Chapter 7: Obstacle and Terrain Avoidance for Miniature Aerial Vehicles[1]

This Chapter presents an approach to miniature aerial vehicle (MAV) obstacle and terrain avoidance that has been developed at Brigham Young University (BYU). The research builds on the notion of utilizing useful but imperfect map information to plan nominal paths through city or mountain terrain. Because maps may be limited in resolution, out of date, or offset in location, MAVs must also utilize sensory information to detect and avoid obstacles unknown to the path planner. In this Chapter, a laser range finder and optic flow sensors are utilized to detect obstacles and terrain. Avoidance algorithms using this sensor information are discussed briefly and flight test results using the BYU MAVs are presented.

7.1 Introduction

Unmanned aerial vehicles (UAVs) are playing increasingly prominent roles in defense programs and strategy around the world. Technology advancements have enabled the development of large UAVs (e.g., Global Hawk, Predator) and the creation of smaller, increasingly capable UAVs. The focus of this Chapter is on smaller fixed-wing miniature aerial vehicles (MAVs), which range in size from ¼ to 2 m in wingspan. As recent conflicts have demonstrated, there are numerous military applications for MAVs including reconnaissance, surveillance, battle damage assessment, and communications relays.

Civil and commercial applications are not as well developed, although potential applications are extremely broad in scope. Possible applications for MAV technology include environmental monitoring (e.g., pollution,

[1] Written by S. Griffiths, J. Saunders, A. Curtis, B. Barber, T. McLain, R. Beard. This work was funded by AFOSR award numbers FA9550-04-1-0209 and FA9550-04-C-0032. This Chapter is the outgrowth of a previously published paper by the authors in the IEEE Robotics and Automation Magazine, Vol. 13, No. 3, 34-43, September 2006.

Kimon P. Valavanis (ed.), Advances in Unmanned Aerial Vehicles, 213–244.

weather, and scientific applications), forest fire monitoring, homeland security, border patrol, drug interdiction, aerial surveillance and mapping, traffic monitoring, precision agriculture, disaster relief, ad-hoc communications networks, and rural search and rescue. For many of these applications to develop to maturity, the reliability of MAVs will need to increase. In addition, their capabilities will need to be expanded and their ease of use will need to be improved. In addition to these technical challenges, the regulatory challenge of integrating UAVs into the national and international air space needs to be overcome.

Critical to the more widespread use of MAVs is making them easy to use by non-pilots including scientists, forest fire fighters, law enforcement officers, or military ground troops. One key capability for facilitating ease of use is the ability to sense and avoid obstacles, both natural and man made. Many potential applications require MAVs to fly at low altitudes in close proximity to structures or terrain. For example, the ability to fly through city canyons and around high-rise buildings is envisioned for future homeland security operations. For MAVs to be effective tools, the challenge of operating in complex environments must be automated, allowing the operator to concentrate on the task at hand.

Performing obstacle and terrain avoidance from a fixed-wing MAV platform is challenging for several reasons. The limited payload and power available on MAV platforms place significant restrictions on the size, weight, and power requirements of potential sensors. Sensors such as scanning LADAR and RADAR are typically too large and heavy for MAVs. Related to limits on sensor payload are those on computing resources. For most MAVs, the primary computational resource is the autopilot microcontroller. Additional computational hardware can be integrated in the MAV, but computers such as PC104-based systems generally exceed the payload capacity and so smaller microcontrollers are typically used.

Another challenge posed by fixed-wing MAVs is that they move fast: ground speeds are often in the range of 10 to 20 m/s (22 to 44 mph). Contrary to the computational limits imposed, obstacle avoidance algorithms must execute and act quickly. Unlike ground robots and unmanned rotorcraft, fixed-wing MAVs cannot stop or slow down while avoidance algorithms process sensor information or plan maneuvers. Reactions must be immediate. Obstacle sensing is further complicated by the fact that sensor readings are altered by changes in aircraft attitude, especially the rolling motions that occur during turns. Attitude changes affect not only the pointing direction of the sensor, but also cause motion of fixed objects in the field-of-view. Obstacle and terrain detection must account for the effects of aircraft attitude changes for avoidance maneuvers to be successful. All

of the challenges associated with MAV obstacle and terrain avoidance are compounded by the reality that for MAVs, mistakes are costly or even catastrophic, as crashes can result in damage to or loss of the MAV and failure to complete the objectives of the flight.

As evidenced by the recent DARPA Grand Challenge, capable obstacle avoidance and terrain navigation systems have been developed for ground vehicles. Obstacle avoidance and path planning have been active areas of research for many years and the associated robotics literature is immense. While providing a guiding influence, most of the proposed methods fail to deal with the sensing and computational challenges imposed by the limited payload capabilities of MAVs.

As autonomous MAVs and feasible obstacle sensors are recent technological developments, the body of experimental research directed specifically toward MAV obstacle and terrain avoidance is small. Related to terrain avoidance is work focused on utilizing vision processing techniques to estimate height above ground. In [1], it was demonstrated that mimicking the landing behavior of bees by maintaining constant optic flow during a landing maneuver, could be used to successfully control the descent of a MAV. Development of lightweight sensors for measurement of optic flow has enabled their use in MAVs [2] [3] [4]. In [5], it was demonstrated that these sensors can be used to follow undulations in terrain with low-flying MAVs.

The next Section describes the BYU aerial platforms.

7.2 BYU Miniature Aerial Vehicle Platforms

Over the past five years, BYU has been involved in the development of MAV airframes, autopilots, user interfaces, sensors, and control algorithms. This Section describes the experimental platform developed specifically for the obstacle avoidance research described next.

7.2.1 Airframe

Figure 7.1 shows the airframe used for obstacle avoidance experiments. The airframe has a 1.5 m wingspan and has been constructed with an EPP foam core covered with Kevlar. This design was selected for its durability, useable payload, ease of component installation, and flight characteristics. The airframe can carry a 0.4 kg payload and can remain in flight for over 45 minutes at a time. The collision avoidance sensors that are embedded in the airframe include three optic-flow sensors, one laser ranger, and two

electro-optical cameras as shown in Figure 7.2. Additional payload includes the Kestrel autopilot, batteries, a 1000 mW, 900 MHz radio modem, a 12-channel GPS receiver, and a video transmitter.

Fig. 7.1. Airframe used for collision avoidance experiments.

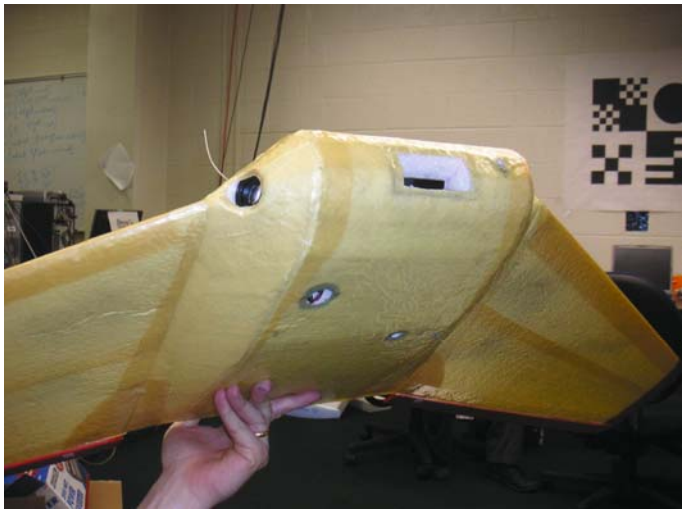

Fig. 7.2. Sensors used for collision avoidance. The round hole on the right and the large hole on the belly are the optic flow sensors. The square hole in the center is the laser ranger, and the other two round holes are for electro-optical cameras.

7.2.2 Kestrel Autopilot

The collision avoidance algorithms have been implemented on Procerus Technologies' Kestrel Autopilot version 2.2 [6], which is shown in Figure 7.3. The autopilot is equipped with a Rabbit 3400 29 MHz processor, three-axis rate gyros, three-axis accelerometers, absolute and differential pressure sensors, and a variety of interface ports. The autopilot measures $3.8 \times 5.1 \times 1.9$ cm and weighs 18 grams. The autopilot also serves as a data acquisition device and is able to log 175 kbytes of user-selectable telemetry at rates up to 60 Hz. The optic flow sensors and the laser ranger are connected directly to the autopilot and the collision avoidance algorithms are executed on-board the Rabbit processor.

Fig. 7.3. The Kestrel autopilot version 2.2. Figure courtesy of Procerus Technologies.

7.2.3 Ground Station

There are two key components of the ground station. The first is the virtual cockpit software that runs on a laptop. A screen shot of the virtual cockpit is shown in Figure 7.4. The virtual cockpit allows the user to configure every parameter on the autopilot during flight. In particular, autopilot and sensor gains can be configured in-flight. In addition, the virtual cockpit

was designed to allow TCP/IP connection to other software modules. The TCP/IP interface allows external programs to read telemetry data from the autopilot, and to access internal autopilot parameters. This architecture facilitates a variety of user interfaces. In particular, PDA and voice interfaces have been implemented using the virtual cockpit [18] [19].

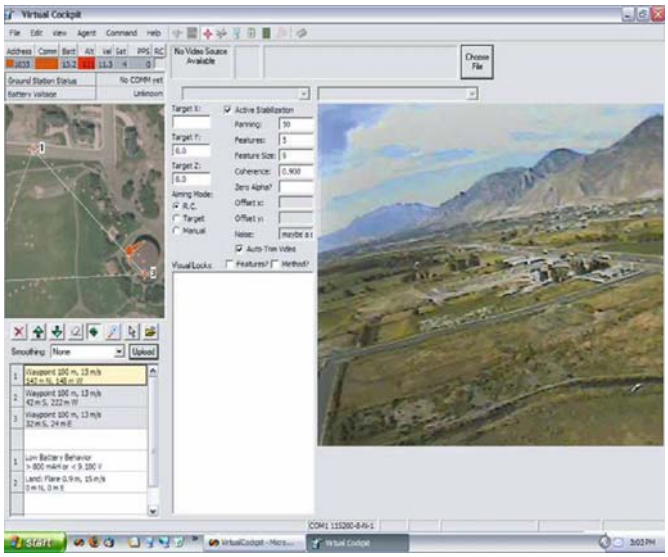

Fig. 7.4. Screen shot of the virtual cockpit used with the Kestrel autopilot. The window on the left is the mission planning window that allows waypoints to be dynamically specified. The window on the right displays the video stream from the MAV.

The second key component to the ground station is the communication box shown in Figure 7.5. The communication box contains a 900 MHz transmitter, a GPS unit, and an interface to an RC transmitter that can be used to maneuver the airframe manually. In addition to standard telemetry the video feed from the cameras is connected to an Imperx VCE-PRO PCMCIA frame grabber [20] hosted on the laptop. The frame grabber provides 640 X 480 RGB images at 30 frames-per-second. The image can be displayed in the virtual cockpit and processed for image in the loop applications.

Fig. 7.5. Communication box connected to the laptop. The communication box contains a 900 MHz transceiver, a GPS unit, and an interface to an RC transmitter which can be used as a stand-by fail-safe mechanism to facilitate safe operations. Figure courtesy of Procerus Technologies.

7.2.4 Optic Flow Sensors

The MAV is equipped with three optic-flow sensors. Two of the optic-flow sensors are forward looking but swept back from the nose by $\alpha = 60$ degrees. The third optic flow sensor points down to determine the height above ground. The optic-flow sensors, shown in Figure 7.6, are constructed by attaching a lens to an Agilent ADNS-2610 optical mouse sensor. The ADNS-2610 has a small form factor, measuring only 10 mm by 12.5 mm and runs at 1500 frames per second. It requires a light intensity of at least 80 mW/m^2 at a wavelength of 639 nm or 100 mW/m^2 at a wavelength of 875 nm. The ADNS-2610 measures the flow of features across an 18 by 18 pixel CMOS imager. It outputs two values, δp_x and δp_y, representing the total optic flow across the sensor's field-of-view in both the x and y directions. The flow data in the camera y direction corresponds to lateral motion of the MAV and is ignored.

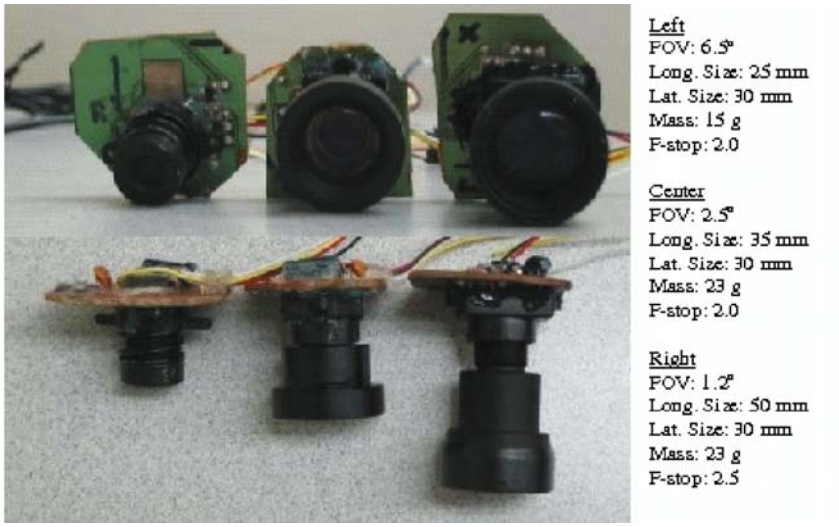

Fig. 7.6. Optic flow sensors with three different lens configurations: 1.2, 2.5, and 6.5 degree field-of-view. The optic flow sensors are constructed by attaching a lens to an optical mouse chip.

Figure 7.7 indicates how distance is computed using the optic flow sensor. The optical mouse chip outputs an optic flow displacement $(\delta p_x, \delta p_y)^T$ at its internal sample rate of 1500 Hz. Since the collision avoidance loop is executed at $T_s = 20$ Hz, the total optical displacement is integrated over T_s to produce $(\Delta p_x, \Delta p_y)$. The distance to the object D is related to the measured distance d by the expression:

$$D = d \cos\phi \sin\alpha, \tag{7.1}$$

where ϕ is the roll angle of the MAV. From geometry, the measured distance to the object is given by:

$$d = \frac{V_{\text{gps}} T_s}{\tan\left(\frac{\lambda_{\text{eff}}}{2}\right)}, \tag{7.2}$$

where λ_{eff} is the effective field-of-view. The effective field-of-view is given by:

$$\lambda_{\text{eff}} = \lambda_{\text{cam}} \frac{\Delta p_x}{P_x} - \dot{\chi} T_s, \tag{7.3}$$

where λ_{cam} is the field-of-view of the camera, P_x is the size of the pixel array along the direction of motion, and $\dot{\chi}$ is the yaw rate with respect to the ground. Using similar reasoning for left-looking and down-looking optic flow sensors we can derive the following expression for the right, left, and AGL distances:

$$D_{\text{right}} = \frac{V_{\text{gps}} T_s}{\tan\left(\frac{\lambda_{\text{cam}} D_{\text{right}} p_x}{2P_x} - \frac{\dot{\chi} T_s}{2}\right)} \cos\phi \sin\alpha \tag{7.4}$$

$$D_{\text{left}} = \frac{V_{\text{gps}} T_s}{\tan\left(\frac{\lambda_{\text{cam}} D_{\text{left}} p_x}{2P_x} + \frac{\dot{\chi} T_s}{2}\right)} \cos\phi \sin\alpha$$

$$D_{\text{down}} = \frac{V_{\text{gps}} T_s}{\tan\left(\frac{\lambda_{\text{cam}} D_{\text{down}} p_x}{2P_x} - \frac{\dot{\theta} T_s}{2}\right)} \cos\theta \cos\phi.$$

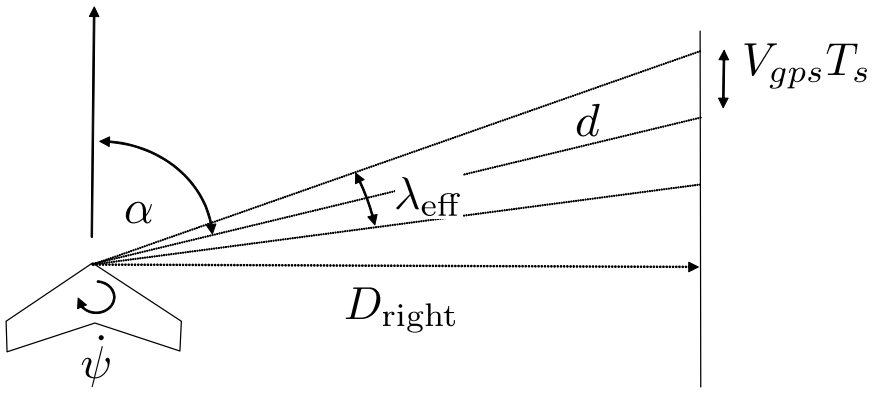

Fig. 7.7. The optic flow sensor is used to compute the distance to an obstacle based on the distance traveled between samples ($V_{\text{gps}} T_s$) and the effective field-of-view λ.

7.2.5 Laser Ranger

For the experiments discussed in this Chapter, the Opti-Logic RS400 Laser rangefinder shown in Figure 7.8 is used. The laser has a sensing range of 400 m with an update rate of 3 Hz. It weights 170 grams and consumes 1.8 W of power. Figure 7.2 shows the laser ranger mounted on the air-

frame. It is important to note that the RS400 is not a scanning laser range-finder. Scanning laser range finders are currently too heavy and consume too much power for MAV applications. The RS400 returns a single distance measurement and must be steered by maneuvering the airframe.

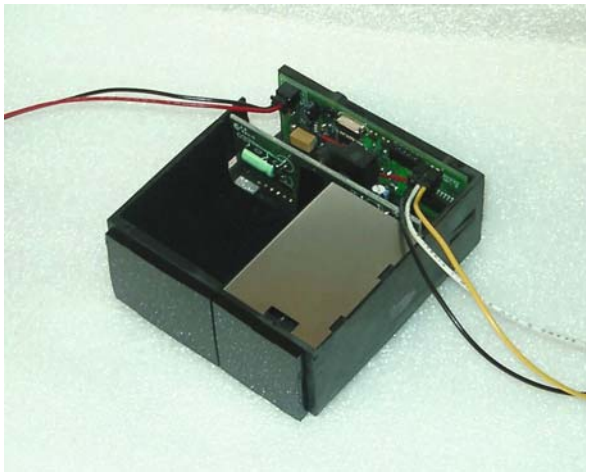

Fig. 7.8. The Opti-Logic RS400 laser range finder. The laser range finder was mounted as a strapdown unit on the airframe shown in Figures 7.1 and 7.2.

7.2.6 Signal Flow Diagram

The signal flow diagram for the system is shown in Figure 7.9. The Kestrel autopilot processes GPS, accelerometers, rate gyroscopes, and pressure sensors at a rate of approximately 80 Hz. The laser ranger is processed at a rate of about 3 Hz, and the optic flow sensors are processed at a rate of about 80 Hz. The autopilot implements standard autopilot functions, as well as the reactive path planner and the vector field guidance algorithm. Telemetry data is transmitted to the ground station via a 900 MHz digital transceiver. In addition, an on-board camera transmits video signals to the ground via a 2.4 GHz analog transmitter. The deliberative path planner described in the next section is implemented on the ground station.

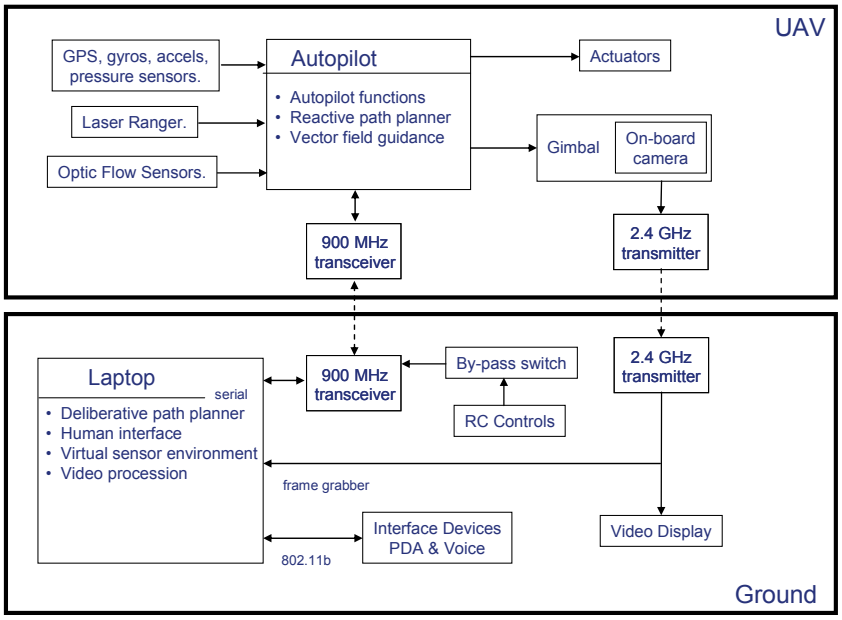

Fig. 7.9. The signal flow diagram for collision avoidance. The collision detection sensors are processed on-board the MAV and provide feedback to the reactive path planner.

7.3 Path Planning and Following

The first step for navigating through complex environments is to plan a nominal path based on known information about the environment, usually in the form of a street map or a topographic map. The MAV must be able to accurately follow the nominal path to avoid known obstacles. This section discusses the methods for planning and following the nominal path. Subsequent sections will discuss reactive, sensor-based obstacle avoidance strategies for maneuvering around obstacles that are unknown during the planning process.

7.3.1 Planning the Nominal Path

When planning paths through complex environments, the computational requirements for finding an optimal path can be significant and unrealistic for near-real-time execution [7]. Because of this, recent research has focused on randomized techniques to quickly find acceptable, though not

necessarily optimal, paths [8] [9]. Path planning for MAVs is also difficult because of the dynamic constraints of flight. Many common path planning algorithms are inadequate for fixed-wing MAV systems because they do not handle turn-radius limitations and airspeed constraints effectively.

One randomized method that addresses these limitations is the Rapidly-exploring Random Tree (RRT) algorithm [7] [10]. RRTs use a dynamic model of the system to build a tree of traversable paths. The search space is quickly explored by applying control inputs to configurations already in the tree. Working with the precise control inputs ensures that the dynamic constraints are not violated; however, it also results in an open-loop solution. This would be adequate if we had a perfect model of the system without disturbances, but this method is not satisfactory for an actual MAV because of model inaccuracies and environmental disturbances such as wind.

The basic idea of the Rapidly Exploring Random Tree (RRT) algorithm is to build a tree that uniformly explores the search space. The uniformity is achieved by randomly sampling from a uniform probability distribution. Figure 7.10 gives a graphical depiction the basic RRT algorithm. As shown in Figure 7.10 (a), the input to the RRT algorithm is a start waypoint w_{start}, an end waypoint w_{end}, and the terrain map. The first step of the algorithm is to randomly select a point p in the workspace. As shown in Figure 7.10 (b) a new waypoint w_1 selected a fixed distance D from w_{start} along the line $\overline{pw_{start}}$, and inserted into the tree. At each subsequent step, a random point p is generated in the workspace, and the tree is searched to find the node that is closest to p. As shown in Figure 7.10 (c) a new waypoint is generated that is a distance D from the closest node in the tree, along the line connecting p to the closest node. Before a waypoint path is added to the tree, it needs to be checked for collisions with the terrain. If a collision is detected, as shown in Figure 7.10 (d), then the waypoint is deleted and the process is repeated. When a new node is added, its distance from the end node w_{end} is checked. If it is less than D, then a waypoint path from w_{end} is added to the tree, and as shown in Fig. 7.10 (f), a complete path through the terrain has been found.

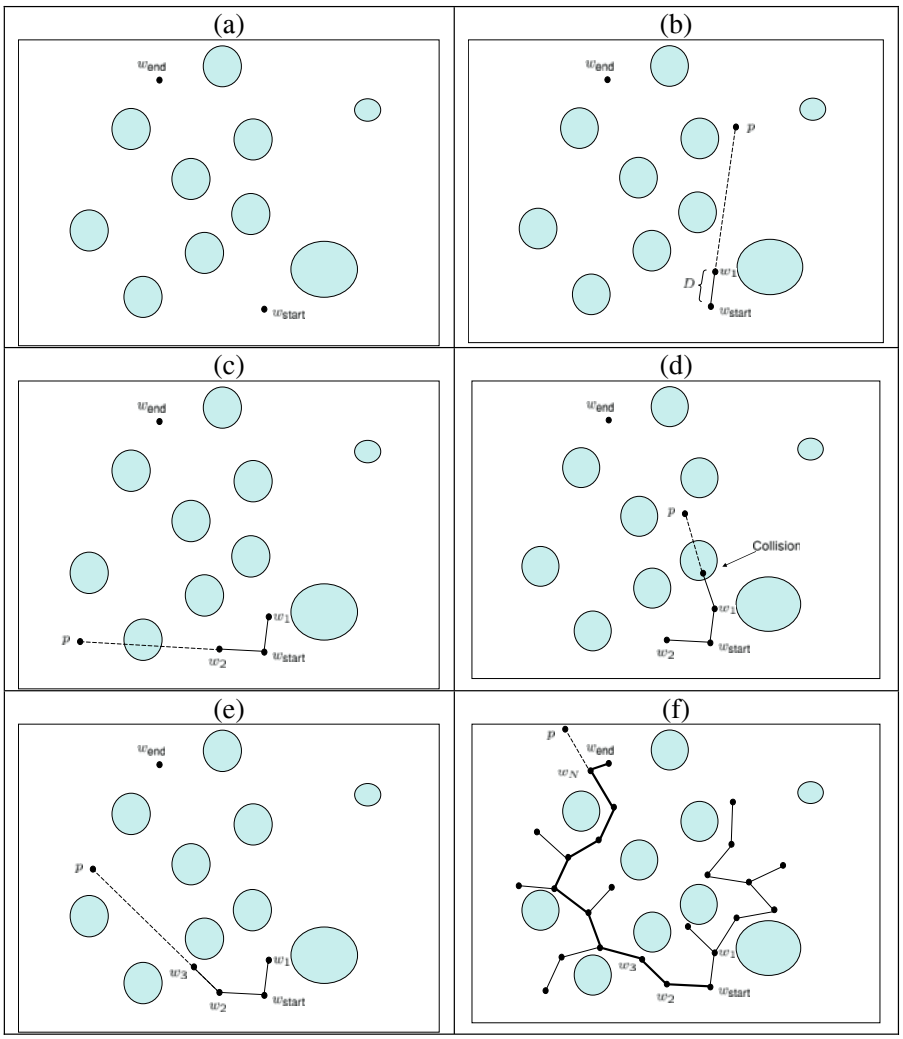

Fig. 7.10. This figure gives a graphical description of the basic RRT algorithm.

The basic algorithm described above is similar to the one presented in [11] in that paths are planned in the output space. The method is easily extended to three dimensions and provides a computationally efficient a priori path planner for the MAVs [12]. Since the MAV will not exactly follow the waypoint path, it is essential to bound the tracking error of the controlled MAV from the waypoint path. For a given waypoint path, one can determine the expected trajectory of the MAV [13] and ensure that only traversable paths are built into the search tree. Branches in the tree are

checked to ensure that they pass tests on turn radius and climb rate, and are collision-free. Figure 7.11 depicts the growth of a 3-D RRT path through a simulated urban environment, while real-time results through a canyon are presented in Section 7.5 (Figure 7.27).

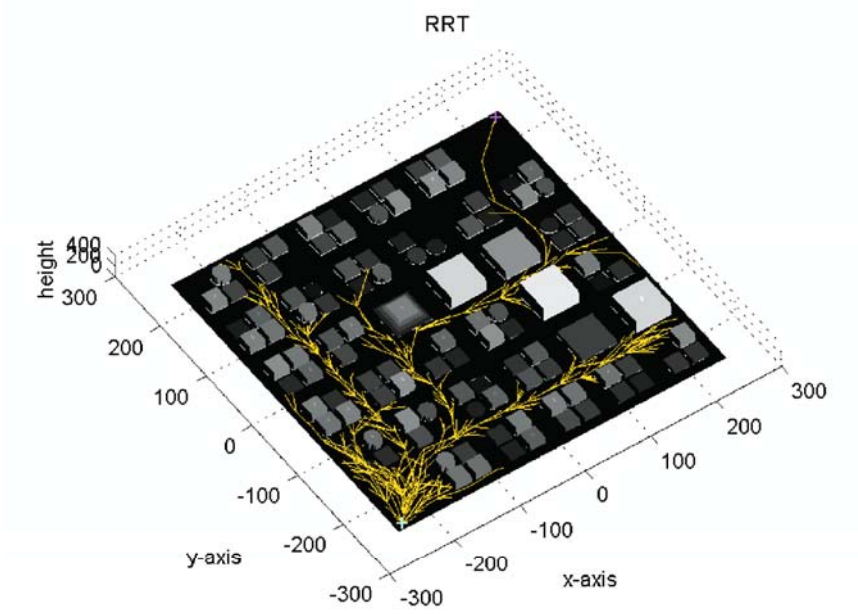

Fig. 7.11. This figure shows the growth of an RRT path tree through a simulated urban environment. The algorithm is terminated once a feasible path to the destination (red X) is found.

This algorithm is used to find nominal paths through different types of terrain. Planned paths through a simulated urban terrain and a canyon represented by USGS terrain data are shown in Figures 7.12 and 7.13, respectively.

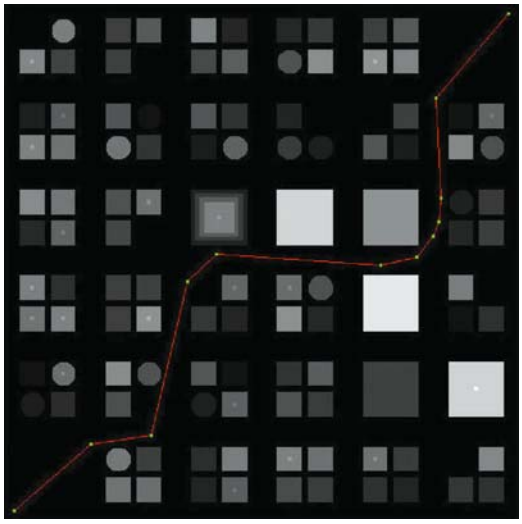

Fig. 7.12. This figure shows the result of the RRT algorithm applied to simulated urban terrain. The minimum turning radius of the MAV is 50 m, and the average street width is 30 m.

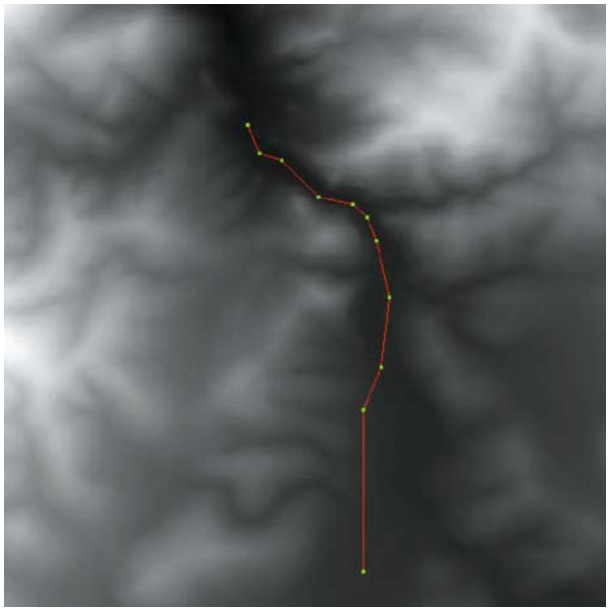

Fig. 7.13. This figure shows the result of the RRT algorithm applied to canyon data downloaded from a publicly available USGS database.

7.3.2. Vector Field Path Following

Given a nominal waypoint path, it is essential for the MAV to have the ability to track the path with precision. MAVs must track these paths despite dynamic limitations, imprecise sensors and controls, and wind disturbances, which are often 20% - 60% of airspeed [14]. Trajectory tracking, which requires the MAV to be at a specific location at a specific time, is difficult in such wind conditions. As an alternative, a path following approach is developed where the focus is simply to be on the path, instead of at a specific point that evolves in time. Similar research in [15] describes a maneuvering method focused on converging to the path then matching a desired speed along the path. The path following method is based on the creation of course vector fields that direct the MAV onto the desired path.

The vector field method produces a field of desired course commands that drive the MAV toward the current path segment. At any point in space, the desired course can be easily calculated. This desired course is used to command heading and roll control loops to guide the MAV onto the desired path. The vector field method uses only the current path segment to find the desired course, avoiding possible singularities and sinks resulting from sums of vectors. Many paths planned for MAVs can be approximated by combinations of straight-line segments and circular arcs [16]. Figure 7.14 shows examples of vector fields for linear and circular paths.

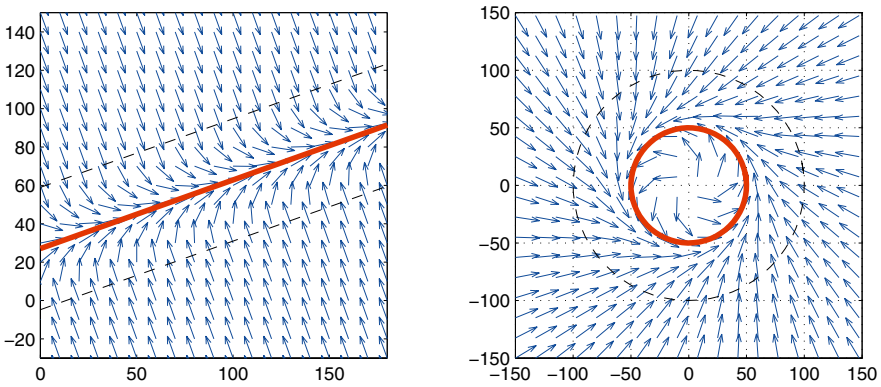

Fig. 7.14. Path following in wind is accomplished by creating a vector field of desired course commands based on the lateral deviation from the path. The Figure on the left shows a possible vector field for a straight-line waypoint path segment. The Figure on the right shows a possible vector field for orbit following.

To account for wind, the course and groundspeed instead of heading and airspeed are used to control the MAV. Ground track motion is the vector sum of the MAV motion relative to the surrounding air mass and the motion of the air mass relative to the ground. Since course direction includes the effects of wind, control based on course is much more effective at rejecting wind disturbances. In implementing the vector field approach, course measurements from GPS are compared with the desired course from the vector field to determine the appropriate control inputs to keep the MAV on the path.

For a given path, the vector field is divided into a transition region and an outer region. This is similar in some respects to the belt zone technique developed in [17]. Outside the transition region, the vector field drives the MAV toward the transition region along a constant course. Once inside, the vector field changes linearly from the entry course direction to the desired course along the path. The effect is to smoothly drive the MAV to follow the path, with larger effort as the error from the path increases. In [14] it is shown that for any initial condition, the MAV will enter the transition region in finite time, and then converge to the desired course asymptotically.

Flight tests have demonstrated the effectiveness of the vector field path following method, even in windy conditions. Figure 7.15 demonstrates path following for straight line segments with acute angles. Wind speeds were approximately 20% of the airspeed during these tests. The vector field method has been shown to be effective in tracking paths of lines and orbits with wind speeds of up to 50% of the airspeed of the MAV. Figure 7.16 shows flight test data for circular orbits and synthetic urban terrain.

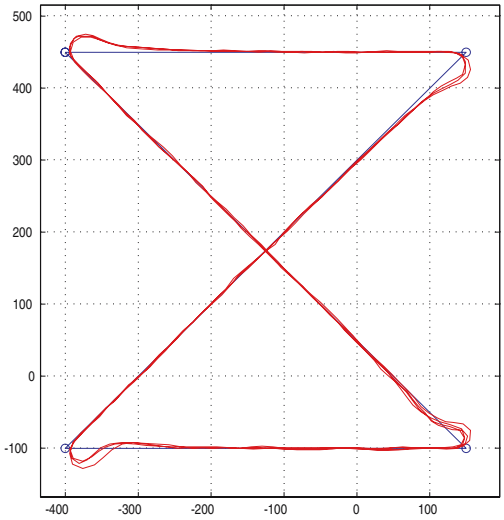

Fig. 7.15. This Figure shows telemetry data for four consecutive traversals of a waypoint path. Wind speeds during the flight were 20%of the MAV airspeed. Note the repeatability of the trajectories even in significant wind.

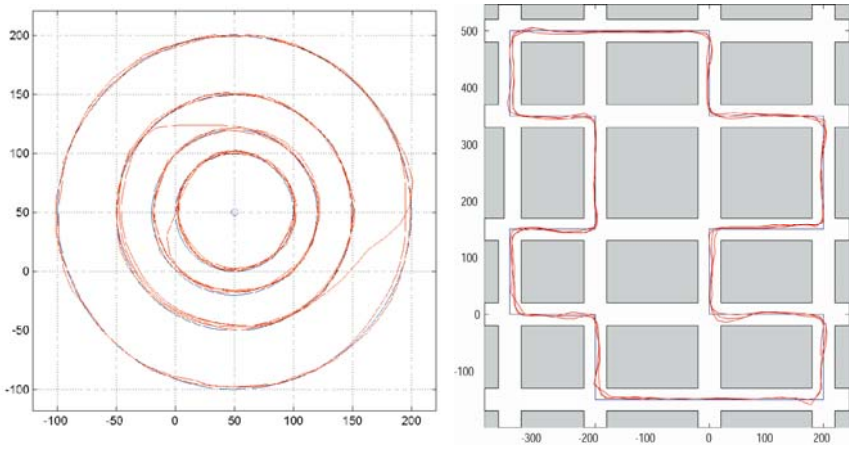

Fig. 7.16. The Figure on the right shows telemetry data for multiple traversals of orbits with radius ranging from 100-300 meters. Wind speeds during the flight were 20%of the MAV airspeed. The Figure on the right shows telemetry data for multiple flights through synthetic urban terrain. While the terrain is synthetic, the data represents actual flight tests in winds that were approximately 30%of the MAV airspeed.

7.3.3 Reactive Obstacle and Terrain Avoidance

Despite having an effective a priori path planner, one cannot guarantee that the flight path will be free of obstacles. This path planner assumes a perfect model of the terrain, but this assumption is not realistic. If an urban terrain model is missing a newly constructed building or a large antenna or tree, a path leading to a collision could result. In fact, the used canyon models are based on 10 m USGS data, which cannot represent small obstacles like trees and power lines. In addition, the GPS sensor used on the MAV has a constant bias that can be as large as 10 m. Path planners can produce a nominal path prior to flight, but the MAV must also have the ability to sense and reactively avoid unanticipated obstacles and terrain in real time.

The following Sections present reactive planners for producing deviations from a nominal path to enable obstacle and terrain avoidance. Section 7.4 presents a method for sensing and avoiding obstacles directly in the flight path and shows results for reactive avoidance of a building. Section 7.5 presents an approach for staying centered between obstacles as might be required for flying through a corridor. Flight test results are presented that demonstrate autonomous navigation of a winding canyon.

7.4 Reactive Obstacle Avoidance

Reactive obstacle avoidance from a MAV platform is challenging because of the size and weight limitations for sensing and computation hardware imposed by the platform. The speed with which avoidance decisions must be made and carried out also causes difficulties. For obstacle avoidance in urban environments, a heuristic algorithm is developed that utilizes a laser ranger to detect and avoid obstacles. The laser ranger points directly out the front of the MAV, and returns range data for objects directly in front of the MAV with a 3 Hz update. For preliminary flight tests, a simple scenario was considered: a single unknown obstacle placed directly in the flight path.

7.4.1 Algorithm

Consider the scenario shown in Figure 7.17 where obstacle avoidance is required. The MAV has a forward ground speed of V and a minimum turn radius R and it is assumed to be tracking the given waypoint path at the time the obstacle is detected by the laser, which has a look ahead dis-

tance L. Figure 7.17 (a) shows the instant when the obstacle is detected by the laser ranger. The basic idea is to construct an internal map of obstacles detected by the laser and to modify the waypoint path to maneuver around the obstacles in the internal map. Refer to the internal representation of obstacles as 'map obstacles'. When the laser detects the location of an obstacle, there is uncertainty about the size and height of the obstacle. In this Chapter map obstacles are represented as cylinders with radius R equal to the minimum turn radius of the MAV, and height equal to the current altitude of the MAV. As shown in Figure 7.17 (b), there are two alternate waypoint paths that maneuver around the map obstacle. The endpoints of the waypoint paths are selected so that the new waypoint paths are tangent to the obstacles in the internal map. As shown in Figure 7.18, the new waypoints are located at a distance $Rd/\sqrt{d^2 - R^2}$ from the original waypoint path, where d is the turn away distance from the obstacle. If both waypoint paths are collision free, then the algorithm randomly selects between the two paths as shown in Figure 7.17 (c). Since the map obstacle may be smaller than the actual obstacle, the laser may again detect the obstacle as it maneuvers on the modified path. If that is the case, a new map obstacle is added to the internal map as shown in Figure 7.17 (d). This process is repeated until the MAV maneuvers around the obstacle as shown in Figure 7.17 (e) and 7.17 (f).

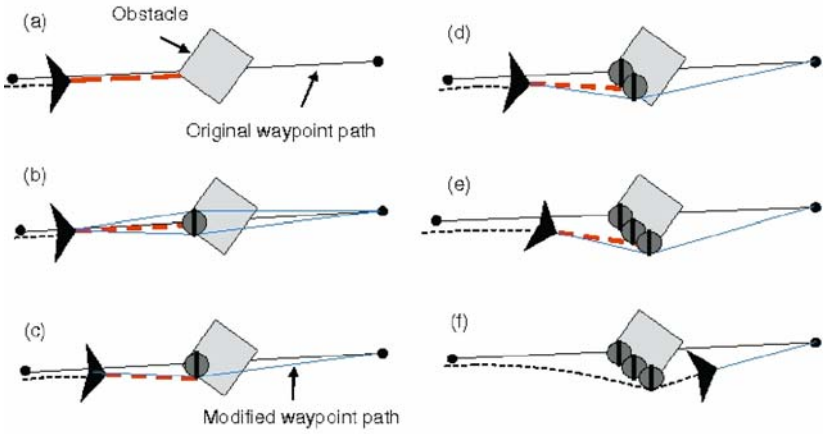

Fig. 7.17. Obstacle avoidance algorithm. (a) The laser detects the obstacle. (b) A map obstacle of radius R is inserted into the map, and two candidate waypoint paths are constructed. (c) A modified waypoint path is randomly selected. (d) The obstacle is again detected by the laser and another map obstacle is constructed. (e-f) The process repeats until the MAV is able to maneuver around the obstacle.

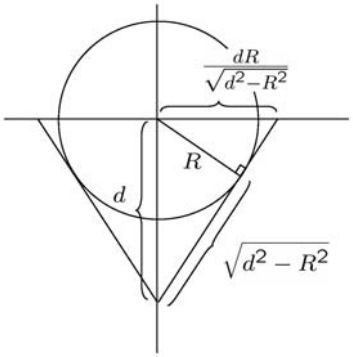

Fig. 7.18. The waypoint path is constructed so that it is perpendicular to the map obstacle. The radius R ensures collision free passage around the map obstacle.

Assuming zero wind, then the 2-D navigation for the MAV is given by:

$$\begin{aligned} \dot{n} &= V\cos\chi \\ \dot{e} &= V\sin\chi \\ \dot{\chi} &= \frac{g}{V}\tan\phi, \end{aligned} \tag{7.5}$$

where g is the gravitational constant, and ϕ is the roll angle of the MAV.

On most MAVs, the roll angle is limited between $-\overline{\phi} \le \phi \le \overline{\phi}$. Considering that the roll dynamics of the MAV are sufficiently fast to assume near instantaneous transitions between $\pm\overline{\phi}$. Therefore, the minimum turn radius is given by $R = \frac{V^2}{g\tan\overline{\phi}}$.

It is essential to establish a minimum turn away distance D so that there is a guarantee to avoid collision with a single rectangular obstacle. The first step is to determine the bounds on the forward and lateral motion of the MAV when it transitions from one waypoint path to the next.

Proposition 7.1: After the insertion of a map obstacle, the MAV requires at most a forward distance of $\frac{2}{\sqrt{3}}R$ and a lateral distance of $\sqrt{\frac{2}{3}}R$ to transition onto the new waypoint path while avoiding the map obstacle.

Proof: Assuming the ability to roll instantaneously between $\pm\overline{\phi}$, the motion of the MAV during the transition can be constrained to lie on circles of radius R. As shown in [13], the path length of the transition increases monotonically with the angle between the old and new waypoint paths. Therefore, the forward and lateral distances are maximized when the

angular separation is maximized, which occurs when instantaneous motion of the MAV follows a circle of radius R that just touches the map obstacle, as shown in Figure 7.19. The proposition follows directly from standard geometrical arguments. Note that the maximum angular separation is therefore given by $\theta = \tan^{-1}\frac{1}{\sqrt{2}} \approx 36^{\circ}$.

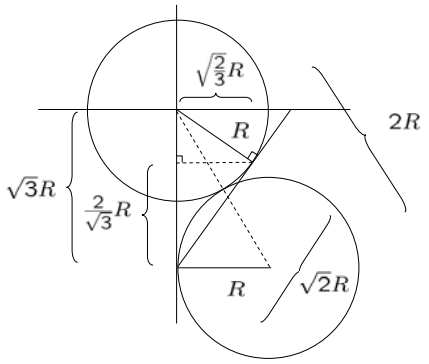

Fig. 7.19. The maximum heading change in waypoint paths is when the MAV must make a full bank to maneuver around the obstacle.

Proposition 7.2: Avoidance of a collision with a flat wall is guaranteed if the turn away distance D satisfies:

$$D > \left(\frac{8 + 2\sqrt{6}}{2\sqrt{3}} \right) R. \tag{7.6}$$

Proof: Consider the worst-case scenario, shown in Figure 7.20, of a MAV that is initially traveling perpendicular to a flat wall. The MAV detects an obstacle and inserts a waypoint at maximum angle $\tan^{-1}\frac{1}{\sqrt{2}}$. After aligning its heading with the waypoint path, the wall is again detected, a map obstacle is inserted, and a new waypoint with maximum angle $\tan^{-1}\frac{1}{\sqrt{2}}$ is planned. This scenario will repeat itself at most three times since $3\tan^{-1}\frac{1}{\sqrt{2}} > \frac{\pi}{2}$. Therefore, the maximum forward direction is bounded by:

$$\sqrt{2}R\left(\left(\sqrt{\tfrac{2}{3}}\right)^1 + \left(\sqrt{\tfrac{2}{3}}\right)^2 + \left(\sqrt{\tfrac{2}{3}}\right)^3 \right) = \left(\tfrac{8+2\sqrt{6}}{2\sqrt{3}}\right) R. \tag{7.7}$$

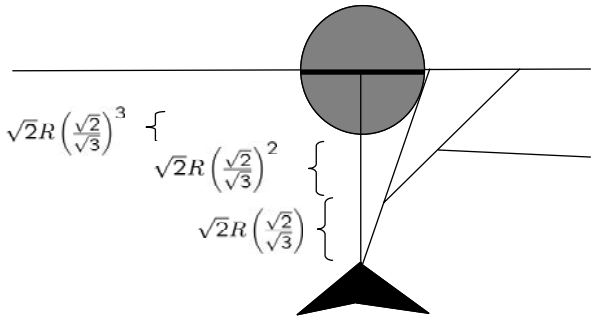

Fig. 7.20. An approximation of the minimum distance required to avoid a straight wall if the laser is only sampled when the MAV is on the waypoint path.

The algorithm described above requires that the laser detect points on the obstacle that are outside of the map obstacles as soon as they become visible. Is this feasible given the update rate of the laser? Let T_s be the time between laser updates.

Proposition 7.3: The maximum distance between laser updates at a range of $d \le L$ is given by:

$$f(d) = 2\sqrt{R^2 + d^2}\,\sin\left(\frac{VT_s}{2R}\right) \tag{7.8}$$

Proof: Assuming the vehicle is turning at its maximum rate, the change in heading between updates is $\frac{VT_s}{R}$. Utilizing the geometry depicted in Figure 7.21, the calculation of $f(d)$ is straightforward. To ensure overlap of map obstacles between samples we require that $f(D) < R$ which implies that:

$$T_s < \frac{2R}{V}\sin^{-1}\left(\frac{R}{2\sqrt{R^2 + D^2}}\right). \tag{7.9}$$

For the BYU airframes, typical values are $V = 13$ m/s, $R = 25$ m, which implies that $D = 93$ m and $T_s < 0.5$ s. The laser ranger sample period of 0.33 s satisfies this constraint, thus ensuring that map obstacles overlap between samples.

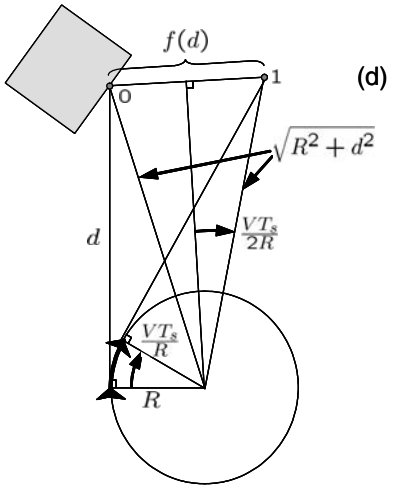

Fig. 7.21. The geometry used to calculate the distance between two consecutive laser updates.

7.4.2 Results

For initial testing of the reactive avoidance algorithm, the choice was to deal with a single obstacle only. It was important that the obstacle be tall enough to allow the MAV to fly at a safe altitude. Flying at an altitude of 40 m also prevented the laser ranger from detecting points on the ground that might be mistakenly interpreted as obstacles, and allowed for losses of altitude that can occur during aggressive maneuvers.

For flight tests, the tallest building on the BYU campus (the Kimball Tower) was used, which is 50 m high and 35 m square. An in-flight image of the building is shown in Figure 7.22. The surrounding buildings are only about 20 m in height. The MAV was directed to fly at 40 m altitude from the south side of the building to the north along a waypoint path that passed directly through the building. Information about the location or the dimensions of the building was not provided to the MAV. A GPS telemetry plot of the results is shown in Figure 7.23.

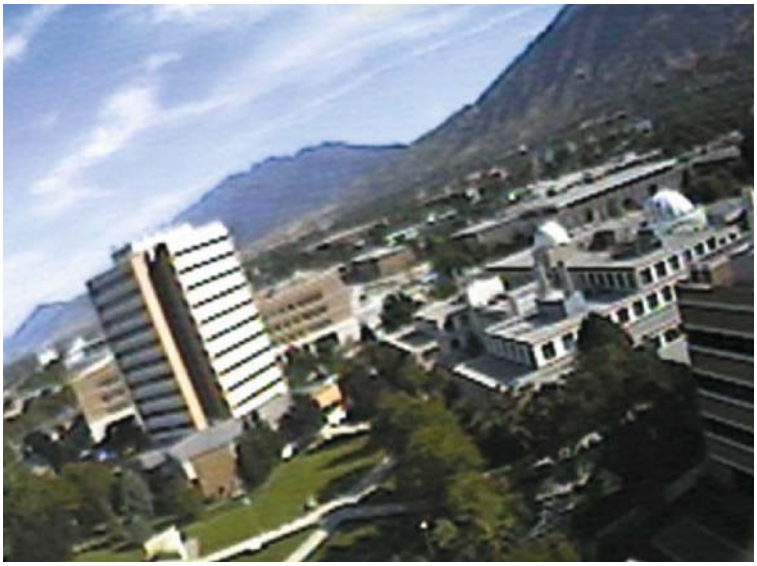

Fig. 7.22. In-flight image of the Kimball Tower on BYU campus during the collision avoidance maneuver using the laser ranger.

As the MAV approached the building, the laser ranger detected the building and calculated its position. When the MAV came within 93 m of the building, the reactive planner generated a path around the building and the MAV began to track the path. Notice that as the MAV began to pass the building, it turned towards the original waypoint path and detected the building a second time. This caused the MAV to execute a second avoidance maneuver before rejoining the original waypoint path. The MAV successfully avoided the building without human intervention. Figure 7.22 shows images of the MAV and its camera view as it executed the avoidance maneuver.

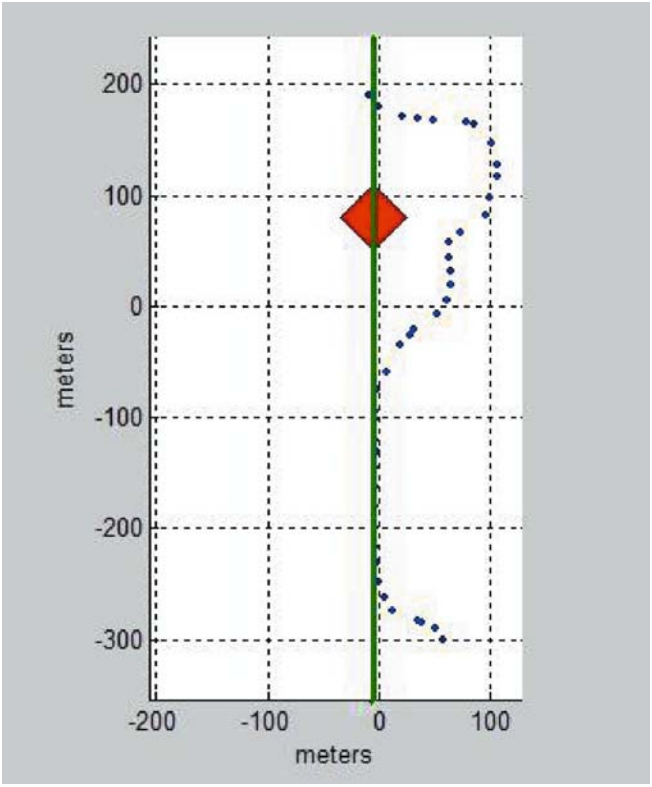

Fig. 7.23. Flight results for collision avoidance using a laser ranger. The green line indicates the planned waypoint path, and the dotted line indicates the GPS track of the MAV.

7.5 Remote Environment Terrain Avoidance

As small MAVs become more reliable and maneuverable, their missions will involve navigating through complex terrain, such as mountainous canyons and urban environments. In this Section, the focus is on terrain avoidance for flying in corridors and canyons. The developed algorithms enable the MAV to center itself within a corridor or canyon, or to fly near walls with a specified offset. The algorithms utilize optic flow sensors like those shown in Figure 7.7. To validate the algorithms, canyon navigation flight experiments were carried out in a mountain canyon.

7.5.1 Canyon Navigation Algorithm

The first step in navigating through a canyon or urban corridor is to select a suitable path through the terrain. This can be done using the RRT algorithm discussed earlier or the operator can utilize maps to define waypoints for the MAV to follow. Preplanned paths will rarely be perfect and some paths could lead the MAV near or even into uncharted obstacles. Reasons for this include inaccurate or biased terrain data, GPS error, and the existence of obstacles that have been added since the terrain was mapped. Therefore, it is important that the MAV be able to make adjustments to its path to center itself between walls and other potential hazards.

In this approach, the MAV follows its preplanned path using the vector field following method. At each time step along the path the MAV computes its lateral distance from objects to the left and right using the optic flow ranging sensors. Using this information, the MAV computes an offset δ from its planned path:

$$\delta = \frac{1}{2}(D_{right} - D_{left}), \tag{7.10}$$

where D_{left} and D_{right} are distances to the walls on the left and right measured by the optic flow sensors. Shifting the desired path by this offset centers the desired path between the detected walls as shown in Figure 7.24. As Figure 7.25 illustrates, shifting the desired path also shifts the vector field accordingly. To improve the performance of this method the optic ranging sensors are pointed forward at a 30 degree angle. This reduces lag caused by filtering the sensor readings and allows the MAV to detect obstacles ahead of its current position.

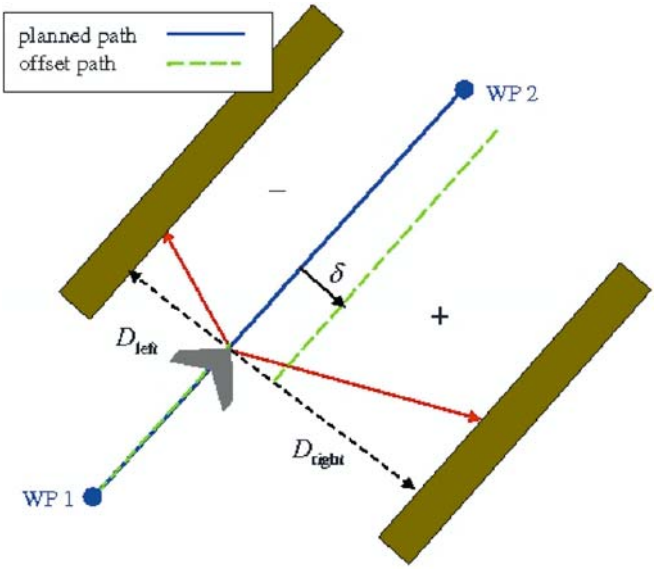

Fig. 7.24. Using the measurements from the optic flow sensors, the planned path (solid blue) is shifted by δ to create a new desired path (dashed green) that is centered between the canyon walls.

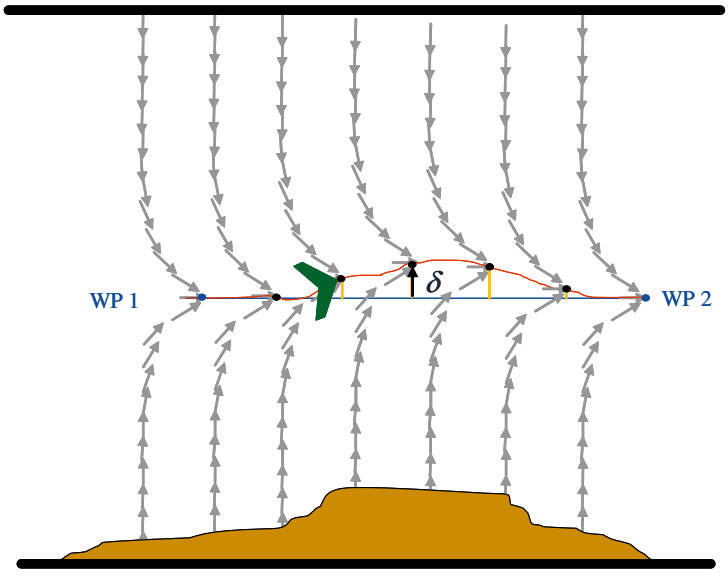

Fig. 7.25. The adjusted path (red) is offset from the preplanned path (blue) by the calculated offset (δ) at each time step to center the desired path between the canyon walls, thus shifting the vector field along with it.

7.5.2 Flight Test Results

Goshen Canyon in central Utah was chosen as a flight test site. This canyon was selected for its steep winding canyon walls that reach over 75 m in height, as well as its proximity to BYU and low utilization by the general population. Flight tests through Goshen Canyon were conducted using the fixed-wing MAV discussed in Section 7.2. Photographs of the flight tests taken by observers and the onboard camera are shown in Figure 7.26. In the first flight through the canyon, the planned path was selected to follow the road. The MAV navigated the canyon with only minor adjustments to its path. For the second flight, the planned path was intentionally biased into the east canyon wall to verify that the navigation algorithms would correct the planned path toward the center of the canyon, enabling the MAV to avoid the canyon walls.

Fig. 7.26. This figure shows the MAV as it enters Goshen Canyon. The inset is an image from the camera on-board the MAV.

Fig. 7.27 Results from the second flight through Goshen Canyon. Flight test results show the planned path (green) and the actual path (blue). The planned path was intentionally biased to the east forcing the MAV to offset from its planned path to center itself through the canyon.

Telemetry data from the second flight has demonstrated that the MAV biased its desired path up to 10 m to the right to avoid the canyon walls. If the MAV had not biased its path it would have crashed into the east canyon wall.

7.6 Summary

Miniature aerial vehicles have demonstrated their potential in numerous applications. Even so, they are currently limited to operations in open air space, far away from obstacles and terrain. To broaden the range of applications for MAVs, methods to enable operation in environments of increased complexity must be developed. In this Chapter, two strategies have been presented for obstacle and terrain avoidance that provide a means for avoiding obstacles in the flight path and for staying centered in a winding corridor. Flight tests have validated the feasibility of these approaches and demonstrated promise for further refinement.

References

1. Chahl J., Srinivasan M., and Zhang S., "Landing Strategies in Honeybees and Applications to Uninhabited Airborne Vehicles", *International Journal of Robotics Research*, Vol. 23, No. 2, 101–110, 2004.
2. Barrows G., and Neely C., "Mixed-Mode VLSI Optic Flow Sensors for In-flight Control of a Micro Air Vehicle", *Proceedings, SPIE*, 52-63, San Diego, August 2000.
3. Ruffier F., and Franceschini N., "Visually Guided Micro-Aerial Vehicle: Automatic Take off, Terrain Following, Landing and Wind Reaction", *Proceedings, IEEE International Conference on Robotics and Automation*, 2339-2346, New Orleans, April-May 2004.
4. Zufferey J.-C., and Floreano D., "Toward 30-gram Autonomous Indoor Aircraft: Vision-Based Obstacle Avoidance and Altitude Control", *Proceedings, IEEE International Conference on Robotics & Automation*, Barcelona, April 2005.
5. Barrows G. L., Chahl J. S., and Srinivasan M. V., "Biomimetic Visual Sensing and Flight Control", *The Aeronautical Journal, London, The Royal Aeronautical Society*, Vol. 107, 159–168, 2003.
6. Procerus Technologies, http://procerusuav.com/.
7. LaValle S. M., and Kuffner J. J., "Randomized Kinodynamic Planning", *International Journal of Robotics Research*, Vol. 20, No. 5, 378–400, May 2001.
8. Kavraki L. E., Švestka P., Latombe J.-C., and Overmars M., "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces", *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, 566–580, 1996.
9. Amato N. M., and Wu Y., "A Randomized Roadmap Method for Path and Manipulation Planning", *Proceedings, IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 1996.
10. LaValle S. M., *Rapidly-Exploring Random Trees: A New Tool for Path Planning*, TR 98-11, Computer Science Department, Iowa State University, October 1998.
11. Frazzoli E., Dahleh M. A., and Feron E., "Real-Time Motion Planning for Agile Autonomous Vehicles", *AIAA Journal of Guidance, Control and Dynamics*, Vol. 25, No. 1, 116–129, 2002.
12. Saunders J. B., Call B., Curtis A., Beard R. W., and McLain T. W., "Static and Dynamic Obstacle Avoidance in Miniature Air vehicles", *Proceedings, Infotech@Aerospace Conference*, September 2005.
13. Anderson E. P., *Constrained Extremal Trajectories and Unmanned Air Vehicle Trajectory Generation*, Master's Thesis, Brigham Young University, Provo, Utah, April 2002.
14. Nelson D. R., *Cooperative Control of Miniature Air Vehicles*, Master's Thesis, Brigham Young University, Provo, Utah, December 2005.

15. Skjetne R., Fossen T., and Kokotović P., "Robust Output Maneuvering for a Class of Nonlinear Systems," *Automatica*, Vol. 40, 373–383, 2004.
16. Anderson E. P., Beard R. W., and McLain T. W., "Real Time Dynamic Trajectory Smoothing for Uninhabited Aerial Vehicles", *IEEE Transactions on Control Systems Technology*, Vol. 13, No. 3, 471-477, May 2005.
17. Loizou S., Tanner H., Kumar V., and Kyriakopoulos K., "Closed Loop Motion Planning and Control for Mobile Robots in Uncertain Environments", *Proceedings, 42nd IEEE Conference on Decision and Control*, 2926–2931, 2003.
18. Quigley M. , Goodrich M. A. , and Beard R. W. , "Semi-Autonomous Human-UAV Interfaces for Fixed-Wing Mini-UAVs", *Proceedings, IEEE/RSJ Conference on Intelligent Robots and Systems,* Sendai, Japan, September 2004.
19. Quigley M., Goodrich M. A., Griffiths S., Eldredge A., and Beard R. W., "Target Acquisition, Localization, and Surveillance Using a Fixed-Wing Mini-UAV and Gimbaled Camera", *Proceedings, IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005.
20. Imperx Incorporated, http://www.imperx.co.