer for these real-time tasks, the Gumstix Overo Fire embedded computer becomes the final choice.

For the onboard vision computer, it is mainly for implementing image processing algorithms, including color segmentation, object identification, object tracking, and localization. Image processing tasks are usually computationally intensive and hence require powerful processors to run the algorithms in real time. We have chosen the Mastermind computer from Ascending Technologies. It has an Intel Core i7 processor but is small and light enough to be carried by NUS²T-Lion. It also has abundant communication ports to interact with peripheral devices, such as USB cameras and WiFi devices. One UART port is used to communicate with the aforementioned control computer.

11.5.2 Software System

To implement the aforementioned GNC algorithm and to solve the logic problems in completing the UAVGP tasks as a unified process, a robust real-time software system needs to be developed. The following will show the key software concepts of the NUS²T-Lion, which includes the multiple-layer and multiple-thread software structure and mission logic in solving the UAVGP competition tasks.

The software structure of the NUS²T-Lion system is illustrated in Figure 11.26. It consists of three separate software programs, namely the onboard control software, the onboard vision software, and the GCS software. For the onboard control and vision software, they both utilize the multiple-thread framework so that time resources can be precisely distributed among different functional blocks (threads). It is also a more reliable way of implementing real-time UAV software so that the malfunction of an individual thread will not halt the executing of others.

The onboard control software is developed using QNX Neutrino, which provides reliable support for high-precision timer and synchronization operations. Multiple tasks (threads), including operating with hardware devices, such as the navigation sensor, the laser scanner, and the servo control board; implementing the automatic control algorithms; logging in-flight data; and communicating with the GCS and the vision computer, are managed by the MAIN function. With the multiple-thread framework, it is also easy to run different threads with different frequencies. More details about this onboard control software can be found in [62].

Similarly, the onboard vision software is also divided by multiple tasks, namely image capturing from the camera sensor, image processing, data logging, and communication with GCS and the control computer. The operating system utilized on the vision computer is the very popular Ubuntu Linux. It supports abundant hardware drivers, such as USB cameras and WiFi adapters, and software libraries, such as OpenCV, which are very suitable for the development of complicated vision algorithms.

For the GCS software, it runs on a laptop with a Windows 7 system. Such a commercial operating system provides strong support for the development of user interfaces. Visual C++ is employed to develop and debug the GCS software. By utilizing the MFC library, the global shared data are hosted in a document class, in which a variety of functions for data operating and visiting are integrated. Although this document class is the kernel of the software program, there are also the communication thread and other threads to control the multiple views at the foreground user interface. The communication thread receives and sends data to the UAV onboard system through a WiFi link, and the multiple displaying views periodically visit the contents of the document and update their respective displaying of new data received.

As the UAV cargo transportation application is mission-oriented, the sequence and logics of the whole task operation are rather important. It is implemented in the CTL thread of the onboard control software. The overall *Mission Logic* is illustrated in Figure 11.27. It consists of six sequential tasks, namely takeoff, navigate to ship, vision initialization, transporting cargo, return home, and landing. Because the mission is time-constrained, a timer interrupt is also implemented in the software. The timer interrupt will trigger the return home task once the predefined maximum mission duration runs out. The details of each task are discussed as follows:

1. The takeoff task will be triggered once the onboard software program receives the "Mission Start" command from the GCS. In this stage, the program lets the helicopter warm up its engine by issuing a small and constant value to the throttle

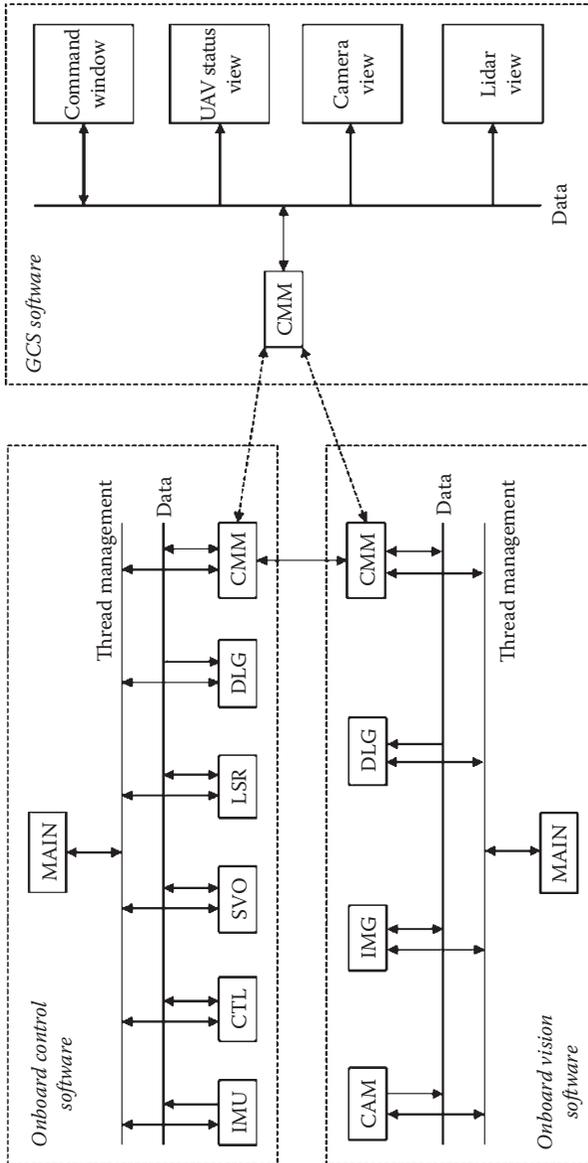


FIGURE 11.26 Software structure of NUS²-Lion. CAM, image capture from camera sensor; CMM, communication; CTL, control law implementation; DLG, data logging; IMG, image processing; IMU, measurement reading from GPS/INS; LSR, laser measurement and processing; MAIN, main program, task management; SVO, servo driving and reading.

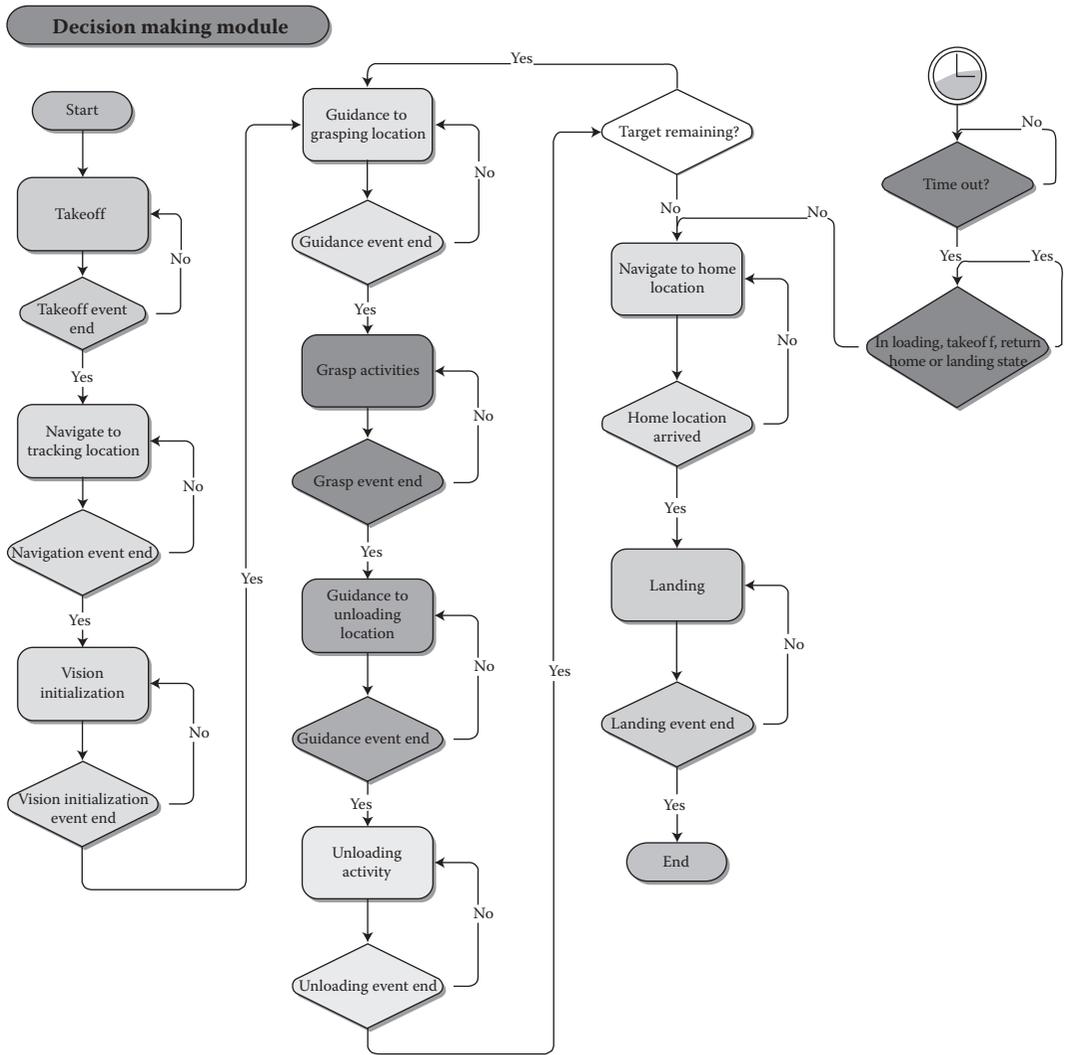


FIGURE 11.27
Mission logics.

channel. After a while, the throttle channel control signal will be increased gradually until the engine enters the governor mode (main blades will now be controlled at a predefined rotational speed of 1750 rpm). After that, the program will slowly increase the control signal of the collective pitch channel so that the lift force increases. Once the collective pitch signal hits its trimming value for the hovering condition, the program will ask the reference generation function to issue a “Going Up” trajectory. At the end of the trajectory, the program throws a “Takeoff Event End” signal.

2. The software program now enters the “Navigate to Ship” mode. In this stage, the program collects the position and velocity information from the GPS/INS system on the ship. A relative path to the ship with continuous relative position, relative velocity, and relative acceleration references will be generated. The flight controller

will continuously ask the helicopter to track this path so that the helicopter can catch up with the ship and have the same position and velocity profiles as the ship at steady state. Once the helicopter catches up with the ship, the software will throw a "Navigation Event End" signal. Note that this decision is made based on GPS/INS information. Physically the UAV may not be hovering so precisely above the center of the two ships.

3. In the "Vision Initialization," the vision system will first check whether it can detect two ships. If only one ship or part of a ship has been detected, the vision system will guide the helicopter to move toward one of the detected circles. In this way, there will be very high probability to see the other ship by taking advantage of the onboard camera's wide viewing angle. Once both ships are successfully detected, the software will be scheduled to the "Transporting Cargo" mode.
4. The "Transporting Cargo" task is the most sophisticated part of the mission. In this stage, the UAV will choose the cargo and fly to a position right above it. When the UAV horizontal position to the target cargo enters a small threshold, its height reference will be gradually adjusted down to an appropriate value so that the mechanical claw can grasp the bucket handle. Once the mechanical claw is closed, the UAV will be commanded to fly upward quickly so that the limit switch sensors mounted under the claw platform can sense whether the cargo weight has been successfully loaded. If it deduces that cargo has not been grasped successfully, the UAV will be commanded to go down again for another try. The above procedure will be repeated until the limit switch system detects a successful cargo loading. After that, the helicopter will be commanded to move to the unloading point. For the unloading task, the UAV has a similar procedure to check whether the cargo has been successfully released. If the detection is false, the UAV will quickly close and open its claw to try another release. For failsafe, when the vision system loses the cargo target for more than 10 sec during the grasping stage, the software will issue a "Going Up" command so that the vision system can have a wider view, which leads to higher chance of retrieving the target. Once the vision system retrieves the target, the UAV will be commanded to go down and try grasping the cargo again. There is a counter to record how many cargos remain to be transported. Once the counter hits zero, the program will jump out the current mode and enter the return home mode.
5. When the helicopter has finished all its transportation tasks or the maximum mission time runs out, the "Return Home" task will be triggered. The software will generate a reference trajectory ending at a predefined height and with the UAV's planar position equal to the initial takeoff point. The UAV will then follow this trajectory back to the home location.
6. The landing will be triggered as the helicopter flies right above its home location. The procedure for the landing task is similar to the takeoff task. The software asks the flight controller to regulate the helicopter moving downward with a constant speed at 0.5 m/s (if height is greater than 5 m) or 0.2 m/s (if height is less than 5 m). Once the UAV landing gear approaches the ground (within 8 cm), the control signal to the throttle channel will jump to a minimum value so that the engine shuts down.

11.5.3 Experimental Results

In preparation for the UAVGP competition, numerous flight tests have been carried out to verify the overall solution feasibility and to tune for the optimal performance. Figures

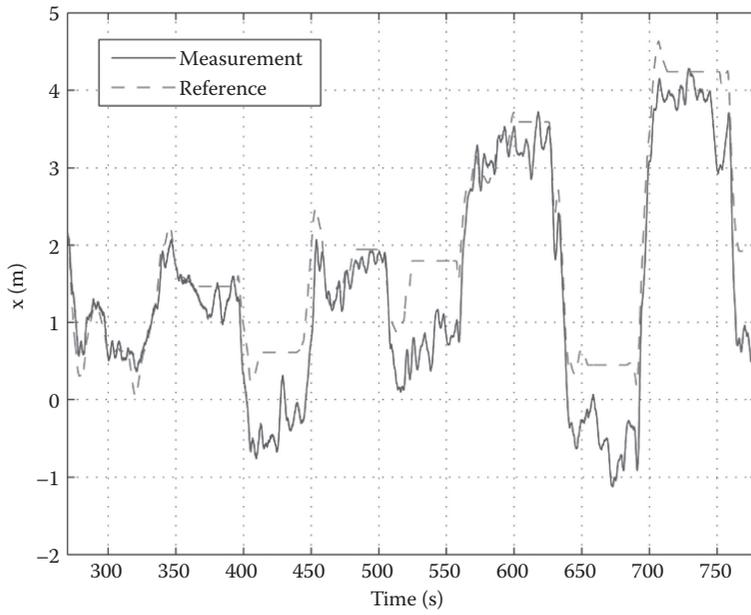


FIGURE 11.28
UAV position response in the ship frame x-axis.

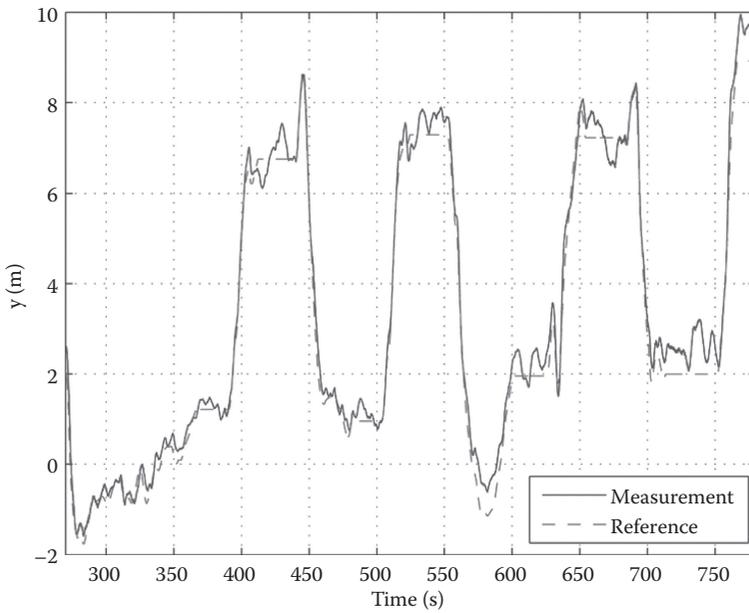


FIGURE 11.29
UAV position response in the ship-frame y-axis.

11.28 through 11.30 show the position data logged in one of the flight tests. As the raw data is obtained by GPS/INS and then converted to the ship frame, it may not be the exact truth. However, it generally shows the control performance and roughly indicates whether the UAV is doing the correct movement. In Figure 11.28, the x-signal becomes larger progressively because the UAV is moving from the first bucket to the fourth bucket. It always comes back to a position around zero because the reference path is defined in a way that the onboard camera has the best view of the two ships before every loading or unloading dive. In Figure 11.29, the y-position signal goes back and forth, indicating alternative movements between the two ships. In Figure 11.30, it is clear to see all the diving motions of the UAV. The UAV will stay at a very low altitude with variable duration depending on how many loading or unloading trials have been performed before the successful one.

With this kind of performance, the NUS²T-Lion has successfully accomplished the competition tasks in the UAVGP rotary-wing category. A final score of 1127.56 with 472.44 from the preliminary contest and 655.13 from the finals has pushed the team into second position in the overall Grand Prix. It should be highlighted that 655.13 was the highest score among the finalists. Moreover, unlike the preliminary contest, the final round of competition requires the UAV to carry out the cargo transportation task with the two ships moving. This demands better robustness and higher intelligence from the participants' UAV systems, and it is indeed the strongest point of the GNC solution proposed. The final competition has been recorded in video format and uploaded to [70] and [71] for the English and Chinese versions, respectively.

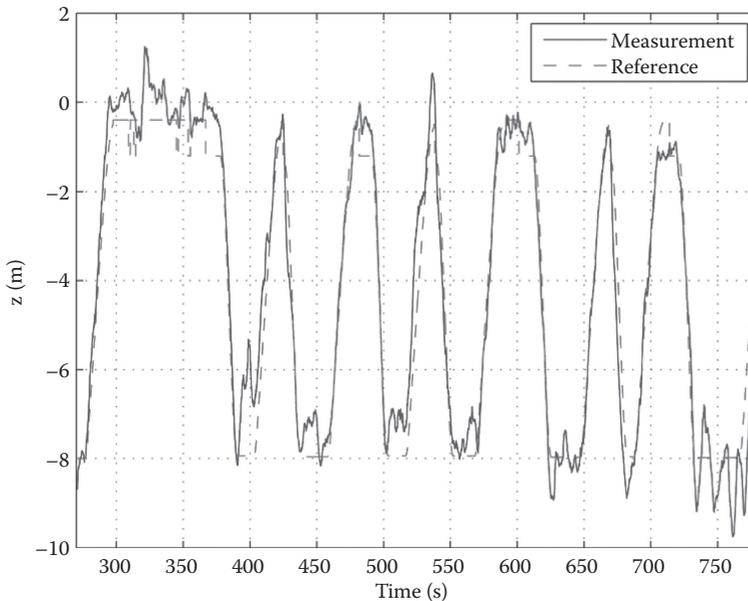


FIGURE 11.30
UAV position response in the NED-frame z-axis.

11.6 Conclusion

In this chapter, we have presented the mechatronics design of UAV systems, including hardware systems and software systems. The modular design approach and computer-aided hardware integration were highlighted in the hardware system construction. The multitask embedded software design has been presented in the software system development. After that, we addressed two case studies including a coaxial rotorcraft GremLion developed for the DARPA UAVForge challenge and a cargo transportation system used in UAVGP 2013. The experimental results show that the proposed methodology is efficient and effective. The constructed UAV systems can be used as excellent platforms for future research development.

In summary, system thinking is the critical part in mechatronics design of unmanned aircraft systems. How well we understand the mission requirements and the characteristics of possible components of the unmanned system plays a crucial role in mechatronics design. A balance between the system performance and the total cost in development, operation and maintenance needs to be achieved to fulfill a good mechatronics design. The proposed systematic design methodology will be able to facilitate in this process. Although we are faced with many obstacles and challenges in these research areas, component development and system integration, including hardware, software, and algorithms will continuously be improved to advanced levels. With the development in avionics, GCS, and mission algorithms, other unmanned systems will also be benefited by mechatronics design with systems thinking.

References

1. G. Cai, B. M. Chen, and T. H. Lee, *Unmanned Rotorcraft Systems*. New York: Springer, 2011.
2. Z. Sarris and S. Atlas, "Survey of UAV applications in civil markets," in *Proceedings of the 9th Mediterranean Conference on Control and Automation*, vol. WA2-A, Dubrovnik, Croatia, 2001, pp. 1–11.
3. B. Enderle, "Commercial applications of UAV's in Japanese agriculture," in *Proceedings of the AIAA 1st UAV Conference*, no. AIAA-2002-3400, Portsmouth, VA, May 20–23, 2002.
4. B. Ludington, E. Johnson, and G. Vachtsevanos, "Augmenting UAV autonomy," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 63–71, 2006.
5. M. E. Campbell and W. W. Whitacre, "Cooperative tracking using vision measurements on seascan UAVs," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 613–626, 2007.
6. J. Keller, D. Thakur, V. Dobrokhodov, K. Jones, M. Pivtoraiko, J. Gallier, I. Kaminer, and V. Kumar, "A computationally efficient approach to trajectory management for coordinated aerial surveillance," *Unmanned Systems*, vol. 1, pp. 59–74, 2013.
7. A. N. Kopeikin, S. S. Ponda, L. B. Johnson, and J. P. How, "Dynamic mission planning for communication control in multiple unmanned aircraft teams," *Unmanned Systems*, vol. 1, no. 1, pp. 41–58, 2013.
8. K. Kang and J. V. R. Prasad, "Development and flight test evaluations of an autonomous obstacle avoidance system for a rotary-wing UAV," *Unmanned Systems*, vol. 1, no. 1, pp. 3–19, 2013.
9. F. Lin, B. M. Chen, and T. H. Lee, "Robust vision-based target tracking control system for an unmanned helicopter using feature fusion," in *Proceedings of the 11th IAPR Conference on Machine Vision Applications*, Yokohama, Japan, 2009, pp. 398–401.
10. T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Synthesis of control protocols for autonomous systems," *Unmanned Systems*, vol. 1, pp. 75–119, 2013.

11. M. J. Adriaans, R. C. Cribbs, B. T. Ingram, and B. P. Gupta, "A160 Hummingbird unmanned air vehicle development program," in *Proceedings of the AHS International Specialists' Meeting—Unmanned Rotorcraft: Design, Control and Testing*, Chandler, AZ, 2007.
12. G. W. Cai, B. M. Chen, K. Peng, M. Dong, and T. H. Lee, "Modeling and control system design for a UAV helicopter," in *Proceedings of the 14th Mediterranean Conference on Control and Automation*, vol. WM1-4, Ancona, Italy, 2006, pp. 1–6.
13. V. Gavrillets, A. Shterenberg, M. A. Dahleh, and E. Feron, "Avionics system for a small unmanned helicopter performing aggressive maneuvers," in *Proceedings of the 19th Digital Avionics Systems Conferences*, Philadelphia, PA, 2000, pp. 1E2/1–1E2/7.
14. J. M. Roberts, P. Corke, and G. Buskey, "Low-cost flight control system for a small autonomous helicopter," in *Proceedings of the 2002 Australian Conference on Robotics and Automation*, Auckland, New Zealand, 2002, pp. 546–551.
15. K. Sprague, V. Gavrillets, D. Dugail, B. Mettler, E. Feron, and I. Martinos, "Design and applications of an avionic system for a miniature acrobatic helicopter," in *Proceedings of the 20th Digital Avionics Systems Conferences*, Daytona Beach, FL, 2001, pp. 3C5/1–3C5/10.
16. K. M. Peng, G. W. Cai, B. M. Chen, M. B. Dong, K. Y. Lum, and T. H. Lee, "Design and implementation of an autonomous flight control law for a UAV helicopter," *Automatica*, vol. 45, no. 10, pp. 2333–2338, 2009.
17. A. Karimoddini, H. Lin, B. M. Chen, and T. H. Lee, "Hybrid 3-d formation control for unmanned helicopters," *Automatica*, vol. 49, no. 2, pp. 424–433, 2013.
18. S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 20–25.
19. F. Wang, J. Q. Cui, B. M. Chen, and T. H. Lee, "A comprehensive UAV indoor navigation system based on vision optical flow and laser fastSLAM," *Acta Automatica Sinica*, vol. 39, no. 11, pp. 1889–1900, 2013.
20. J. Kim and S. Sukkarieh, "Autonomous airborne navigation in unknown terrain environments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 40, pp. 1031–1045, 2004.
21. H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I," *IEEE Robotics & Automation Magazine*, vol. 13, pp. 99–110, 2006.
22. J. H. Kim, J. Lyou, and H. K. Kwak, "Vision coupled GPS/INS scheme for helicopter navigation," *Journal of Mechanical Science and Technology*, vol. 24, no. 2, pp. 489–496, 2010.
23. A. Martinelli, "Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.
24. P. Doherty, F. Heintz, and J. Kvarnstrom, "High-level mission specification and planning for collaborative unmanned aircraft systems using delegation," *Unmanned Systems*, vol. 1, no. 1, pp. 75–119, 2013.
25. J. Hu, J. Xu, and L. Xie, "Cooperative search and exploration in robotic networks," *Unmanned Systems*, vol. 1, pp. 121–142, 2013.
26. "Unmanned systems integrated roadmap fy 2013–2038," Department of Defense, Tech. Rep., December 2013.
27. A. K. Kundu, *Aircraft Design*. The Edinburgh Building, Cambridge, UK: Cambridge University Press, 2010.
28. R. Austin, *Unmanned Aircraft Systems*. UK: Wiley, 2010.
29. A small list of IMU/INS/INU. [Online]. Available: <http://damien.douxchamps.net/research/imu/>.
30. P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008.
31. H. Chao, C. Coopmans, L. Di, and Y. Q. Chen, "A comparative evaluation of low-cost IMUs for unmanned autonomous systems," in *Proceedings of the 2010 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Salt Lake City, UT, September 2010, pp. 211–216.

32. B. Mettler, M. Tischler, and T. Kanade, "Circular error probable (CEP)," USA: Air Force Operational Test and Evaluation Center, Tech. Rep., 1987.
33. G. C. Guenther, A. G. Cunningham, P. E. LaRocque, and D. J. Reid, "Meeting the accuracy challenge in airborne lidar bathymetry," in *Proceedings of EARSeL-SIG-Workshop LIDAR*, June 2000, pp. 1–27.
34. Phoenix aerial UAV lidar. [Online]. Available: <http://www.phoenix-aerial.com/>.
35. D. Wood and M. Bishop, "A novel approach to 3D laser scanning," in *Proceedings of Australasian Conference on Robotics and Automation*, New Zealand, 2012.
36. H. Baltzakis, A. Argyros, and P. Trahanias, "Fusion of laser and visual data for robot motion planning and collision avoidance," *Machine Vision and Applications*, vol. 15, no. 2, pp. 92–100, 2003.
37. O. Amidi, T. Kanade, and R. Miller, "Vision-based autonomous helicopter research at Carnegie Mellon robotics institute 1991–1997," in *Proceedings of American Helicopter Society International Conference*, Gifu, Japan, 1998, pp. 1–12, paper no. T73.
38. N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 331–340, 2008.
39. E. N. Johnson, A. J. Calise, Y. Watanabe, J. Ha, and J. C. Neidhoefer, "Real-time vision-based relative aircraft navigation," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, pp. 707–738, 2007.
40. S. Hrabar, G. S. Sukhatme, P. Corke, K. Usher, and J. Roberts, "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3309–3316.
41. O. Shakernia, Y. Ma, T. J. Koo, and S. Sastry, "Landing an unmanned air vehicle: Vision based motion estimation and nonlinear control," *Asian Journal of Control*, vol. 1, pp. 128–204, 1999.
42. C. S. Sharp, O. Shakernia, and S. S. Sastry, "A vision system for landing an unmanned aerial vehicle," in *Proceedings of the 2001 IEEE International Conference on Robotics & Automation*, Seoul, Korea, 2001, pp. 1720–1727.
43. J. Kim and S. Sukkarieh, "Slam aided GPS/INS navigation in GPS denied and unknown environments," in *The 2004 International Symposium on GNSS/GPS*, Sydney, Australia, 2004.
44. M. Meingast, C. Geyer, and S. Sastry, "Vision based terrain recovery for landing unmanned aerial vehicles," in *Proceedings of IEEE Conference on Decision and Control*, Atlantis, Bahamas, 2004, pp. 1670–1675.
45. K. Litomisky, "Consumer RGB-D cameras and their applications," 2012, introductory document about the state-of-the-art RGB-D camera applications.
46. P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments," in *Proceedings of the 12th International Symposium on Experimental Robotics (ISER)*, 2010.
47. P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using kinect-style depth cameras for dense 3D modeling of indoor environments," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.
48. H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox, "Interactive 3D modeling of indoor environments with a consumer depth camera," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, 2011.
49. A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, "Visual odometry and mapping for autonomous flight using an RGB-D camera," in *Proceedings of the International Symposium on Robotics Research*, 2011.
50. E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the 9th European Conference on Computer Vision*, 2006.
51. N. Fioraio and K. Konolige, "Realtime visual and point cloud SLAM," in *Proceedings of the RGB-D Workshop, Robotics Science and Systems*, 2011.
52. F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *ICRA, IEEE*, 2012, pp. 1691–1696.

53. G. Cai, F. Lin, B. M. Chen, and T. H. Lee, "Systematic design methodology and construction of UAV helicopters," *Mechatronics*, vol. 18, no. 10, pp. 545–558, 2008.
54. S. G. Fowers, D. J. Lee, B. J. Tippetts, K. D. Lillywhite, A. W. Dennis, and J. K. Archibald, "Vision aided stabilization and the development of a quad- rotor micro UAV," in *Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, FL, 2007, pp. 143–148.
55. R. He, A. Bachrach, M. Achtelik, A. Geramifard, D. Gurdan, S. Prentice, J. Stumpf, and N. Roy, "On the design and use of a micro air vehicle to track and avoid adversaries," *The International Journal of Robotics Research*, vol. 29, pp. 529–546, 2010.
56. M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments," in *The First Symposium on Indoor Flight Issues 2009*, Mayagez, Spanish, 2009.
57. S. K. Phang, J. J. Ong, R. T. C. Yeo, B. M. Chen, and T. H. Lee, "Autonomous mini-UAV for indoor flight with embedded on-board vision processing as navigation system," in *Proceedings of the IEEE R8 International Conference on Computational Technologies in Electrical and Electronics Engineering*, Irkutsk Listvyanka, Russia, 2010.
58. Wikipedia. Real-time operating system. [Online]. Available: http://en.wikipedia.org/wiki/Real-time_operating_system.
59. Qnx neutrio rtos. [Online]. Available: <http://www.qnx.com/>.
60. Vxworks rtos. [Online]. Available: <http://www.windriver.com/products/vxworks/>.
61. Rt linux rtos. [Online]. Available: <http://www.rtlinuxfree.com/>.
62. M. Dong, B. M. Chen, G. Cai, and K. Peng, "Development of a real-time onboard and ground station software system for a UAV helicopter," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 4, no. 8, pp. 933–955, 2007.
63. Visual C++ developer center. [Online]. Available: <http://msdn.microsoft.com/en-us/visualc/default.aspx>.
64. F. Lin, Z. Y. Ang, F. Wang, B. M. Chen, T. H. Lee, B. Q. Yang, M. B. Dong et al., "Development of an unmanned coaxial rotorcraft for the DARPA UAVForge challenge," *Unmanned Systems*, vol. 1, pp. 211–245, 2013.
65. A. Koehl, H. Rafaralahy, M. Boutayeb, and B. Martinez, "Aerodynamic modelling and experimental identification of a coaxial-rotor UAV," *Journal of Intelligent & Robotic Systems*, vol. 68, pp. 53–68, 2012.
66. B. Mettler, M. Tischler, and T. Kanade, "System identification of a model-scale helicopter," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-00-03, January 2000.
67. C. Harris, *Shock and Vibration Handbook*, 4th ed. New York: McGraw-Hill, 1996.
68. D. J. Kruglinski, *Inside Visual C++*, 4th ed. Washington, DC: Microsoft Press, 1995.
69. P. Liu, X. Dong, B. M. Chen, and T. H. Lee, "Development of an enhanced ground control system for unmanned aerial vehicles," in *Proceedings of the IASTED International Conference on Engineering and Applied Science*, Colombo, Sri Lanka, December 2013, pp. 136–143.
70. Unmanned Aircraft Systems Group, National University of Singapore, "The video of the final round competition of UAVGP (English version)." Available: <http://www.youtube.com/watch?v=Ex0jgtX2ZrY>, 2013.
71. Unmanned Aircraft Systems Group, National University of Singapore, "The video of the final round competition of UAVGP (Chinese version)." Available: <http://v.youku.com/vshow/idXNjI5MTM2MDI0.html>, 2013.