Guidance and Nonlinear Control System for Autonomous Flight of Minirotorcraft Unmanned Aerial Vehicles

Farid Kendoul  
Robotics and Control Laboratory, Department of Electronics and Mechanical Engineering, Chiba University, Chiba City 263-8522, Japan  
e-mail: fkendoul@restaff.chiba-u.jp

Zhenyu Yu  
State Key Laboratory of Rail Traffic Control and Safety, Beijing JiaoTong University, Beijing 100044, China  
e-mail: yzycn@yahoo.com

Kenzo Nonami  
Robotics and Control Laboratory, Department of Electronics and Mechanical Engineering, Chiba University, Chiba City 263-8522, Japan  
e-mail: nonami@faculty.chiba-u.jp

Received 18 March 2009; accepted 18 September 2009

Small unmanned aerial vehicles (UAVs) are becoming popular among researchers and vital platforms for several autonomous mission systems. In this paper, we present the design and development of a miniature autonomous rotorcraft weighing less than 700 g and capable of waypoint navigation, trajectory tracking, visual navigation, precise hovering, and automatic takeoff and landing. In an effort to make advanced autonomous behaviors available to mini- and microrotorcraft, an embedded and inexpensive autopilot was developed. To compensate for the weaknesses of the low-cost equipment, we put our efforts into designing a reliable model-based nonlinear controller that uses an inner-loop outer-loop control scheme. The developed flight controller considers the system’s nonlinearities, guarantees the stability of the closed-loop system, and results in a practical controller that is easy to implement and to tune. In addition to controller design and stability analysis, the paper provides information about the overall control architecture and the UAV system integration, including guidance laws, navigation algorithms, control system implementation, and autopilot hardware. The guidance, navigation, and control (GN&C) algorithms were implemented on a miniature quadrotor UAV that has undergone an extensive program of flight tests, resulting in various flight behaviors under autonomous control from takeoff to landing. Experimental results that demonstrate the operation of the GN&C algorithms and the capabilities of our autonomous micro air vehicle are presented. © 2009 Wiley Periodicals, Inc.

1. INTRODUCTION

Rotorcraft unmanned aerial vehicles (RUAVs) are being widely used in a multitude of applications, mostly military but also civilian. In recent years, there has been a growing interest in developing autonomous miniature RUAVs and micro air vehicles (MAVs) capable of achieving more complex missions, and increasing demands are being placed on the hardware and software that comprise their guidance, navigation, and control (GN&C) systems. Miniature RUAVs offer major advantages when used for aerial surveillance, reconnaissance, and inspection in cluttered environments and small spaces in which larger RUAVs and fixed-wing aircraft are not well suited. Their usefulness results from their low-cost, small size, VTOL (vertical takeoff and landing) capabilities, and ability to fly in very low altitudes, hover, cruise, and achieve aggressive maneuvers.

The design of autopilots for miniRUAVs has many theoretical and technical challenges. Indeed, the limited payload of miniature vehicles imposes severe constraints on the selection of navigation sensors and onboard electronics. Furthermore, due to their complex dynamics, nonlinearities, and high degree of coupling among control inputs and state variables, the design of reliable and robust controllers is a challenge. During the past decade, most research activities on the control of miniature RUAVs have been devoted to performing hovering in constrained environments. However, little work has been done on the development of fully autonomous miniRUAVs that can achieve real-world applications. In fact, milestones in UAVs and most reported experimental results about advanced flight behaviors have been achieved using fixed-wing aircraft and small-scale helicopters that weigh several kilograms, such as the CMU robotic helicopter (La Civita, Papageorgiou,
Messner, & Kanade, 2006; Scherer, Singh, Chamberlain, & Elgersma, 2008), the Berkeley BEAR project (Kim & Shim, 2003), the USC AVATAR helicopter (Saripalli, Montgomery, & Sukhatme, 2003), the GTMax Unmanned Helicopter (Johnson & Kannan, 2005), and the MIT acrobatic helicopter (Gavrilets, Mettler, & Feron, 2004).

These robotic helicopters have relied on various linear and nonlinear control techniques to perform autonomous flight. Conventional approaches to flight control and most initial attempts to achieve autonomous helicopter flight have been based on linear controller design. These approaches involve simplified model derivation and model linearization about a set of preselected equilibrium conditions or trim points. Linear controllers are then designed, such as decentralized single-input single-output (SISO) proportional integral derivative (PID) controllers (Kim & Shim, 2003), multiple-input multiple-output (MIMO) linear quadratic regulator (LQR) controllers (Shin, Fujiwara, Nonami, & Hazawa, 2005), and robust $H_{\infty}$ controllers (La Civita et al., 2006). The main drawback of linear controllers relates to overlooking coupling terms and nonlinearities, resulting in performance degradation when the aircraft moves away from a design trim point or hovering condition.

To overcome some of the limitations and drawbacks of linear approaches, a variety of nonlinear flight controllers have been developed and applied to helicopter UAV control. Among these, feedback linearization (Koo & Sastry, 1998), model predictive control (Kim & Shim, 2003), dynamic inversion (Reiner, Balas, & Garrard, 1995), adaptive control (Johnson & Kannan, 2005), and backstepping methodology (Mahony & Hamel, 2004; Olfati-Saber, 2001) have received much of the attention and showed great promise. For aggressive flight control, human-inspired and learning-based strategies have been developed and successfully tested using acrobatic helicopters (Abbeel, Coates, Quigley, & Ng, 2007; Gavrilets et al., 2004).

Among noted and recent research contributions to the miniature RUAV control problem, we find many on mini-quadrotor platforms. Indeed, quadrotor helicopters are emerging as a popular platform for UAV research, and many research groups are now working on quadrotors as UAV test beds for autonomous control and sensing. In several projects, quadrotor vehicles achieved autonomous flight using linear controllers on linearized dynamic models. The most applied and successful linear controllers are PID controllers (Hoffmann, Huang, Waslander, & Tomlin, 2007) (STARMAC quadrotor), proportional-derivative (PD) controllers (Bouabdallah, Murrieri, & Siegwart, 2005) (OS4 quadrotor), and LQR controllers (How, Bethke, Frank, Dale, & Vian, 2008) (RAVEN test bed). Many researchers from the control community have also developed nonlinear control systems for mini-quadrotor vehicles. One such system, based on the Draganflyer quadrotor airframe, has demonstrated successful hovering using nested saturation-based nonlinear controllers (Castillo, Dzul, & Lozano, 2004; Kendoul, Lara, Fantoni, & Lozano, 2007). In the X4-flyer project (Guenard, Hamel, & Moreau, 2005), a backstepping technique was applied to design a nonlinear controller for a miniature quadrotor vehicle. Attitude control was also demonstrated on tethered quadrotor test beds using sliding mode control (Madani & Benallegue, 2006). Another tethered test bed (Tayebi & McGilvray, 2006) used a quaternion-based nonlinear control scheme to perform attitude stabilization. This list is not exhaustive, and there are many other interesting works on the control, navigation, and visual servoing of mini-quadrotor UAVs (e.g., Altug, Ostrowski, & Taylor, 2005; He, Prentice, & Roy, 2008).

Although significant progress has been made on the control and development of miniature RUAVs, the achieved flight performances are modest compared to those of larger UAVs. From a control systems perspective, most of the control designs previously cited present a trade-off between flight performance and control law complexity. Indeed, linear controllers that are designed for near-hovering flight fail to provide good flight performance at nonnominal operating conditions in which attitude angles are big, such as during high-speed and aggressive flights. On the other hand, nonlinear controllers are often difficult to implement on small microprocessors and to tune online. From an experimental and achieved performance perspective, the available experimental results are generally limited to hovering flight in structured and indoor environments.

The aim of our research is to demonstrate the feasibility of using autonomous mini- and microRUAVs for search and rescue missions by developing and validating a low-cost aerial platform capable of achieving various mission scenarios autonomously (Figure 1). This paper presents the development of an embedded lightweight autopilot that strikes a balance between simplicity and performance. A detailed description of the autopilot hardware and GN&C algorithms and their evaluation through real-time experiments is also provided.

Although there have been good examples of autonomous control of rotorcraft UAVs and of mini-quadrotor helicopters in particular, our work and obtained results represent significant advances for autonomous MAVs:

1. From a control systems perspective, we have designed a hierarchical model-based nonlinear controller that uses an inner–outer-loop control scheme and has the following benefits:
   - It considers system nonlinearities and couplings while guaranteeing the asymptotic stability of the closed-loop system.
   - It is a multipurpose controller that can handle different flight modes such as hovering, flying forward, flying sideward, takeoff and landing, and trajectory tracking.
2. From a hardware perspective, we have carefully selected lightweight avionics components that fit the limited payload of miniature vehicles, and we have designed an embedded real-time architecture that includes navigation sensors (inertial measurement unit (IMU), global positioning system (GPS)), pressure sensor (PS), mission sensors (camera), a flight control computer (FCC), and a wi-fi communication module. In contrast to other experiments reported in the literature, we do not rely on any accurate IMU or GPS whose costs and weights are significantly higher. We believe that cost reduction will yield a substantial speedup of the use of MAVs for civilian applications.

3. From a UAV system integration perspective, we have developed and implemented guidance, navigation, and vision algorithms that offer advanced autonomous behaviors to MAVs and miniature rotorcraft that weigh less than 0.7 kg.

4. From an experimental perspective, we have performed several flight tests in outdoor and natural environments using a mini-quadrotor helicopter (size 53 cm and total weight 700 g). The quadrotor MAV, equipped with the developed autopilot, has undergone an extensive program of flight tests, resulting in various autonomous flight behaviors (automatic takeoff and landing, accurate hovering, long-distance flight, waypoint navigation, trajectory tracking, and vision-based target tracking).

Roughly speaking, the main achievement of this work is to provide miniature rotorcraft and MAVs with advanced flight behaviors and autonomous capabilities despite their limited payload and available computational resources.

In the next section, the problem statement is discussed. The air vehicle and the autopilot hardware are described in Section 3. Section 4 gives an overview about the GN&C systems. The theory of the designed nonlinear controller, including the analysis of the closed-loop system stability, is presented in Section 5. The flight results, presented in Section 6, are a clear illustration that waypoint navigation, trajectory tracking, and visual navigation capabilities can be extended to smaller platforms such as MAVs.

2. PROBLEM STATEMENT THROUGH A TYPICAL MISSION SCENARIO

An autonomous aerial vehicle, equipped with an onboard camera, can be considered as an eye in the sky and thus can be used for many applications. In this section, we motivate the use of autonomous UAVs/MAVs for search and rescue applications and we present the main issues to be addressed for achieving these kinds of missions autonomously.

Let us consider the mission scenario in which MAVs are used to assist rescue teams during a natural disaster such as an earthquake. Many difficulties make search and rescue missions challenging. Among the problems that should be addressed by rescue teams, one can cite the following:

1. Determination of priority areas: When an earthquake occurs, many areas and cities are generally damaged. Simultaneous intervention in all inhabited areas may not be possible. Therefore, the most heavily damaged areas where help and assistance are most urgently needed should be determined quickly.

2. Damage evaluation: Once a high-priority area is identified, rescue teams need to have an idea about the nature and degree of damage. A priori knowledge about damage is a key determinant for mission success. This information will help to define the appropriate size of rescue teams and needed equipment and material, etc.

3. Identification of safe access roads: Another common and important problem during rescue missions is the identification of access roads to the area of interest. Bridges and roads may be destroyed by earthquakes.

4. Search for survivors and injured persons: Injured persons need quick help and immediate medical assistance. Finding survivors in a short time is a challenging task because of the large search area and building debris.

Many lives can be saved if the above issues can be effectively addressed. We believe that the use of advanced technologies such as autonomous rotorcraft MAVs will increase mission success and facilitate search and rescue operations. Miniature RUAVs present many advantages for
search and rescue applications because of their maneuverability and their ability to perform stationary flight. Despite their low cost, rotorcraft MAVs can be deployed quickly and operated easily to achieve the previously listed tasks autonomously. Furthermore, their small size allows them to fly between buildings at low altitudes for accurate data collection without any risk and danger for people.

To develop a fully autonomous rotorcraft MAV that can achieve the previously listed tasks autonomously, some flight capabilities and behaviors are required:

- A robust platform with onboard navigation and mission sensors that can send images in real time to the operation station.
- Autonomous GPS-based waypoint navigation capability in order to fly to the area of interest and provide information (images) about access roads and the nature of damage.
- When the MAV arrives at the target zone, it may execute some motion patterns to explore the area and to avoid obstacles. Therefore, accurate trajectory tracking capability is necessary, especially during search operations.
- The GPS signal may be weak or not available at some places. An alternative option for increasing the navigation accuracy is to use vision systems. Cameras are small and lightweight and provide rich information for mission requirements but also for MAV motion estimation. A visual navigation system will significantly increase the autonomous capabilities of the rotorcraft.

This paper addresses the above issues and describes the design of an autonomous rotorcraft MAV with onboard intelligent capabilities. Furthermore, the paper presents experimental results that show that our quadrotor MAV can achieve autonomously all the navigation tasks listed. With these flight capabilities, the developed platform will be ready to embark on many of the applications envisaged for it, including search and rescue missions.

3. AIR VEHICLE AND AUTOPILOT HARDWARE

To demonstrate autonomous flight of a rotorcraft MAV (RMAV), the vehicle platform should be suitably integrated with subsystems as well as hardware and software. The choice of the air vehicle and avionics is a crucial step toward the development of an autonomous platform. That choice is mainly dependent on the intended application as well as performance, cost, and weight. In this section, we will describe the air vehicle and introduce the major subsystems implemented in our RMAV system, shown in Figure 2.

3.1. Air Vehicle Description

Safety and performance requirements drove the selection of the quadrotor vehicle as a safe, easy-to-use platform with very limited maintenance requirements. A versatile hobby quadrotor, the X-3D-BL produced by Ascending Technologies GmbH in Germany, is adopted as a vehicle platform for our research. The vehicle design consists of a carbon-fiber airframe with two pairs of counterrotating, fixed-pitch blades as shown in Figure 2. The vehicle is 53 cm rotor tip to rotor tip and weighs 400 g, including the battery. Its payload capacity is about 300 g. The propulsion system consists of four brushless motors, powered by a 2,100-mAh three-cell lithium polymer battery. The flight time in hovering is about 12 min with full payload.

The original hobby quadrotor includes a controller board called X-3D that runs three independent PD loops at 1 kHz, one for each rotational axis (roll, pitch, and yaw). Angular velocities are obtained from three gyroscopes (Murata ENC-03R), and angles are computed by integrating gyroscope measurements. More details about the X-3D-BL platform and its internal controller can be found in Gurdan et al. (2007). In our implementation, the internal attitude controller that comes with the original platform has been disabled except for gyro feedback stabilization and yaw angle control, which cannot be disabled.

3.2. Navigation Sensors and Embedded Architecture

The hardware components that constitute the basic flight avionics of our platform include a small microcontroller from Gumstix and the MNAV100CA sensor from Crossbow.

3.2.1. FCC

The embedded architecture is built around the Gumstix (connex 400) microcontroller. A Gumstix motherboard is a very small Linux OpenEmbedded computer that features a Marvell PXA255 processor. This computing unit presents many advantages for MAV applications because it weighs only 8 g despite its good performance. Indeed, it has a CPU clock of 400 MHz, a flash memory of 16 MB, and SDRAM of...
3.2.2. Navigation and Control Sensors

After studying and comparing different sensors, we have selected the MNAV100CA sensor from Crossbow due to its low cost and light weight (about 35 g without GPS antenna). The MNAV100CA is a calibrated digital sensor system with servo drivers, designed for miniature ground and air robotic vehicle navigation and control. All sensors required for complete airframe stabilization and navigation are integrated into one compact module. It includes an IMU that outputs the raw data of three gyroscopes, three accelerometers, and three magnetometers at 50 Hz. It also includes a static PS (Motorola freescale MPXH6115A6U with a sensitivity of 46 mV/kPa), which is used to determine the height above the ground. Position and velocity measurements are provided at 4 Hz using an integrated GPS receiver module (u-blox TIM-LP). The position accuracy is about ±2 m for the horizontal position and about ±5 m for altitude. The PPM interface allows for software interpretation of radio control (RC) receiver commands (PPM) and switching between autonomous and manual flight modes.

3.2.3. Autopilot Architecture

When designing the embedded system, much attention has been paid to simplicity, modularity, and safety. A diagram showing the interaction between the different components onboard the vehicle is shown in Figure 3. All sensor data, including IMU measurements (three accelerations, three angular rates, and three magnetometers readings), GPS data [latitude, longitude, altitude, and three translational velocities in the north-east-down (NED) frame], PS data (height), and the RC receiver decoded signals (throttle, pitching/rolling/yawing torques, switch, communication status), are sent from MNAV to the Gumstix FCC through the serial port RS-232. For wireless communication with the ground control station (GCS), we have mounted the wifistix card from Gumstix on the 92-pin connector of the Gumstix motherboard, thereby providing a communication module with high bandwidth (about 50 Mbits/s). The communication range is about 500 m, but it can be increased up to 800 m by reducing the communication bandwidth to 2 Mbits/s.

The FCC has three main tasks: (1) read data from the MNAV sensor and estimate the three-dimensional (3D) pose (position and orientation) of the vehicle, (2) implement guidance and control algorithms, and (3) manage the wi-fi communication (uplink and downlink or telemetry) with the GCS. The X-3D-BL hobby quadrotor is designed to be controlled manually through an RC transmitter–receiver. It has no high-level commands and data interface. Thus, in order to interface our autopilot with the X-3D-BL board, we have added a small AVR microcontroller (Atmega32) that receives the control inputs from the FCC and generates a PPM signal. In fact, the four control inputs (total thrust, pitching torque, rolling torque, and yawing torque), computed by the control system, are sent from the main FCC to the AVR microcontroller through the serial port RS-232. They are then encoded into a PPM signal, which is sent to the X-base board, thereby simulating an RC receiver signal. From Figure 3, we can see that the sixth channel of the RC receiver is also connected to the auxiliary AVR microcontroller. This channel is used to terminate the flight at any moment if the embedded system is not behaving correctly.

Our platform also includes an analog wireless camera that is used as the main mission sensor. The vision module is composed of a camera, KX171 from Range Video, a 1.3-GHz video transmitter and receiver, and a tilt system that is driven by the FCC through the MNAV servo driver interface.

![Figure 3](https://example.com/figure3.png)

**Figure 3.** The hardware components of the autopilot and their interaction onboard the vehicle. The flight avionics of our platform include a Gumstix microcontroller, an IMU, a GPS, static and dynamic PSs, and a wireless camera.
Table I. Platform components.

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
<th>Weight (g)</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airframe</td>
<td>Carbon fiber frame + CSM-core</td>
<td>95</td>
<td>≈1,400</td>
</tr>
<tr>
<td>Propulsion</td>
<td>4 brushless motors + 4 X-BLDC controller + 4 propellers</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>X-3D-BL electronics</td>
<td>X-3D + X-base</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Battery</td>
<td>Thunder Power Li-polymer, 2,100 mAh</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>RC receiver</td>
<td>Futaba, 40 MHz (6 channels)</td>
<td>10</td>
<td>50</td>
</tr>
<tr>
<td>Sensors</td>
<td>Crossbow MNAV100CA</td>
<td>35</td>
<td>1,500</td>
</tr>
<tr>
<td>Flight computer</td>
<td>Gumstix connex 400 MHz + wi-fi module</td>
<td>28</td>
<td>230</td>
</tr>
<tr>
<td>Antennas</td>
<td>GPS antenna + wi-fi antenna</td>
<td>38</td>
<td>0</td>
</tr>
<tr>
<td>PPM generator</td>
<td>Atmega32 microcontroller</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Video system</td>
<td>Kx-171 camera + 1.3-GHz video transmitter + tilt system</td>
<td>77</td>
<td>410</td>
</tr>
<tr>
<td>Other</td>
<td>Pad + autopilot board/support + ···</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>658</td>
<td>3,620</td>
</tr>
</tbody>
</table>

Table I shows the repartition of mass and cost between the different subsystems of the vehicle.

4. GN&C SYSTEMS

GN&C algorithms are the core of the flight software of the MAV to successfully complete the assigned mission through autonomous flight. They offer the MAV the ability to follow waypoints and to execute other preprogrammed maneuvers such as automatic takeoff and landing, hovering, and trajectory tracking. As shown in Figure 4, the overall system architecture considered in this paper consists of six layers: (1) GCS for mission definition and high-level decision making, (2) guidance for path planning and trajectory generation, (3) navigation for vehicle state vector estimation, (4) nonlinear controller for stabilization and trajectory tracking, (5) communication with the GCS and interface between the autopilot and the vehicle, and (6) the MAV platform. All GN&C algorithms have been implemented in visual C using multithread programming. In our design, the autopilot software is implemented as a process within the Linux operating system. It is composed of smaller units of computation, know as tasks (or threads). By using POSIX-style semantics, theses tasks are called and scheduled separately (Jang & Liccardo, 2006) while sharing CPU and memory resources.

4.1. GCS

The GCS shown later in Figure 17 provides several critical functionalities. The main role of the GCS is to display the flight information downloaded from the MAV in real time and to upload the various operating commands such as operation mode, guidance commands, and controller gains. Furthermore, images from the onboard camera

1Here, waypoint navigation is defined as the process of automatically following a predetermined path defined by a set of geodetic (GPS) coordinates.

Figure 4. Overall software architecture of the autopilot. GN&C algorithms are implemented onboard the Gumstix microcontroller using multithread programming.
4.2. Guidance Strategy

The proposed guidance system takes inputs from the navigation system and uses targeting information to send signals to the flight control system that will allow the vehicle to reach its destination within the operating constraints of the vehicle. In the current system, the flight plan or mission is defined by a human operator using the GCS interface. This plan is expressed in terms of desired 3D waypoints and other maneuvers such as hovering, landing, trajectory tracking, and target tracking. During mission execution, the flight plan can be modified or updated in real time. These commands are transmitted from the GCS to the processor of the onboard avionics unit via a 2.4-GHz wi-fi link.

The embedded guidance system has the role of selecting the appropriate maneuver and generating the corresponding reference trajectories. It is composed of three main parts: (1) path planning, (2) mode transition management, and (3) trajectory generation. Although this guidance system is simple, it provides sufficient capabilities for UAVs operating in obstacle-free environments for many applications, such as reconnaissance and exploration. Advanced guidance and navigation capabilities are necessary to achieve more complex tasks and missions.

4.2.1. Path Planning

The transmitted waypoints from the GCS are expressed in the geodetic LLA (latitude–longitude–altitude) coordinates. They are then converted into the NED local frame, where they are expressed in meters according to the starting point. A path planning routine constructs a path, which is defined by a sequence of $N$ desired waypoints and desired speeds of travel along path segment $\omega_i$ connecting waypoint $\text{wpt}_{i}$ to $\text{wpt}_{i+1}$. The path planner is also required to have some strategy to detect when the waypoint $\text{wpt}_i$ is reached by the rotorcraft in order to switch to the next one ($\text{wpt}_{i+1}$). The proposed strategy attempts to achieve the desired position initially with 10-m accuracy in both lateral and longitudinal directions. It then takes an additional 20 s to achieve the position with 2-m accuracy and finally takes an additional 5 s to achieve higher accuracy. If the waypoint is not achieved to within 2 m in 20 s, the path planner assumes that external factors (i.e., wind) are interfering and ends the attempt. Flight paths that passed within these thresholds are considered to have reached the waypoint, so the guidance law then directs the UAV to reach the next waypoint. In this way, we are guaranteed some baseline level of performance (10 m), and the vehicle will attempt to achieve a higher level of accuracy without excessive time delays.

4.2.2. Mode Transition Management

The mode transition manager has the role of managing three autonomous flight modes (GPS-based flight, vision-based flight, and target tracking mode) and various flight behaviors (landing, hovering, trajectory tracking, etc.). These different modes and maneuvers are either selected by the human operator through the GCS interface or by the Fail-Safe system, described in Subsection 4.5. To minimize errors and reduce risk, each flight mode and maneuver has preconditions that must be satisfied; otherwise the task will be rejected. For example, when the GPS signal is weak or lost, the system switches automatically to vision-based flight and the landing procedure is activated after some waiting time to achieve a safe vision-based landing.

4.2.3. Trajectory Generation

The goal of this routine is to generate a physically feasible trajectory according to the flight behavior selected by the mode transition manager. In the case of waypoint navigation, for example, the trajectory generation routine takes straight-line path segments and produces feasible time-parameterized desired trajectories that are the desired 3D position $[x_d(t), y_d(t), z_d(t)]$, the desired heading $\psi_d(t)$, and their time derivatives $[\dot{x}_d(t), \dot{y}_d(t), \dot{z}_d(t), \dot{\psi}_d(t)]$. The kinematic model used for trajectory generation uses specifiable limits on the maximum speed and acceleration the rotorcraft may have during a maneuver. In fact, reference trajectories are generated in two steps: first, the desired velocity trajectory is computed from the specified accelerations and the desired maximum speed during forward flight. Second, the position reference trajectory is obtained by integrating the velocity trajectory with respect to time. When the vehicle approaches a desired waypoint, it starts decelerating in order to reach the waypoint by zero velocity. This simple
but effective strategy allows accurate and robust transition from one waypoint to the next without overshoot on sharp corners. Furthermore, the obtained position and velocity trajectories are differentiable with respect to time, which is needed for some control designs.

To achieve nonaggressive takeoff and soft landing, smooth reference trajectories $z_d(t)$ along the vertical axis are generated in real time based on the desired ascent/descent velocity and the actual height of the vehicle. For search operations, a spiral trajectory is generated in real time, thereby allowing the rotorcraft to explore some target area. Furthermore, this spiral trajectory will allow evaluation of the performance of the developed flight controller for tracking nonstraight and arbitrary trajectories. Our implementation also allows the easy incorporation of many other trajectories.

Although the resulting trajectories are not necessarily optimal, the lower computation burden of this trajectory generator ensures its implementation on small onboard microprocessors.

### 4.3. Navigation System

The main objective of a navigation system is to provide information about the vehicle’s self-motion (orientation, position, etc.) and its surrounding environment (obstacles, target, etc.). This information is then exploited by the flight controller for motion control and by the guidance system for mission achievement. The navigation system that we have developed and implemented on our platform can be split into two main parts: (1) a conventional navigation system for full state estimation, which includes attitude heading reference system (AHRS), GPS/INS, and PS/inertial navigation system (PS/INS); and (2) an advanced visual navigation system, which includes an optic flow–based localization module and a visual odometer for flight control and target tracking.

a. AHRS: An extended Kalman filter (EKF) is used to fuse IMU raw data (3 gyro., 3 acceler., 3 magneto.) in order to provide estimates about the vehicle’s attitude, heading, and angular rates at an updating rate of 50 Hz.

b. GPS/INS: GPS measurements (3D position in LLA coordinates and 3D velocity vector in the NED frame) are fused with the INS data using an EKF. The filtered and propagated position and velocity estimates are provided at an updating rate of 10 Hz.

c. PS/INS: The height estimated by the GPS/INS is not accurate enough (5–10-m error) to allow good altitude control. The altitude controller is thus based on the height estimated by the static PS after fusing it with the vertical acceleration using a kinematic Kalman filter. We have then improved the height estimation accuracy to ±1.5 m in normal flight conditions. However, the low-cost barometric PS used is sensitive to weather conditions such as wind and temperature.

d. Optic flow–based localization system: The standard navigation sensors and capabilities onboard the rotorcraft have been enhanced to include vision-based navigation. Details about this system as well as experimental results can be found in our previous paper (Kendoul, Fantoni, & Nonami, 2009). Functionally, the proposed vision system is based on optic flow computation (offboard processing) and solving the structure–from–motion problem (onboard processing) for vehicle ego-motion estimation and obstacle detection. Because it is necessary for the rotorcraft to fly between buildings and indoors during search missions, a strong GPS signal will probably not be available. To combat drift in the navigation solution, position and velocity updates are based on the vision system.

e. Visual relative navigation system: Because many real-world applications may require relative navigation capability, we have augmented our navigation system by a second vision algorithm that provides relative position and velocity to some selected target. The proposed vision system determines the vehicle’s relative position, velocity, and height by extracting and tracking visual features in video sequence of the onboard camera. The target can be chosen either by the operator by selecting the desired target on the image window displayed at the GCS or by another high-level recognition algorithm. In the present configuration, only the first solution is available. This vision system is of great interest and importance for search and rescue missions because it allows the MAV to fly in close proximity to objects of interest (windows, etc.) and to track targets (survivors, etc.) in order to collect more accurate information. A detailed description of this vision system and experimental validation can be found in Kendoul, Nonami, Fantoni, and Lozano (2009). In Section 6, we also present experimental results from a real-time flight test in which the rotorcraft tracks some moving ground target using visual estimates.

It is not currently possible to connect a camera directly to the FCC; therefore, it is necessary to do the image processing offboard. The present configuration transmits the video signal to the GCS for all image processing (optic flow computation and feature tracking), and then the results are relayed back to the onboard FCC as inputs to the visual navigation system running on the FCC at 10 Hz.

### 4.4. Nonlinear Hierarchical Controller

To enable more complex missions for autonomous RUAVs and for quadrotor MAVs in particular, we have developed a multipurpose nonlinear controller that allows accurate hovering and precise trajectory tracking but also waypoint navigation and moving target tracking. It takes as inputs
reference trajectories (guidance system outputs) and state estimates (navigation system outputs) and produces forces and torques that are required to control the vehicle’s motion. The theory of the designed nonlinear controller and its stability analysis are provided in Section 5.

4.5. Safety Procedures and Flight Termination System

When developing autonomous UAVs and MAVs for civilian applications, some level of safety must be guaranteed. Therefore, we have implemented several safety procedures on our rotorcraft to keep single failures from having catastrophic effects on both people and the vehicle itself.

4.5.1. Preprogrammed Fail-Safes

The developed autopilot has a number of preprogrammed fail-safes instructing the vehicle to perform a certain task when some unexpected problem occurs. Some of those are discussed here:

- Wi-fi communication: If the vehicle loses signal with the GCS for more than 5 s, the hovering mode is automatically enabled. If the communication is not established in 20 s, then the vehicle will switch to the “home mode,” and it will navigate to a preassigned location (home).
- Fly zone: When the vehicle travels out of the fly zone, the hovering mode is activated, waiting for new commands from the GCS to correct the trajectory or to recover the vehicle.
- GPS unit failure: In the case of GPS outages or failure, the rotorcraft will achieve hovering using visual estimates. Depending on the situation, the operator may continue the mission using the vision system or activate the safe landing procedure.
- Undesired behavior: If the aircraft experiences some undesired behavior, then the safety pilot can switch to manual flight or terminate the flight.

4.5.2. Flight Termination System

The safety procedures discussed above are implemented in the main FCC. A malfunction of the flight computer may occur due to some software bug or hardware problem. Thus, all the programmed safety procedures cannot be executed. To provide more safety, a simple independent flight termination system (FTS) is used to reduce and decrement the thrust from its nominal value to zero, thereby resulting in a soft emergency landing.

As discussed in Subsection 3.2., a secondary microcontroller is used as the primary interface to the physical vehicle. Moreover, it forms a FTS because it implements a flight termination program that can be activated in three different ways:

1. At any point in the mission, the operator can terminate the flight by pushing the “emergency” button on the GCS interface.
2. If the main FCC stops working for any reason, then the flight termination program is automatically activated.
3. The sixth channel of the RC receiver is directly connected to the FTS. Therefore, the safety pilot can use this channel to terminate the flight at any time, provided that the vehicle is in the radio communication range.

These safety precautions are very important especially during the autopilot development and testing stages, when the system is vulnerable due to programming errors, bad controller tuning, etc. We have successfully tested all the listed safety procedures without damaging the vehicle.


Generally, it is difficult to design a reliable 3D flight controller for rotorcraft UAVs/MAVs, because helicopter dynamics is inherently unstable, highly nonlinear, and coupled in the inner axis. In this research, we designed a controller based on the nonlinear model of the rotorcraft. Our objective is to design a 3D flight controller that performs well in practice as well as in theory. This is accomplished by deriving a mathematical model for miniRUAV dynamics and exploiting its structural properties to transform it into two cascaded subsystems coupled by a nonlinear interconnection term. A partial passivation design has been used to synthesize control laws for each subsystem, thereby resulting in a hierarchical and nonlinear inner–outer-loop controller. The inner loop with fast dynamics performs attitude tracking and generates the required torques. The outer loop with slow dynamics is used to generate the thrust and the reference angles required to follow a commanded translational trajectory. The asymptotic stability of the entire connected system is proven by exploiting the theories of systems in cascade. The resulting nonlinear controller is thus simple to implement and easy to tune, and it results in good flight performance.

5.1. Rotorcraft Dynamics Model

Controller design for rotorcraft UAVs and MAVs is a complex matter and typically requires the availability of the mathematical model of its dynamics. A rotorcraft can be considered a rigid body that incorporates a mechanism for generating the required forces and torques. The derivation of the nonlinear dynamics is first performed in the body-fixed coordinates \( \mathcal{B} \) and then transformed into the NED inertial frame \( I \). Let \( \{e_1, e_2, e_3\} \) denote unit vectors along the respective inertial axes and \( \{x, y, z\} \) denote unit vectors along the respective body axes, as defined in Figure 5.

The equations of motion for a rigid body of mass \( m \in \mathbb{R} \) and inertia \( J \in \mathbb{R}^{3 \times 3} \) subject to external force \( F_{ext} \in \mathbb{R}^3 \) and torque \( \tau \in \mathbb{R}^3 \) are given by the following Newton–Euler
where $\xi$ expressed as follows: given by a rotation matrix $R$ convention "ZYX," the airframe orientation in space is translational acceleration and other body force components. The quadrotor body diagram with the associated forces and frames.

where $V = (u, v, w)$ and $\Omega = (p, q, r)$ are, respectively, the linear and angular velocities in the body-fixed reference frame. The translational force $F_{\text{ext}}$ combines gravity, main thrust, and other body force components.

Using Euler angle parameterization and the aeronautical convention “ZYX,” the airframe orientation in space is given by a rotation matrix $R$ from $B$ to $I$, where $R \in SO(3)$ is expressed as follows:

$$R = R_\psi R_\theta R_\phi = 
\begin{pmatrix}
  c\theta c\psi & c\theta s\psi & s\theta \\
  c\phi s\theta c\psi - c\phi s\psi & c\phi s\theta s\psi + c\phi c\psi & s\phi s\psi - s\phi c\psi \\
  -s\theta c\phi & s\phi c\theta & c\theta
\end{pmatrix},$$

where $\eta = (\phi, \theta, \psi)$ denotes the vector of three Euler angles and $s$ and $c$ are abbreviations for $\sin(\cdot)$ and $\cos(\cdot)$.

By considering this transformation between the body-fixed reference frame and the inertial reference frame, it is possible to separate the gravitational force from other forces and write the translational dynamics in $I$ as follows:

$$\begin{align*}
\dot{\xi} &= \upsilon, \\
\dot{m}\upsilon &= -RF + mg_e_3,
\end{align*}$$

where $\xi = (x, y, z)$ and $\upsilon = (\dot{x}, \dot{y}, \dot{z})$ are the rotorcraft position and velocity in $I$. In the above notation, $g$ is the gravitational acceleration and $F$ is the resulting force vector in $B$ (excluding the gravity force) acting on the airframe.

For control design purposes, we will transform the attitude dynamics into an appropriate form using Euler angle parameterization. Let us first recall the kinematic relation between $\Omega$ and $\dot{\eta}$:

$$\dot{\eta} = \Phi(\eta)\Omega,$$

where the Euler matrix $\Phi(\eta)$ is given by

$$\Phi(\eta) = 
\begin{pmatrix}
  \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) & 0 \\
  \cos(\phi) & -\sin(\phi) & 0 \\
  \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) & 0
\end{pmatrix}.$$  

It is important to note that the matrix $\Phi$ has a singularity at $\theta = \pm \pi/2$, and its inverse matrix $\Psi(\eta) = \Phi^{-1}(\eta)$ is given by

$$\Psi(\eta) = 
\begin{pmatrix}
  1 & 0 & -\sin \theta \\
  0 & \cos \varphi & \cos \theta \sin \varphi \\
  0 & -\sin \varphi & \cos \varphi \cos \theta
\end{pmatrix}.$$  

By differentiating Eq. (4) with respect to time and recalling the second equation (1), we write

$$\dot{\eta} = \Phi \Omega + \Phi \dot{\Omega} = \Phi \Psi\dot{\eta} - \Phi J^{-1} \mathbf{s}(\Omega) J \Omega + \Phi J^{-1} \tau.$$

The sk operation is defined here from $\mathbb{R}^3$ to $\mathbb{R}^{3 \times 3}$ such that $\mathbf{s}(x)$ is a skew-symmetric matrix associated to the vector product $\mathbf{s}(x) y := x \times y$ for any vector $y \in \mathbb{R}^3$.

By multiplying both sides of the last equation by $\Psi(\eta)^T J \Psi(\eta)$, the attitude dynamics of the quadrotor can be expressed in the form (Olafí-Saber, 2001)

$$M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} = \Psi(\eta)^T \tau$$

with a well-defined positive definite inertia matrix $M(\eta) = \Psi(\eta)^T J \Psi(\eta)$. The Coriolis and centrifugal matrix $C(\eta, \dot{\eta})$ is given by

$$C(\eta, \dot{\eta}) = -\Psi(\eta)^T J \Psi(\eta) + \Psi(\eta)^T \mathbf{s}[\Psi(\eta) \dot{\eta}] J \Psi(\eta).$$

By considering the symmetry of the quadrotor vehicle, the inertia matrix $J$ can be approximated by $J = \text{diag}(J_1, J_2, J_3)$. Therefore, the matrix $M(\eta)$ is given by

$$M(\eta) = 
\begin{pmatrix}
  J_1 & 0 & -J_1 s\theta \\
  0 & J_2 s^2 \phi + J_3 c^2 \phi & J_2 \phi \cos \varphi - J_3 \phi \sin \varphi \\
  -J_1 c\theta & (J_2 - J_3)c\theta c\phi \sin \varphi + J_3 \phi c\theta \sin \varphi & J_3 s^2 \phi + J_2 c^2 \phi
\end{pmatrix},$$

where $\phi$, $\theta$, and $\psi$ are the rotor speeds. Therefore, the force and torque vectors in Eqs. (10) can be expressed as $F = (0, 0, u)^T$ and $\tau = (\tau_\phi, \tau_\theta, \tau_\psi)^T$. Aerodynamic effects, rotor dynamics, and

2We have exploited the fact that $\Psi = \Phi \Phi \Psi$.
gyroscopic effects are ignored. This is justified by the fact that mini-quadrotor platforms have a small airframe, fly at relatively low speeds, and have small propellers. For a more accurate model of quadrotor helicopters, one can refer to Bouabdallah et al. (2005) and Hoffmann et al. (2007) and the references therein.

A common simplified model used to represent the relation between the control inputs \((u, \tau_\phi, \tau_\theta)\) and the rotor speeds \((w_1, w_2, w_3, w_4)\), is given by

\[
\begin{bmatrix}
u \\
\tau_\phi \\
\tau_\theta \\
\end{bmatrix} =
\begin{bmatrix}
\rho & \rho & \rho & 0 \\
-1 & 0 & 0 & 1 \\
-1 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
w_1^2 \\
w_2^2 \\
w_3^2 \\
w_4^2 \\
\end{bmatrix},
\tag{11}
\]

where \((\rho, \kappa)\) are positive constants characterizing the propeller aerodynamics and \(l\) denotes the distance from rotors to the center of mass.

The dynamical model considered for control design and that can represent a wide range of RUAV configurations including the quadrotor helicopter is given by

\[
m \ddot{\xi} = -u R(\eta) e_3 + m g e_3,
\]

\[
M(\eta) \ddot{\eta} + C(\eta, \dot{\eta}) \dot{\eta} = \Psi(\eta)^T \tau.
\tag{12}
\]

The rotorcraft dynamics described by system (12) can be considered as two connected subsystems in which the translation and rotation dynamics are coupled through the rotation matrix \(R(\eta)\). It is also important to note that the rotation dynamics do not depend on translation components; on the other hand, the translational dynamics do depend on angles via the rotation matrix \(R(\eta)\). In fact, the rotorcraft translation is a direct result of its attitude change, and it is controlled by directing the force vector in the appropriate direction. This structural property of rotorcraft UAVs is exploited here to design a nonlinear hierarchical controller.

Remark 1. The flight controller, designed in the next section, is based on nonlinear model (12). Therefore, it can be applied to control rotorcraft UAVs whose dynamics can be represented by system (12). However, for larger RUAVs, we believe that control design should consider aerodynamic effects and rotor dynamics for more effective and accurate control (Huang, Hoffmann, Waslander, & Tomlin, 2009; Pounds, Mahony, & Corke, 2006).

5.2 Flight Controller Design

Controller design for nonlinear systems subject to strong coupling offers both practical significance and theoretical challenges. In this paper, the control design for small RUAVs and the quadrotor MAV in particular is addressed in three steps:

1. Decouple the translational and attitude dynamics by transforming nonlinear model (12) into two linear subsystems coupled by a nonlinear interconnection term.
2. Synthesize two independent controllers for the translation and rotation subsystems.
3. Prove the asymptotic stability of the entire connected closed-loop system by exploiting cascaded systems theories.

Because the attitude dynamics in Eqs. (12) is a fully actuated mechanical system for \(\theta \neq k \pi/2\), then it is exact feedback linearizable. In fact, by applying the change of variables

\[
\tau = J \Psi(\eta) \ddot{\eta} + \Phi^T C(\eta, \dot{\eta}) \dot{\eta},
\tag{13}
\]

we obtain a 3D double integrator where \(\dot{\tau}\) is a new control input. Hence, the dynamics of the rotorcraft in Eqs. (12) transforms into

\[
\dot{x} = -\frac{1}{m} u (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi), \quad \dot{\phi} = \dot{x},
\]

\[
\dot{y} = -\frac{1}{m} u (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi), \quad \dot{\theta} = \dot{y},
\tag{14}
\]

\[
\dot{z} = -\frac{1}{m} u \cos \phi \cos \psi + g, \quad \dot{\psi} = \dot{z}.
\]

In contrast to previous works on quadrotors, our objective here is to design a multipurpose controller that can perform autonomous stabilization of the vehicle but also accurate trajectory tracking for both position and attitude. These capabilities are required to achieve many realistic applications.

Let \(\xi_d(t), \eta_d(t), \psi_d(t), \xi_d^T(t), \eta_d^T(t), \psi_d^T(t)\) be the desired position, velocity, attitude, and angular velocity vectors, respectively. The control objective is then to find control laws \(u = \alpha(\xi, \xi_d, v, \eta_d)\) and \(\tau = \beta(\eta, \eta_d, \dot{\eta}, \dot{\eta_d})\) such that the tracking errors \(\chi = (\xi - \xi_d, v - \eta_d)\in \mathbb{R}^6\) and \(\epsilon = (\eta - \eta_d, \dot{\eta} - \dot{\eta_d})\in \mathbb{R}^6\) converge asymptotically to zero.

Remark 2. In fully autonomous mode, only the reference trajectories \([\xi_d, \eta_d, \psi_d]\) are given by the operator or computed by some high-level guidance system. The reference angles \([\phi_d, \theta_d, \psi_d]\) and their derivatives are computed by the outer-loop controller. The proposed control architecture also allows semi-autonomous flight (attitude tracking control) in which the reference angles \([\phi_d, \theta_d, \psi_d]\) are directly given by the operator through the GCS software or RC transmitter.

Proposition 1. The trajectory tracking control problem for rotorcraft UAVs represented by dynamical model (14) can be formulated as the control of two linear subsystems that are coupled
by a nonlinear term $\Delta(u, \eta_d, e_\eta)$, that is,

$$
\dot{x} = A_1 x + B_1 (\mu - \xi_d) + \frac{1}{m} H(\eta_d, e_\eta),
$$

$$
\dot{\xi} = A_2 e + B_2 (\bar{\eta} - \bar{\eta}_d),
$$

where the vector $H(\eta_d, e_\eta) \in \mathbb{R}^6$ represents dynamic inversion errors and the matrices $A_1 \in \mathbb{R}^{6 \times 6}$, $B_1 \in \mathbb{R}^{6 \times 3}$, $A_2 \in \mathbb{R}^{6 \times 6}$, and $B_2 \in \mathbb{R}^{6 \times 3}$ are defined as

$$
A_1 = A_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_1 = B_2 = \begin{bmatrix} 0 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 1 \\
0 & 0 \end{bmatrix}.
$$

Proof. Let us define a virtual or intermediary control vector $\mu \in \mathbb{R}^3$ as follows:

$$
\mu = \{u, \phi_d, \theta_d, \psi_d\} = -\frac{1}{m} R(\phi_d, \theta_d, \psi_d)e_3 + g e_3,
$$

(17)

where $R : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a continuous invertible function. Physically, the control vector $\mu$ corresponds to the desired force vector. Its magnitude is the total thrust $u$ produced by the propellers, and its orientation is defined by the body attitude $(\phi, \theta, \psi)$, $(\phi_d, \theta_d, \psi_d)$, and $\psi_d$ in Eq. (17) are thus the desired roll, pitch, and yaw angles.

By recalling Eq. (17), the components of $\mu$ are given by

$$
\mu_x = \frac{1}{m} u (\cos \phi_d \sin \theta_d \cos \psi_d + \sin \phi_d \sin \psi_d),
$$

$$
\mu_y = \frac{1}{m} u (\cos \phi_d \sin \theta_d \sin \psi_d - \sin \phi_d \cos \psi_d),
$$

$$
\mu_z = \frac{1}{m} u \cos \theta_d \cos \phi_d + g.
$$

(18)

($\mu_x$, $\mu_y$, $\mu_z$) are the force vector components along $X$-$Y$-$Z$ axes that are needed for tracking some reference trajectory. These desired control inputs are computed by the outer-loop controller. They are then used to compute the desired force vector magnitude and desired attitude angles, $(u, \phi_d, \theta_d) = f^{-1}(\mu_x, \mu_y, \mu_z)$. From Eqs. (18), one can solve for $u$ and $(\phi_d, \theta_d)$ as

$$
u = m \sqrt{\mu_x^2 + \mu_y^2 + (g - \mu_z)^2},
$$

$$
\phi_d = \sin^{-1}\left(\frac{-\mu_x \sin \psi_d - \mu_y \cos \psi_d}{\nu}\right),
$$

$$
\theta_d = \tan^{-1}\left(\frac{-\mu_x \cos \psi_d + \mu_y \sin \psi_d}{g - \mu_z}\right).
$$

(19)

Because the desired angles $(\phi_d, \theta_d, \psi_d)$ are the outputs of the orientation subsystem, they cannot be assigned or provided instantaneously. They are thus considered as reference trajectories for the inner-loop controller. By replacing $(\phi, \theta, \psi)$ in Eqs. (14) by $(\phi_d + \epsilon_\phi, \theta_d + \epsilon_\theta, \psi_d + \epsilon_\psi)$ and recalling Eqs. (18), the translational dynamics can be expressed as follows:

$$
\dot{\xi} = \mu + \frac{1}{m} h(\eta_d, e_\eta),
$$

(20)

where $e_\eta = (\eta - \eta_d)$ is the attitude tracking error vector. The components of the coupling term $h = (h_x, h_y, h_z)^T$ can be computed in the same manner as in Kendoul, Fantoni, and Lozano (2008).

Now, the cascaded system (15) is obtained by computing the time derivative of both position and attitude tracking errors $(\chi, e)$ and recalling Eq. (20).

To synthesize the control laws $\mu = \alpha(\chi, \bar{\xi}_d)$ and $\bar{\tau} = \beta(e, \bar{\eta}_d)$ for connected system (15), we will use the following theorem expressed by Sontag (1988):

**Theorem 1.** If there is a feedback $\mu = \alpha(\chi, \bar{\xi}_d)$ such that $\chi = 0$ is an asymptotically stable equilibrium of $\dot{\chi} = f(\chi, \alpha(\chi, \bar{\xi}_d), \bar{\xi}_d)$, then any partial state feedback control $\bar{\tau} = \beta(e, \bar{\eta}_d)$, which renders the $e$-subsystem equilibrium $e = 0$ asymptotically stable, also achieves asymptotic stability of $(\chi, e) = (0, 0)$. Furthermore, if the two subsystems are both globally asymptotically stable (GAS), then, as $t \rightarrow \infty$, every solution $(\chi(t), e(t))$ either converges to $(\chi, e) = (0, 0)$ (GAS) or is unbounded.

This theorem states that partial-state feedback design can be applied to control system (15) by synthesizing two independent controllers $\mu = \alpha(\chi, \bar{\xi}_d)$ and $\bar{\tau} = \beta(e, \bar{\eta}_d)$. In this control design, the interconnection term $\Delta(\mu, \eta_d, e_\eta)$ acts as a disturbance on the $\chi$-subsystem, which must be driven to zero.

Because the $\chi$- and $e$-subsystems in system (15) are linear, we can use simple linear controllers such as PD or PID. Therefore, we choose

$$
\mu = -K_\chi \chi + \bar{\xi}_d, \quad K_\chi \in \mathbb{R}^{3 \times 6},
$$

$$
\bar{\tau} = -K_e e + \bar{\eta}_d, \quad K_e \in \mathbb{R}^{3 \times 6},
$$

(21)

such that the matrices $A_\chi = A_1 - B_1 K_\chi$ and $A_e = A_2 - B_2 K_e$ are Hurwitz.

By substituting Eqs. (21) into Eqs. (15), the closed-loop system dynamics is given by

$$
\dot{\chi} = A_\chi \chi + \Delta(\chi, e_\eta),
$$

$$
\dot{e} = A_e e.
$$

(22)

Although $A_\chi$ and $A_e$ are Hurwitz, the global asymptotic stability of closed-loop system (22) cannot be directly deduced because of the interconnection term $\Delta(\chi, e_\eta)$. According to Theorem 1, one should prove that the trajectories...
\[ \chi(t), \epsilon(t) \] are bounded in order to guarantee the GAS for closed-loop system (22).

5.3. Stability Analysis of the Complete Closed-Loop System

One of the major tools usually used to show the boundedness of connected system trajectories is the input-to-state-stability (ISS) property (Sontag, 1988). The ISS property is a strong condition that is often difficult to verify. In Kendoul et al. (2008) and Kendoul, Zhenyu, and Nonami (2009), we have proposed a theorem that renders the stability analysis of a class of connected systems less complex. That theorem is recalled below, and its proof can be found in Kendoul et al. (2008).

Theorem 2. Let \( \bar{\tau} = \beta(\epsilon, \eta_d) \) be any \( C^1 \) partial-state feedback such that the equilibrium point \( e = 0 \) is GAS and locally exponentially stable (LES). Suppose that there exist a positive constant \( c_1 \) and one class-\( K \) function \( \gamma(\|e_n\|) \), differentiable at \( \|e_n\| = 0 \), such that

\[ \|x\| \geq c_1 \Rightarrow \|\Delta(x, e_n)\| \leq \gamma(\|e_n\|) \|x\|. \]  (23)

If there exist a positive semidefinite radially unbounded function \( V(x) \) and positive constants \( c_2 \) and \( c_3 \) such that for \( \|x\| \geq c_2 \)

\[ \frac{\partial V}{\partial x} f(x, \alpha(x, \tilde{\epsilon}_d), \tilde{\epsilon}_d) \leq 0, \]  (24)

then the feedback \( \bar{\tau} = \beta(\epsilon, \eta_d) \) guarantees the boundedness of all the solutions of Eqs. (22). Furthermore, if \( x = f(x, \alpha(x, \tilde{\epsilon}_d), \tilde{\epsilon}_d) \) is GAS, then the equilibrium point \( (x, e) = (0, 0) \) is GAS.

Here, we apply Theorem 2 in order to prove the global asymptotic stability of connected closed-loop system (22).

Proposition 2. Closed-loop system (22) satisfies all the conditions of Theorem 2, which implies that all the trajectories \( \{\chi(t), \epsilon(t)\} \) are bounded and the equilibrium point \( (\chi, e) = (0, 0) \) is GAS.

Proof. Because \( A_\chi \) and \( A_e \) are Hurwitz, the \( \chi \)-subsystem (without the interconnection term) and the \( e \)-subsystem are globally exponentially stable (GES), which is stronger than the GAS property. The GES of the \( \chi \)-subsystem implies that there exist a positive definite radially unbounded function \( V_\chi(x) \) and positive constants \( c_1 \) and \( c_3 \) such that for \( \|x\| \geq c_2 \)

\[ \frac{\partial V_\chi}{\partial x} A_\chi x \leq 0 \]  and \( \|\frac{\partial V_\chi}{\partial x}\| \|x\| \leq c_3 V_\chi(x) \).

Therefore, condition (24) of Theorem 2 is satisfied. Now, it remains to show that the interconnection term \( \Delta(x, e_n) \) satisfies growth restriction (23) of Theorem 2, and this is done in Kendoul et al. (2008) and Kendoul, Zhenyu, et al. (2009).

Remark 3. The final control inputs \( (u, \tau_\phi, \tau_\theta, \tau_\psi) \) are computed using Eqs. (13) and (19), which are nonlinear and consider system’s nonlinearities and coupling between state variables:

\[ u = m \| \mu(\chi, \tilde{\epsilon}_d) - g e_3 \| = m \| - K_\chi \chi + \tilde{\xi}_d - g e_3 \|, \]

\[ \tau = J \Psi(\eta) \tilde{\tau} + \Phi^T C(\eta, \eta) \eta = J \Psi(\eta)(-K_e e + \tilde{\eta}_d) + \Phi^T C(\eta, \eta) \eta. \]  (25)

Note that the nonlinear attitude controller in Eqs. (25) acts as a linear controller at the near-hovering condition where the attitude angles are small. However, in contrast to linear controllers that fail to provide good performance at non-nominal conditions, the nonlinear terms in Eqs. (25) become significant when the angles are big and participate positively to control the attitude, yielding higher tracking accuracy and robustness even at relatively big angles (see Figure 6).

Remark 4. In practice, the implementation of controller (25) requires some consideration. First, intermediary control laws (21) have been slightly modified to include an integral term, thereby increasing the tracking accuracy without destroying the GAS property of the closed-loop system. Second, the reference angles \( [\phi_d(t), \theta_d(t), \psi_d(t)] \) are processed using second-order low-pass digital filters to reduce noise and to compute both first and second derivatives. For effective and robust control in most cases, one also has to pay attention to many other things, such as integral terms resetting when the vehicle is on the ground (before takeoff and after landing) or also when a waypoint is reached (for the outer-loop controller).

Remark 5. The gain matrices \( (K_\chi, K_e) \) in Eqs. (25) were tuned empirically based on observing flight properties during experimental tests. Because the controller gains appear in the linear part of the controller, it was easy to adjust the gains despite the absence of accurate values for the plant parameters. Table II shows the controller gains used during flight tests.

The block diagram of the overall controller is shown in Figure 7.

6. FLIGHT TESTS AND EXPERIMENTAL RESULTS

To evaluate the performance of our autonomous rotorcraft MAV, we have performed real-time flight tests with various mission scenarios. Here, we present experimental results from six flight tests that demonstrate the autonomous capabilities of our vehicle including accurate attitude tracking, automatic takeoff and landing, long-distance flight, waypoint navigation, trajectory tracking, and vision-based flight. The objective of these flight tests is to show that the developed rotorcraft, equipped with the designed autopilot, is capable of achieving autonomously the search and rescue mission described in Section 2.
Figure 6. Performance of the inner-loop nonlinear controller during attitude trajectory tracking. The vehicle is able to track attitude commands with high accuracy even at relatively big and rapidly varying angles. In translational flight (70–100 s), the tracking error increased slightly because the pitch and roll dynamics are sensitive to aerodynamic effects such as blade flapping.

Figure 7. Structure of the inner–outer-loop nonlinear controller.
Table II. Controller gains.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{px}, k_{py}$</td>
<td>0.8</td>
</tr>
<tr>
<td>$k_{ix}, k_{iy}$</td>
<td>0.02</td>
</tr>
<tr>
<td>$k_{dx}, k_{dy}$</td>
<td>1</td>
</tr>
<tr>
<td>$k_{p_i}$</td>
<td>0.8</td>
</tr>
<tr>
<td>$k_{i}$</td>
<td>0.02</td>
</tr>
<tr>
<td>$k_{d_i}$</td>
<td>1</td>
</tr>
<tr>
<td>$k_{p_y}, k_{p_0}$</td>
<td>28</td>
</tr>
<tr>
<td>$k_{iy}, k_{iy}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$k_{dy}, k_{dy}$</td>
<td>1</td>
</tr>
<tr>
<td>$k_{p_b}$</td>
<td>3</td>
</tr>
<tr>
<td>$k_{i_b}$</td>
<td>0.05</td>
</tr>
<tr>
<td>$k_{d_b}$</td>
<td>0.2</td>
</tr>
</tbody>
</table>

6.1. Attitude Trajectory Tracking

In this flight test, we conducted attitude flight control in order to explore the effectiveness and robustness of the inner-loop nonlinear controller. For best evaluation of the attitude controller performance, we performed an outdoor test in which reference trajectories are generated in the following manner:

1. Between 0 and 25 s, preprogrammed sinusoidal trajectories are generated with 0.5-Hz frequency and a time-varying magnitude. The pitch angle magnitude increases from 0 to 45 deg and the roll angle magnitude is set to 0 deg (0–22 s), then to 30 deg (22–24 s), and finally to 0 deg (24–25 s).

2. Between 25 and 110 s, the reference trajectories are sent by the operator via the RC transmitter (semiautonomous control), such that the induced forward velocities are relatively high (about 4 m/s).

The attitude control results are shown in Figure 6, where we can see the rotorcraft accurately tracking the reference commands. Good tracking was obtained even at relatively big and rapidly varying angles. Furthermore, the controller handles coupling between pitch and roll axes even when the angles are big.

At relatively high forward speeds, the controller is also able to track reference commands without apparent degradation in performance, as shown in Figure 6 (70–100 s). The small tracking errors are due to aerodynamic effects, such as blade flapping and the pitching-up phenomenon, that occur at relatively high speeds and significant angles of attack.

6.2. Automatic Takeoff, Hovering, and Landing

The nonlinear controller given by Eq. (25) is used in this flight test to achieve automatic takeoff, accurate hovering, and precise autolanding. The experimental results, shown in Figure 8, demonstrate accurate tracking of the height reference command, yielding to effective altitude control and automatic takeoff and landing. The rotorcraft also achieved a stable hovering flight and was able to stay inside a 50-cm-radius circle. The horizontal motion is also accurately controlled during takeoff and landing maneuvers with less than 1-m error, which is a good performance for this scale rotorcraft flying outdoors and subject to external disturbances such as wind.

6.3. Long-Distance Autonomous Flight

This flight test was performed in March 2008 at Agra, India, during the U.S.-Asian MAV competition. The objectives of this test were to do the following:

1. Demonstrate the capability of our MAV to fly autonomously until the zone of interest, located at about 1 km from the launching point.

2. Check the quality and range of the wireless communication as well as the video transmission.

The rotorcraft was thus tasked to achieve an autonomous forward flight at a translational velocity\(^3\) of 4.25 m/s while transmitting images to the GCS.

The obtained results are shown in Figure 9, where we can see that the velocity and attitude reference trajectories are well tracked. Position trajectories show that the MAV flew a relatively long distance autonomously, which is about 1.2 km ($\sqrt{x^2 + y^2}$). When the MAV reached the limits of the fly zone, the safety pilot switched to manual flight and recovered the vehicle. The test also showed that the range of the wi-fi wireless communication was about 600 m, whereas the quality of the video transmission is acceptable until 1,000 m. In this test, the safety procedure related to communication lost was disabled, thereby allowing the vehicle to continue mission execution even when the communication link is lost.

6.4. Fully Autonomous Waypoint Navigation

Here, we demonstrate the ability of the GN&C system to achieve accurate waypoint navigation and to perform hovering and automatic takeoff and landing. In this flight test, a set of four waypoints\(^4\) were chosen by just clicking the desired locations on the two-dimensional (2D) map of the GCS interface (see Figure 10). The MAV should then pass the assigned waypoints in a given sequence. This flight test simulates a reconnaissance mission in which the UAV is tasked to fly some target areas for information collection.

\(^3\) $V = \sqrt{V_x^2 + V_y^2} = \sqrt{3^2 + 3^2} = 4.25$.

\(^4\) The current GN&C system allows an unlimited number of desired waypoints with more than 1,000 waypoints that can be sent at one time from the GCS.
Figure 8. Experimental results from an outdoor flight test in which the quadrotor achieved a fully autonomous takeoff, hovering, and landing. The rotorcraft was able to stay inside a 50-cm-radius circle during hovering. The attitude reference trajectories were accurately tracked with an error of less than 0.5 deg during hovering.

The mission is achieved autonomously when the operator just has to send high-level commands through the GCS interactive interface. The mission is started by clicking the “takeoff” button. The MAV then performs an automatic takeoff and hovers at a 10-m height. The altitude can be changed at any time from the GCS. When the “upload waypoint” button is pushed, the MAV starts waypoint navigation. When the mission is finished, the “land” button can be pushed for automatic landing.

In Figure 10, we can see the selected waypoints as well as the MAV trajectory plotted in real time. We can clearly observe that the MAV passed successfully through all the waypoints. Position and velocity reference trajectories are tracked with high accuracy, as shown in Figures 11–12. The observed small tracking errors may be attributed to wind gust disturbances and to GPS data errors (about ±2 m) and latency (about 0.8 s).

The automatic takeoff and landing capabilities are also demonstrated in this flight test with accurate altitude control even at relatively high horizontal speeds of 5 m/s. It is interesting, however, to note that altitude control is more accurate at low forward speeds as shown in Figure 8 and later in Figure 18 (about 50-cm maximum error). Indeed, the thrust variation created by different angles of attack at varying forward speeds and wind conditions causes a disturbance that pushes the vehicle above the reference height. The controller takes some time to reject these disturbances because of the time delay in thrust.

Figure 13 shows that the inner-loop controller performs well and the desired angles are tracked with small errors. As discussed in Subsection 6.1., degradation in the attitude tracking capabilities as forward speed increases are observed in this flight test.

Waypoints can be updated and/or modified in real time.

6We believe that this time delay is introduced voluntarily in the original platform to facilitate manual flight.
A video clip of this flight test can be found at http://www.youtube.com/watch?v=Lo9qJz69uuQ&feature=channel_page.

6.5. Arbitrary Trajectory Tracking

The performance of the autopilot for trajectory tracking is an important evaluation. The final objective of the rotorcraft MAV is to perform autonomously search, rescue, and surveillance missions. Thus, trajectory tracking capability is very useful because a spiral trajectory following, for example, allows the MAV to explore some area of interest. Furthermore, accurate trajectory tracking is required for precise maneuvers in cluttered environments in order to avoid obstacles, for example.

In this flight test, a spiral trajectory was implemented to demonstrate the tracking performance of the nonlinear controller. The reference trajectory is generated using a kinematic model of a modified Archimedean spiral in order to obtain a spiral with constant separation distance (10 m) between successive turnings but also with a constant tangential speed (3 m/s).

Results from a flight test in which the MAV autonomously executes a spiral trajectory tracking are shown in Figures 14 and 15 (see also Figure 10). One can see that the MAV tracked successfully the reference trajectories including the height trajectory for automatic takeoff and altitude holding. Figure 14 also shows the effectiveness and importance of relying on PS/INS rather than GPS/INS for height control, especially for MAVs flying at a low altitude where accurate height control is necessary.

Attitude control results, shown in Figure 16, confirm the good performance of the inner-loop controller even in this complicated motion pattern. One thing to note in the attitude control results, shown in Figures 13 and 16, is the significant errors in yaw control compared to pitch and roll control. This is mainly due to the internal yaw controller implemented in the X-3D board, which affects the performance of our yaw controller (see Subsection 3.1.).

In this flight test, we have also demonstrated the flight termination system. The MAV achieved a soft emergency landing without damaging the vehicle, as can be seen in Figure 14 and the associated video clip, http://www.youtube.com/watch?v=r4eOUDA3Jo&feature=channel_page.
Figure 10. Part of the GCS interface showing a 2D map of the test field and the MAV trajectories during waypoint navigation and spiral trajectory tracking.

Figure 11. Autonomous waypoint navigation experiment. In this test, the quadrotor vehicle is commanded to perform automatic takeoff, waypoint navigation, stationary flight at a 10-m height, and automatic landing. In this test, the flight path is defined by four waypoints and the reference trajectories are generated in real time by the guidance system. These results show that the rotorcraft passed successfully through all the waypoints and reference trajectories are accurately tracked along the 350-m flight path.
Figure 12. MAV translational velocities during waypoint navigation. The velocity reference trajectories are well tracked at low and relatively high speeds (5 m/s). Tracking errors in the vertical velocity may be attributed to the fast dynamics of the vertical motion and noisy measurements, which are obtained by differentiating the PS signal.

Figure 13. MAV attitude during waypoint navigation. Good tracking of commanded attitude is obtained even in the presence of wind gust disturbances. At high translational speeds, aerodynamic disturbances become significant, giving tracking errors on the order of 3–5 deg.
Figure 14. Tracking a spiral trajectory outdoors at a constant forward speed of 3 m/s. In this test, the quadrotor MAV is commanded to take off and fly a spiral trajectory at a constant speed while maintaining a constant height of 10 m and a constant separation distance (10 m) between successive turnings. Notice that the controller relies on the PS estimates for height control, which are more accurate than the GPS measurements. These results demonstrate that the designed nonlinear controller enables miniature rotorcraft to accurately track nonstraight trajectories and to execute different flight patterns.

Figure 15. MAV translational velocities during spiral trajectory tracking. The time-parameterized desired velocities are generated by the guidance system in such a way as to maintain a constant speed of 3 m/s along a spiral pattern.
6.6. Vision-Based Flight and Autonomous Target Tracking

This flight test demonstrates the possibility of the developed platform locking and tracking some ground moving target using vision. We have placed a box of about 70 × 50 cm at approximately 16 m from the GCS. This box serves as a target that must be tracked by the MAV.

In the first phase of the experiment, the rotorcraft was relying on GPS measurements to control its motion, simulating a search maneuver. When the target occurs in the field of view (FOV) of the camera, the operator selects that target on the image window displayed in real time on the GCS screen. By pushing “TGT mode” button (see Figure 17), the flight mode is switched into vision-based flight. The vision system provides real-time estimates about
Figure 18. Autonomous vision-based flight for moving target tracking. In the first phase, the vehicle is commanded to take off and hover at a 6-m height using GPS measurements. When the target occurs in the camera FOV, the flight mode is switched to vision-based flight and the vehicle is commanded to track the target by controlling the relative distance (black line) to zero. The vehicle flew about 16 m along the X axis, as shown in the first graph (blue line), tracking a ground moving target before landing on it using vision. For best comparison of GPS measurements and visual estimates in the Y axis, GPS drift was compensated at $t = 180$ s by adding 2 m to its Y measurement.

the vehicle’s position, velocity, and height relative to the selected target. The tracking process is performed by controlling this relative pose in order to regulate it to zero.

The target was displaced by pulling some wire attached to its center. To demonstrate the good performance of the tracking system, the target was moved for some time and then stopped and then moved again. Finally, the MAV was commanded to achieve a precise autolanding on the target using visual estimates.

As we can see in Figure 18, the transition from the GPS mode to the vision mode was performed stably and smoothly without any oscillation or aggressive maneuver. The GPS ground-track (blue line) in the first graph of Figure 18 shows that the MAV flew a 16-m distance while keeping the relative position (black line), estimated by the vision system, near zero. This means that the MAV tracked effectively the moving target and kept it in its FOV, ideally at the image center.

The height was also controlled accurately by holding a desired altitude of 6 m. Takeoff and landing maneuvers were performed autonomously, resulting in a precise landing\(^7\) on the target using visual measurements (see Figure 19).

A video clip of this flight test can be found at http://www.youtube.com/watch?v=-IpbOd-UuG4. Other videos about vision-based flights are available at http://www.youtube.com/fkendoul.

The flight results presented in this paper demonstrate a significant improvement in capability over previous minirotorcraft and quadrotor MAVs in particular. From the obtained experimental results according to six different mission scenarios, we conclude that the implemented autopilot and GN&C algorithms present good performance for autonomous flight. In contrast to standard linear controllers that fail to provide good performance at nonnominal conditions, the designed nonlinear control system allowed a miniquadrotor to accurately track reference

\(^7\)The rotorcraft landed on the edge of the 70 × 50 cm box, which is a very good performance for vision-based landing using a miniquadrotor UAV.
trajectories even at relatively high speeds, big angles, coupled motions, and windy conditions. These results also demonstrate that rotorcraft MAVs, weighing less than 0.7 kg, can achieve advanced autonomous flight despite their small size and the weaknesses of the low-cost sensors.

7. CONCLUSIONS
In this paper, we have described the design of a minirotorcraft UAV with onboard intelligence capabilities that can be used for search and rescue missions. This research dealt with both theoretical and practical issues of autonomous RMAVs, ranging from controller stability analysis to experimental flight tests. Indeed, we have proposed a nonlinear flight controller that is practical for real-time implementation while guaranteeing the global asymptotic stability of the closed-loop system. We have also demonstrated the ability to provide effective waypoint navigation, accurate trajectory tracking, and visual navigation capabilities to small low-cost systems such as the mini-quadrotor helicopter. This proves the relevance of the GN&C system, which relies solely on light weight and low-cost sensors. From the flight test results, we conclude that the developed rotorcraft MAV shows good and reliable performance for autonomous flight, which makes it ready to embark on many applications including search and rescue missions. This constitutes an important step toward our goal of developing fully autonomous MAVs capable of achieving real-world applications.

In future work, it would be interesting to improve the control system by considering the aerodynamic effects at higher speeds and aggressive maneuvers. It may also be useful to increase the vehicle autonomy by developing guidance systems that are able to tackle a number of operational events without operator intervention such as, for example, trajectory replanning following the detection of an obstacle or changes in the mission or environment configuration. Currently, we are developing a GPS-aided visual system for cooperative navigation and control of several minirotorcraft UAVs. We are also investigating different technologies for obstacles and collision avoidance in single- and multi-UAV platforms.

ACKNOWLEDGMENTS
This work was supported by the Japan Society for the Promotion of Science (JSPS).

REFERENCES


