# 1

# Introduction



Autonomy is replacing the human operator in many applications. Examples involve military systems where there is some element of danger to the human operator, civilian systems when handling hazardous materials, as well as monotonous operations such as surveillance, reconnaissance and *dull, dirty and dangerous* missions, such as operations in chemical and biological environments (Blyenburgh 1999; NMAB and ASEB 2000) and in environmental monitoring (Roberts *et al*. 2008). The replacement of the human operator in such systems necessitates the development of autonomous systems techniques. Such autonomous systems operate in many environments, such as in the air, in and under water, in space or on the land.

In this book, unmanned aerial vehicles (UAVs) are studied, operating as a group. The large range of potential applications of UAVs in military and civilian sectors have generated a lot of academic as well as commercial research (OSD 2005; Wilson 2007). Inspired by examples in nature, such as flocks of birds, shoals of fish, swarms of bees, and colonies of ants, cooperative control (Rabbath *et al*. 2004; Uny Cao *et al*. 1997) has become one of the active research areas in autonomous systems. Employing a group of UAVs rather than a single UAV can result in cost-effective and fault-tolerant systems. Advances in avionics, navigation based on GPS (Global Positioning System), flight control techniques and low-cost electronics have further fuelled the use of UAVs in commercial and military applications. Future UAVs will be more autonomous than the remotely piloted reconnaissance platforms in use today.

*Cooperative Path Planning of Unmanned Aerial Vehicles*
Antonios Tsourdos, Brian White and Madhavan Shanmugavel

One of the open issues in the development of autonomous systems is that of path planning. A path planning algorithm produces one or more safe flyable paths for the UAVs. The path has to be of a specified (usually minimal) length, and, as the UAV has limited range, the time spent surveying specific areas should be minimised. In addition, when surveying an area or a location, it is beneficial to be able to approach from specific directions in order to minimise obscuration as well as to aid identification. Hence the path length and direction will always be major factors in any path planning algorithm.

The UAVs should be capable of following any resulting path. This implies that the trajectory must comply with the speed and manoeuvre constraints of the UAVs. The path planning algorithms must also allow for the deployment of several UAVs in a coordinated manner, which will involve collision avoidance and simultaneous arrival at one or more locations. Finally, path planning algorithms are required to be coded in software that runs on a processor carried on-board the UAVs. Thus, they must be computationally efficient, take up a small amount of memory and operate in real time, enabling the UAV to re-plan its trajectory if needed, with no significant delay.

## 1.1 Path Planning Formulation

The primary aim of path planning is to provide structured mobility, that is, to facilitate moving or flying multiple UAVs from one location to another. There may be several locations to visit before reaching the final destination, and hence several consecutive paths may be required. Generally, there will be several predefined points of interest (POIs) on a known or partially known map/area. The UAV will have a specific attitude, which is combined with its location to give the UAV pose $P(x, y, z, \theta, \psi)$, where $(x, y, z)$ is the UAV location or waypoint and $(\theta, \psi)$ are the horizontal and vertical angles, respectively. Consider a UAV moving from one pose, $P_s$, to another, $P_f$, where $P_s$ and $P_f$ are labelled the start and finish poses, respectively. Path planning involves producing one or more flight paths $r(q)$ connecting $P_s$ and $P_f$. Mathematically, this can be represented as

$$P_s \xrightarrow{r(q)} P_f, \tag{1.1}$$

where $r(q)$ is the resulting path, and $q$ is defined as a path parameter. This parameter can be a length variable $(0 \le q \le s)$ for a straight-line path or an angle variable $(0 \le q \le \theta)$ for a curved path. The choice of path variable depends on the path formulation.

Equation (1.1) is in a very simple form. For a single UAV flying from a location with start pose $P_s(x_s, y_s, z_s, \theta_s, \psi_s)$ to a location with finish pose $P_f(x_f, y_f, z_f, \theta_f, \psi_f)$, equation (1.1) can be written in the form

$$P_s(x_s, y_s, z_s, \theta_s, \psi_s) \xrightarrow{r(q)} P_f(x_f, y_f, z_f, \theta_f, \psi_f). \tag{1.2}$$

Extending equation (1.2) for $N$ UAVs, where each pair of poses are connected by paths $r_i(q)$, gives

$$P_{si}(x_{si}, y_{si}, z_{si}, \theta_{si}, \psi_{si}) \xrightarrow{r_i(q)} P_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi}, \psi_{fi}), \qquad i = 1, \ldots, N. \tag{1.3}$$

Equation (1.3) connects a pair of points by a path. This problem is well known as route planning in the fields of operations research, communications, computational geometry and graphics, where a route is generated between one or more nodes of a network. However, applying the route planning concept from these fields to flying vehicles becomes challenging. The route is usually defined by a set of waypoints joined by straight-line segments, which connect the start and finish waypoints, and hence may not be flyable because the UAV cannot turn instantaneously through each waypoint. For a flyable path, each segment must have a common tangent to produce a continuous path. Hence it is important to specify the orientations at each waypoint that each segment must match. This implies that some segments must be curved rather than straight in order that each end of the segment meets the common tangent condition. Also, some missions – such as target acquisition, search and track, and disaster area surveying – require the sensors to be pointed in specific directions for effective detection and identification.

## 1.2 Path Planning Constraints

Producing a path between the start and finish poses is straightforward in the absence of any constraints. In practice, there are various constraints involved in path planning, most being UAV-specific and the remainder arising from obstacles in the environment. Sometimes the violation of constraints, for example, communication failure, may lead to the complete loss of the UAV in extreme circumstances and to the loss of the UAV as a sensor platform in less extreme cases. Hence, safety of the UAVs is important throughout the mission. Other constraints such as the minimum turn radius will also dictate which paths are flyable and which are not.

Hence, the two most important constraints for path planning of a UAV are that the path must be flyable and safe. Flyable paths meet kinematic or motion constraints and dictate the manoeuvrability of the UAVs. The safety of the UAV is achieved by avoiding obstacles, either fixed or moving, that intersect the path. The obstacles may be UAVs, aircraft and buildings, which are common in airspace and in urban environments. Other constraints, such as maintaining communication range, as well as minimising time and/or path length, can be added into the system where necessary. We use the symbol $\coprod$ to represent the constraints. Hence the constrained path planning can be written in the form

$$P_{si}(x_{si}, y_{si}, z_{si}, \theta_{si}, \psi_{si}) \xrightarrow{\coprod r_i(q)} P_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi}, \psi_{fi}). \tag{1.4}$$

From equation (1.4), we can make an analogy that the path planner acts like a black box, which produces the flyable path from given inputs as shown in Figure 1.1. The inputs are the poses, with additional constraints, uncertainties and measurements. The feedback loop senses the measured states of the UAV and also feeds back the success in terms of meeting the constraints to the path planner.

### 1.2.1 Flyable Paths: Capturing Kinematics

As the properties of the path influence the motion of the UAV and vice versa, it is important to understand the characteristics of the path. The UAV needs a path in order to move from one location to another. However, the path has to meet the dynamic constraints of the UAV. In order to understand

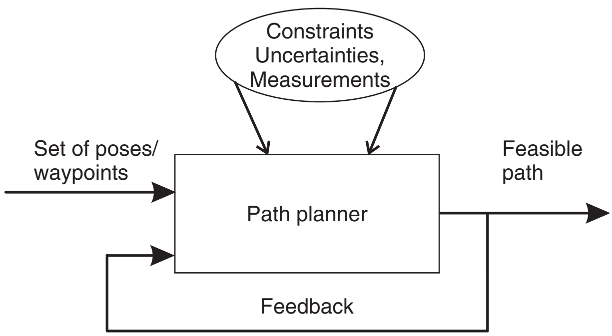


**Figure 1.1** A block diagram approach to path planning

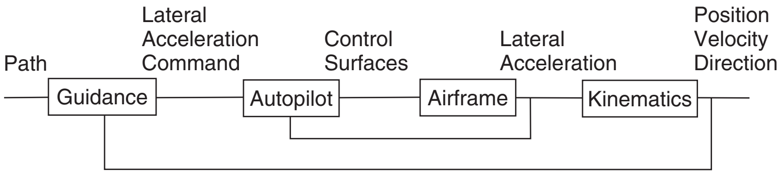


**Figure 1.2** Autopilot and guidance control loops

the dynamic constraints of the UAV and hence the kinematic constraints of the path, consider the UAV as a control system. Two control loops are present in the UAV system, as shown in Figure 1.2. The inner loop is known as the autopilot and the outer loop is known as the guidance system. The guidance system provides lateral acceleration commands to keep the UAV following the path, whereas the autopilot controls the UAV elevator, ailerons and rudder to achieve the required lateral acceleration.

The UAV dynamics include the aerodynamics, which produces forces and torques on the airframe. From Newton's third law of motion, the forces and torques produce lateral, longitudinal and rotational accelerations. The accelerations are usually expressed in UAV body axes and these provide the link to the kinematics. The kinematics is produced by integrating the UAV lateral and rotational acceleration vectors to obtain the UAV velocity vector. The attitude angles and the current UAV position in the inertial frame give the UAV's translational and rotational velocity. For example, the two-dimensional (2D) path planner uses the kinematic model in equation (1.5). The kinematics and hence the current state of the UAV in 2D are thus

$$\dot{x} = |\boldsymbol{v}|\cos(\theta), \tag{1.5a}$$

$$\dot{y} = |\boldsymbol{v}|\sin(\theta), \tag{1.5b}$$

where $\boldsymbol{v}$ is the UAV velocity and $\theta$ is the horizontal heading angle.

Whether a given path is flyable or not is determined by the curvature of the path. The path planner has to produce a path $r(q)$ that meets the dynamic turn rate constraint of the UAV, which is translated into the kinematic curvature constraint – in 3D, it is determined by both the curvature and torsion (Lipschutz 1969). By satisfying this constraint, the motion of the UAV stays within its maximum-curvature (acceleration) bounds. The curvature is proportional to the lateral acceleration. Thus the curvature at any point on the path must be less than the maximum-curvature constraint of the UAV.

A curve segment of zero curvature is a straight line and a curve segment of constant curvature is an arc of a circle. At a given speed $\boldsymbol{v}$, the lateral acceleration $\boldsymbol{a}$ is proportional to the curvature $\kappa$ such that

$$|\boldsymbol{a}| = |\boldsymbol{v}|^2\kappa \propto \kappa, \tag{1.6}$$

where $\boldsymbol{a}$ is the lateral acceleration vector, $\kappa$ is the curvature and $\boldsymbol{v}$ is the velocity vector. Note that, for a constant speed, the acceleration vector $\boldsymbol{a}$ is normal to the velocity vector $\boldsymbol{v}$. This ensures that the velocity vector rotates without changing its magnitude. This is illustrated in Figure 1.3, where the velocity vector is aligned with the tangent vector $\boldsymbol{t}$ and the acceleration vector is aligned with the normal vector $\boldsymbol{n}$.

### 1.2.2 UAV Inertial Manoeuvre Coordinates

All manoeuvres can be defined by reference to a set of axes. These can be fixed in inertial space or defined relative to a set of axes that are attached to the path defined by a curve known as the Frenet–Serret (FS) framework, which will be dealt with later in the book. A set of axes may be defined that comprises a tangent vector $\boldsymbol{t}$ to the curve, together with a normal to this tangent $\boldsymbol{n}$. Together with a binormal vector $\boldsymbol{b}$, these make up a right-handed triple, which can be used to define the manoeuvres of the UAV. Two-dimensional path planning confines the manoeuvre of the UAV to a plane. If the plane is horizontal, this is equivalent to flying at a constant altitude. The plane need not be horizontal, but can be inclined at an angle to facilitate a change in altitude if required. However, in reality, UAVs fly in three-dimensional (3D) space, and so path planning needs to be able to produce flyable paths in three dimensions. In 3D, flyable paths need to accommodate both curvature and torsion in their design. Curvature defines the turn radius of the path in 2D, which is defined as a rotation about an axis normal to the manoeuvre.

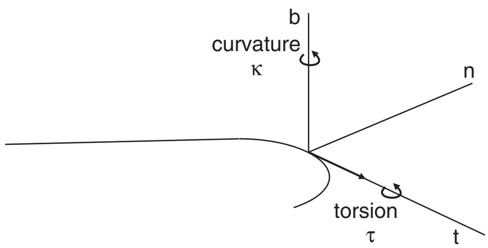


**Figure 1.3** Curvature and torsion

Torsion is defined as a rotation about an axes that coincides with the path tangent vector. This is shown in Figure 1.3. Note that positive curvature and positive torsion are defined as being clockwise along the appropriate axis when looking along the axis from the origin.

For the UAV, in terms of body axes, curvature is equivalent to a yaw rate turn and torsion is equivalent to a roll rate turn. However, in practice, the UAV will perform a turn by first rolling to a fixed bank angle and then using the elevator to produce a manoeuvre normal to the wings. This is known as a bank-to-turn manoeuvre.

### *1.2.3 Generation of Safe Paths for Path Planning*

The second important constraint of path planning is safety. The planned path should be flyable and also safe. Safety is measured by the ability of the UAV to avoid fixed and moving obstacles and other UAVs. As the UAV follows the path defined by the path planner, the safety constraint has to be defined with respect to the properties of the path. The path must maintain collision avoidance with other friendly UAVs and also must be flexible enough to avoid obstacles. The safety of the UAV is represented by $\coprod_{\text{safe}}$, and can be used in equation (1.4). Hence, taking into account the curvature constraint $\coprod_{\kappa}$ and safety, the equation of path planning is now modified to

$$P_{\text{s}i}(x_{\text{s}i}, y_{\text{s}i}, \theta_{\text{s}i}, \psi_{\text{s}i}) \xrightarrow{\coprod_{\text{safe}},\ \coprod_{\kappa},\ r_i(q)} P_{\text{f}i}(x_{\text{f}i}, y_{\text{f}i}, \theta_{\text{f}i}, \psi_{\text{f}i}). \tag{1.7}$$

There may be other constraints, such as maintaining communication in a complex urban environment, as well as time, task completion and resource management, depending on the mission objectives. The subscript can be changed for different constraints. For example, imposing a communication constraint on UAV separation distance would be represented by $\coprod_{\text{comm}}$.

## 1.3 Cooperative Path Planning and Mission Planning

The previous section defines path planning for individual UAVs. Cooperation involves using the path planning algorithms to produce a coordinated mission. One such mission is for a group of UAVs to set off from a base and reach a rendezvous position at the same time. The UAVs will perform tasks on the way, such as area searches and object detection.

The payload, sensor suite and duration of operation, together with the cost and size of the UAV, are important in their use and deployment. The limited sensor range, period of operation and payload limitations usually result in the need to use more than one UAV. The mission can be accomplished faster if a single UAV is replaced with a group of UAVs. Such operations do not simply involve operating together in an environment. The challenging task is cooperation among the UAVs, and coordination of their activities. Therefore, a group of UAVs of the same (homogeneous) or different (heterogeneous) capabilities can act together as a single entity to achieve the mission. Also, a system of cooperative UAVs is fault-tolerant, and the performance is better than that of a single UAV.

For the case of autonomous UAV systems, cooperation between UAVs means shared utilization of resources, sharing of information, allocation of tasks, coordination of actions and operations, coping with disturbances and dealing with conflict resolution. The level of cooperation can be set by a central decision-maker in a ground control station, for example, or it may be left to individual UAVs in a distributed manner. Also, it is important to note that cooperation is a complex problem. Consider collision avoidance between two cooperating UAVs. Though they mutually manoeuvre to avoid collision, the resulting modified path plans may lead to collision with other UAVs in the group. Therefore, cooperation has to be planned with respect to the other members of the UAV group, as well as obstacles (both fixed and moving) in the environment and uncertainties. It has been argued (Ren *et al*. 2004) that information sharing is central to cooperation and coordination, where information sharing is represented by a coordination variable. However, a minimum amount of information for cooperation has to be defined for the system. This, in turn, affects the cost of communication and computation. Cooperative behaviour is also studied in robotics research (Fong *et al*. 2003; Howard *et al*. 2006; Uny Cao *et al*. 1997). Other related research areas are ‘multi-agent control’, ‘distributed networks’, ‘consensus algorithms’, ‘cooperative control’, ‘network control’ and ‘swarm intelligence’ (Qu 2008; Shamma 2007). All these research areas emphasise that the sharing of information is the important factor in cooperative system. A variety of other applications, such as task allocation (Beard *et al*. 2002), flight formation (Fowler and D’Andrea 2003), surveillance, suppression of enemy air defence (SEAD) and radar jamming, have been studied for the cooperative system in recent research. An overview of recent research on cooperative systems can be seen in Murray (2007).

Cooperation can be achieved by central coordination or distributed decision-making. Can cooperation be achieved simply by solving a set of equations or by taking decisions based on testing conditions and/or constraints? This problem is still an open research area for autonomous systems.

For example, collision avoidance between two flight paths $r_1(q)$ and $r_2(q)$ can be represented by

$$r_1(q) \cap r_2(q) = \varnothing. \tag{1.8}$$

This equation captures the condition required to be satisfied for collision avoidance. A cooperative path planning of multiple UAVs for collision avoidance can be formulated as producing flight paths $\{r_1(q), r_2(q), \ldots, r_n(q)\}$ subject to the following constraints:

$$\coprod = \begin{cases} \coprod_\kappa, & \kappa \leq \kappa_{\max}, \\ \coprod_\tau, & \tau \leq \tau_{\max}, \\ \coprod_{\text{safe}}, & r_i(q) \cap r_j(q) = \varnothing, \qquad i \neq j. \end{cases} \tag{1.9}$$

Such problems can be solved by an optimisation algorithm, but this involves heavy computation. Owing to the complexity of the problem, in many cases the problem is divided into different phases/stages. The optimal or suboptimal solution will be a sequential solutions to the subproblems.

As cooperative behaviour arises when two or more UAVs are involved, a functional architecture is necessary to coordinate the communication and

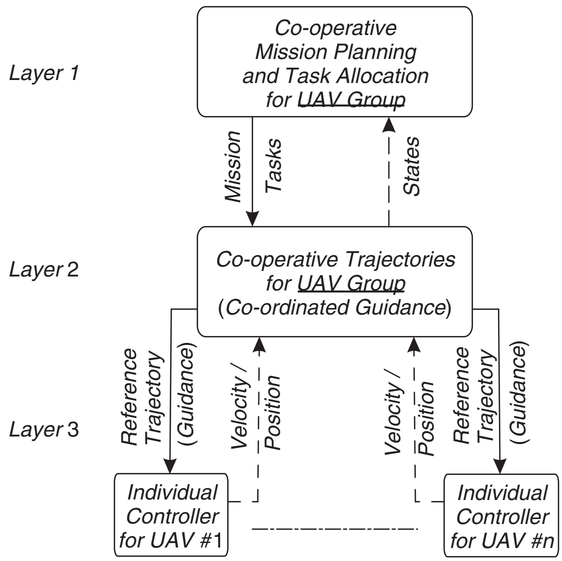


**Figure 1.4** Hierarchy of mission planning. Reprinted with permission of Elsevier, and ASME

control of each UAV. One such architecture is shown in Figure 1.4. The cooperative behaviour is defined by the central mission planner or can be in-built into each UAV, depending on the autonomy architecture. As shown in the figure, the path planning activity is a subsystem of the mission planner. The figure shows three subsystems or layers. However, the number of subsystems and their functions may vary for different applications and mission objectives. The top layer holds and keeps track of the objective(s) of the mission. Based on these objectives, this layer allocates resources and tasks to the UAVs and also acts as decision-maker. The intermediate (second) layer produces trajectories (paths) for the UAVs. In this layer, the path planning and their associated algorithms such as collision avoidance to produce feasible trajectories (paths) are located. The lower (third) level produces guidance and control actions, which ensure that the UAVs fly on the reference trajectories produced by the second level. This book focuses on the second layer, where the path planner produces flight trajectories (paths) to fulfil the mission objectives. The mission objective considered here is the simultaneous arrival at a specified location by a group of UAVs. Note that one of the main requirements for successful cooperation and coordination is that the UAVs have to be in communication most of the time during the mission. This is dealt with using the communication constraint $\coprod_{\text{comm}}$ mentioned in section 1.2.3.

## 1.4 Path Planning – An Overview

Path planning is an integral part of an autonomous system, which is responsible for moving from one point to another. An autonomous system may be operating either on land, in the air, or on/in water. The paths are designed by various techniques based on whether the system has to traverse an area that is known, unknown, or partially known. The physical limitations, operating environment and communication requirements make the path planning more complex. It is essential to have an on-board processor to design and execute paths and trajectories. Recent work in cooperative application of unmanned systems has employed multiple autonomous vehicles to enhance the effectiveness of operation success. Such applications further increase the complexity of the path planning.

Research into path planning is widely documented in ground robotics and manipulator systems. Most of the path planning techniques that are currently in use are borrowed from the ground robotics field. Originally, research on ground robotics focused on indoor robots. The lessons learned from them were applied to outdoor autonomous vehicles, and to aerial robotics, with improvements and modification. In all these applications, path planning plays an integral part and an important role. This is described in various references, for example, Bender (1991) and Shladover (1991) in ground robotics, Chan and Foddy (1985), Hebert *et al*. (2001), Vian and More (1989) and Zabarankin *et al*. (2002) on aerial vehicles, Yuh (1995, 2000), Smith *et al*. (2001) and Oliveira *et al*. (1998) in underwater vehicles, and Tompkins *et al*. (2004) and Gennery (2004) in space. Therefore, it is appropriate to review the techniques originally developed for ground robotics.

The objectives and approaches of path planning differ depending on the application domain: surveillance, search and track, rescue missions and disaster monitoring. Currently, there are a multitude of solution approaches available in the research literature, and each approach has its own merits. Moreover, new approaches are required as the complexity of the problem increases. However, the majority of solution approaches can be represented by a simplified diagram as shown in Figure 1.5. Using this approach, the inputs to the path planner are the waypoints, the obstacle positions and size, and the associated uncertainties. Optimisation techniques are applied to these data to produce routes. The resulting routes are usually polygonal paths, which do not have inherent curvature constraints built into the solution. In most cases the path planner produces routes that are applicable to mobile robots as they can virtually stop and turn. But in the case of UAVs, these

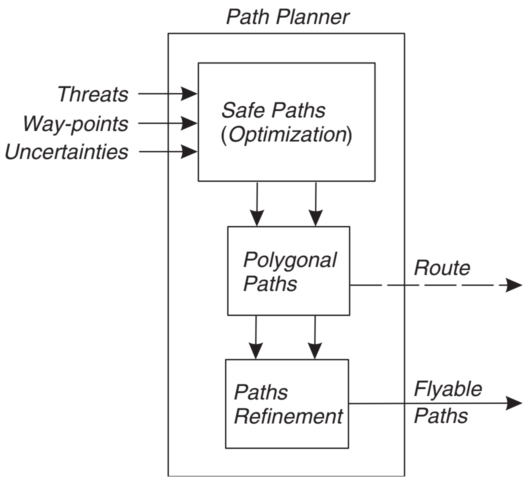


**Figure 1.5** Existing approach to path planning. Reprinted with permission of Elsevier

routes may not be flyable, driveable or manoeuvrable. A flyable path meets the kinematic constraints and the imposed dynamics of the robot. Therefore, attention has turned towards the development of paths that can be flyable or manoeuvrable. Therefore, these routes have to be further refined to produce flyable paths.

In the literature, a wide variety of approaches are used to produce routes using a route planner. The route planner produces a series of nodes that are connected by straight lines, in turn connected to the start and finish points. This is also called a 'global path planner', and produces one or more routes for the given map of known locations. The locations may be places to visit or places to avoid. The route is produced by isolating the obstacles, no-fly zones and threats. The road map, and optimization methods that produce routes using straight-line segments are route planners. Though the routes cannot be followed by the UAVs directly, they are important, as they can be produced faster than producing a flyable path in an environment containing fixed obstacles. Hence, it is common for the path planner first to produce a route and then later to refine the route for a flyable path. However, this is not the only way to produce the path.

Path planning approaches can be divided into several categories, based on different criteria. However, the nature of the applications, environment, medium of operation and path constraints have produced a variety of algorithms and techniques. The predominant methods used in ground robotics and documented in Latombe (1991) are: (i) the road map method, (ii) the cell decomposition method, and (iii) the potential field method. These methods generate routes for a robot to move from a start point to a finish point. Also, these methods rely on specific definition of the environment. The environment is defined by a map, which contains known and unknown obstacles. Finding a path between two points on the map is simplified by (i) discretizing the map into small areas or cells or (ii) converting the map into a continuous field. Hence, path planning can classified as discrete or continuous. A suitable search algorithm is then used to find a path connecting the start and finish points on this simplified map. The road map and cell decomposition methods transform the environment into a discrete map, while the potential field method transforms the map into a continuous function. The textbook by Latombe (1991) is a thorough review of these methods. Here, only a descriptive version is presented for completeness. These methods, which produce a path for a given map or environment, are also called global path planners. These methods transform the given environment into a searchable database. We consider each in turn.

## 1.5 The Road Map Method

A road map itself is not a path planner. It is a two-dimensional network of straight lines connecting the initial and final points without intersecting the obstacles defined in the map. In other words, it is a representation of the 2D environment by a non-directional graph. The road map methods work in configuration space, where the robot is treated as a point while the workspace is modified to accommodate the physical size of the vehicle. For a given start point $P_s$ and finish point $P_f$, all possible connecting paths are generated by avoiding obstacles. A typical metric such as A* search algorithm (LaValle 2006) is used for generating the shortest route between destinations. The resulting route consists of a series of waypoints connecting the start and finish locations.

In general, the algorithms work in the following way (see Figure 1.6). The map is defined by a space named the total space, $C_{space}$. This is then split into the free space (obstacle-free space) $C_{free}$ and the obstacle space $C_{obst}$. Next, a network of graph connectivity $Q_{free}$ is generated by selecting a set of points that can be connected by straight lines such that the resulting discretization produces a set of polygons that surround each obstacle in $C_{space}$. The resulting connectivity graph is used to produce a network of all

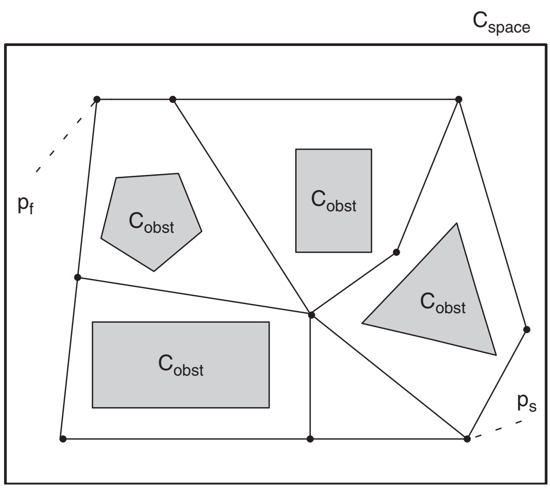


**Figure 1.6** The road map method

possible collision-free routes. Search algorithms such as A* are then used to find one or more routes based on some metric between the start and finish poses or locations. Two such methods used in the literature employ the connectivity graph to produce a route. These are (i) the visibility graph, and (ii) the Voronoi diagram.

### 1.5.1 Visibility Graphs

As the name suggests, the visibility graph produces a line-of-sight route through an environment. This is one of the earliest methods used for path planning. The route is formed by a connectivity graph network of a non-directed graph of straight lines. For this method, only polygonal obstacles are considered. For a graph $G = (V, E)$, the vertices $V$ are the vertices of the obstacles, while the edges $E$ connect all the vertices with straight lines, provided the lines do not intersect the obstacles. Hence only vertices that are visible in the sense that each vertex can be seen from the other are included. A schematic sketch of a visibility graph is shown in Figure 1.7. A route is then found using a graph search algorithm, which connects the start and finish locations, treating them as vertices. Hence only visible vertices are connected to the start and finish vertices.

### 1.5.2 Voronoi Diagrams

A Voronoi diagram is a connectivity graph generated by forming polygons or ‘fences’ around the obstacles. Each edge of the fence polygons is defined by first constructing a set of lines connecting the centres of the obstacles.

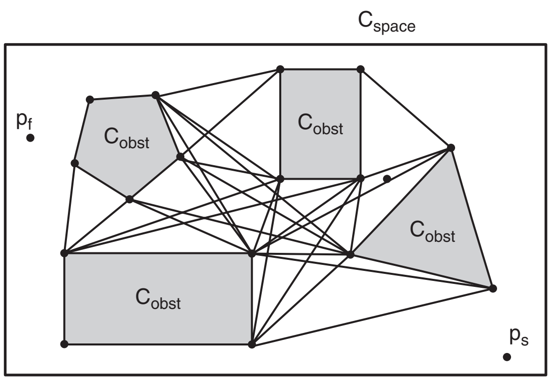


**Figure 1.7** Visibility graph

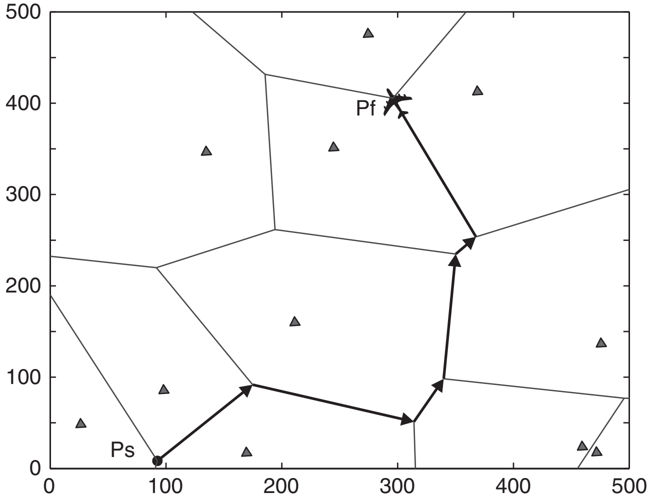


**Figure 1.8** Voronoi diagram: polygonal fences around obstacles

The set of polygonal fences is then constructed by drawing a set of lines perpendicular to the lines connecting the obstacles. These are then adjusted to meet at a minimum set of vertices. An example of a Voronoi diagram is shown in Figure 1.8.

The resulting polygonal fences can be considered as a connected graph. A search algorithm such as A* can be used to find a route connecting an initial and final vertex in the graph. The Voronoi diagram is used in Chandler *et al*. (2000) to produce a route for a UAV flying in an environment of static radar sites whose locations are known *a priori*. The route is refined by adding fillets at the vertices. Simultaneous arrival of multiple UAVs is coordinated by a high-level manager based on the sensitivity function (cost versus time of arrival) broadcast by each UAV. A similar approach is adopted in Bortoff (2000), where an analogy of a chain connected by sequences of spring–mass–damper systems to the UAV path is used. The ends of the chain are located at the initial and final configurations. The obstacles induce a repulsive force, which causes the masses in the chain to move away from the obstacles. However, this method involves complexity in solving ordinary differential equations (ODEs) with curvature constraints. Also, the accumulation of only a few masses around the obstacle location will lead to coarse path resolution, which is undesirable. In McLain and Beard (2000), the above approach is extended by replacing the spring–damper system with rigid links between masses to eliminate sharp corners. However, this method does not guarantee that the resultant path is flyable by a UAV. Later (Judd and McLain 2001), the Voronoi path is interpolated with a series of cubic splines, assigning a cost to each obstacle location (Chan and Foddy 1985).

## 1.6 Probabilistic Methods

Probabilistic methods work by a random selection of neighbourhood points that meet some metric such as the shortest length resulting in the probabilistic random road map (PRR) method. This method samples the given space for probable solutions in the form of a network of graphs. It normally uses uniform sampling of a given space. The path planning problem is treated as a search problem in the partitioned cells (Eagle and Yee 1990). Probabilistic road maps (PRMs) (Kavraki *et al*. 1996; Pettersson and Doherty 2004) connect the starting point to the goal point by adding successive trajectories to a pre-computed route. In another approach called rapidly exploring random trees (RRTs) (Cheng *et al*. 2001; LaValle 1998), a tree of trajectory segments is extended from the start point to the goal point. Every successive trajectory in the tree is selected randomly by connecting to a closest point in the existing tree. In Amin *et al*. (2006), the path planning is achieved by the RRT and is further enhanced by using the Dijkstra search algorithm, which finds the route for the UAV flying among known static obstacles represented as quadtrees. Probabilistic methods are applied to path planning by considering the positional uncertainty of threat regions in Jun and D'Andrea (2003). The final path is refined with circular arcs at the points of line joining.

## 1.7 Potential Field

The potential field method was first proposed by Khatib (1985). In this method, the environment is represented as an artificial potential field. The destination point is assigned an attractive potential, while the obstacles are assigned repulsive potentials. The idea is that a robot moving in the field will be attracted towards the destination, while being repelled by the obstacles. In contrast to the road map and cell decomposition methods, the path resulting from this method follows the line of maximum potential of a continuous field function. However, this method has some drawbacks, the main one being that the vehicle may get trapped in a local maximum, for example, when encountering a C-shaped obstacle (Borenstein and Koren 1991; Koren and Borenstein 1991). However, improved versions of this method have been developed to eliminate the local maxima and to reduce the computational complexity. A variation of the technique using multiple temporary attraction points together with genetic algorithms (GAs) are used in Arámbula and Padilla (2004). The potential field algorithm (Eun and Bang 2006) solves the path planning by generating an attractive field towards the goal point and a repulsive field at the obstacles. In another approach (Polymenakos *et al*. 1998; Tsitsiklis 1995), a Dijkstra-like method (Dijkstra 1959) is suggested for solving a continuous-space shortest-path problem in a 2D plane by optimization. An analytical and discrete optimization approach has been used (Zabarankin *et al*. 2002) for optimal risk path generation in 2D space with constant radar cross-section, arbitrary number of sensors and a constraint on path length.

## 1.8 Cell Decomposition

In the cell decomposition method, the environment is divided into non-overlapping cells. Possible routes are then generated that pass through adjacent free cells. Free cells are ones that are not occupied by obstacles. The obstacles are isolated by finding the connectivity between the free cells. Thus a discrete version of the environment is produced. A search algorithm is used to connect adjacent free cells. Figure 1.9 shows the schematic of the process. The shaded (cross-hatched) cells are eliminated because they are occupied by obstacles (grey). A connectivity between the start and finish points is found by connecting the free cells by a series of straight lines.

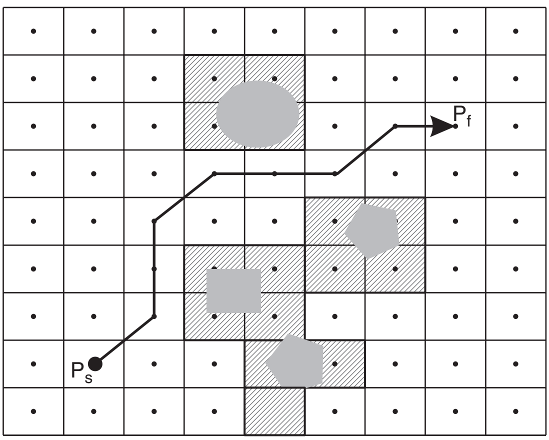


**Figure 1.9** Cell decomposition

The figure shows a simple division of the environment, which is called the exact method of cell decomposition. An approximate method is proposed to reduce the memory consumption in Kambhampari and Davis (1986). This is also called the quadtree or octree representation. Further improvements of this method can be found in Jung and Gupta (1996).

## 1.9 Optimal Control

Optimal control is the most natural way to solve problems involving objective functions, constraints and boundary conditions. However, the dimension and complexity of optimal control problems cause a heavy burden on computational time in the solution. Also, the nature of the problem may require either suboptimal or feasible solutions rather than the optimal solution. It is quite difficult to solve path planning problems in a cluttered environment using optimal control theory (Duleba and Sasiadek 2003; Sasiadek and Duleba 2000). General planar motion is a combination of straight-line and circular trajectories, which are proved for a mobile robot using Pontryagin's maximum principle in Balkcom and Mason (2000). Optimization techniques such as probabilistic methods, mixed integer linear programming and genetic programming have been applied to path planning of UAVs. These techniques produce paths by optimizing certain cost functions. The cost functions differ based on the applications, such as minimum time of arrival, optimizing fuel consumption and coordinated motion. They are mostly search algorithms.

## 1.10 Optimization Techniques

The challenges and complexity of path planning with optimization techniques and uncertainties are discussed in Patcher and Chandler (1998). In Rabbath *et al*. (2004) an overview of coordinated control of UAVs and their complexities is presented. Branch and bound optimization is used for path planning and is compared with mixed integer linear programming (MILP) in Eele and Richards (2009).

The use of MILP for path planning applications can be found in Ademoye *et al*. (2006), Schouwenaars *et al*. (2001, 2006), Richards *et al*. (2001) and Richards and How (2002). MILP is an application of the operations research method, called linear programming with integer or binary constraints. These constraints are used for logical decisions, such as turn left and move up. This method produces a safe route for UAVs. But the route has to be smoothed further to make it flyable. Also, the optimization methods are associated with long computational time. Accomplishing the mission objectives with the physical and functional limitations of UAVs further increases the complexity of the solution to the path planning problem (Mitcheel and Sastry 2003; Yang and Kapila 2002).

Evolutionary algorithms are used in Mittal and Deb (2007), where the path length, risk of collision and limits on maximum and minimum heights are optimized in path planning of a UAV. The discrete points generated by the planner are connected by *B*-spline curves for producing flyable paths. Also the evolutionary optimization principle in path planning can be seen in Dong and Vagners (2004), Zheng *et al*. (2005) and Nikolos *et al*. (2003). A genetic algorithm is used in Wu *et al*. (2007) for path planning of autonomous underwater vehicles for surveying a given area.

## 1.11 Trajectories for Path Planning

The optimization methods, randomized search approaches and Voronoi diagram approach use exhaustive search and computational methods that result in route planning. The route planning does not consider the kinematic constraints of the path. Also, the reactive behaviour of the UAV needs a flyable path at any point of its flight. In such a situation, route planning would be a handicap. For this reason, it appeared reasonable to attempt to use the curves directly in path planning. In this manner, planar and spatial Dubins paths (Dubins 1957; Shanmugavel *et al*. 2006b, 2007c), Pythagorean hodographs (Shanmugavel *et al*. 2006a, 2007b) and 2D clothoids (Shanmugavel *et al*. 2007a) have been used to solve the problem of simultaneous arrival on target. Though flyable paths are essential for manoeuvring, straight-line trajectories are used in other applications, such as task allocation for multiple robot problems in Zhang *et al*. (2008) and Shima *et al*. (2005). The Dubins path is used for airborne problems (Bicchi and Pallottino 2000; Massink and Francesco 2001; Robb *et al*. 2005; Shanmugavel *et al*. 2006b). The simultaneous arrival of UAVs using the Dubins path is studied in Shanmugavel *et al*. (2005, 2006b), where the path is produced using differential geometry. Parametric curves are used for path planning in Shanmugavel *et al*. (2006a) and Nikolos *et al*. (2003).

A comprehensive review of the types of paths used for path planning is presented in Segovia *et al*. (1991). Dubins (1957) showed in his work that the shortest path between two vectors in a plane that meets a minimum bound on turning radius is a composite path formed by line and circular arc segments. This paper received widespread attention by the research community and is extensively cited in ground robotics works (Bui *et al*. 1994; Latombe 1991; McGee *et al*. 2005; Shkel and Lumelsky 1996). This work has also motivated the derivation of the shortest path for a vehicle that can move forwards and backwards for parking a car. Reeds and Shepp (1990) developed the shortest path for a vehicle that can move both forwards and backwards. Clothoid arcs are used for the path planning of car-like vehicles using a composite path made of circular and clothoid arcs (Laumond 1986; Scheuer and Fraichard 1997). The Cornu spiral or clothoid curve has been used (Dai and Cochran 2009; Shanmugavel *et al*. 2009) for path planning of multiple UAVs. $B$-splines (Komoriya and Tanie 1989), quintic polynomials (Takahashi *et al*. 1989), polar splines (Nelson 1989), clothoids (Liscano and Green 1989), cubic spirals (Kanayama and Hartman 1989) and $G^2$-splines (Piazzi and Bianco 2000) have all been used for path planning of mobile robots. Robot path planning using the Voronoi diagram has been studied widely since the mid-1980s (Iyengar *et al*. 1986), and in the late 1990s the focus on coordinated path planning of multiple robots began. Much of the work done on path planning is carried out in ground robotics, but their approaches cannot be applied directly to unmanned aerial vehicles. The path of a UAV is limited by high-$G$ turns and also by the fact that it has a threshold speed below which it cannot fly.

## 1.12 Outline of the Book

The book is divided into seven chapters. Chapter 1 introduces the importance of path planning for an autonomous system. Current research on path planning approaches is discussed in this chapter. As path planning is borrowed from the mobile robotics field, some focus on techniques used in this field are also discussed.

Chapter 2 deals with producing flyable paths in two dimensions. Three types of paths are considered. The chapter begins with the design of Dubins path, because this is the shortest path between two poses in 2D and also it is simple. The Dubins path is produced using the principles of Euclidean and differential geometries. Besides the equivalence of the results from the two approaches, it is shown that the differential geometric principles are advantageous in generalisation of the path. The lack of curvature continuity of the Dubins path motivates the use of other paths. In this respect, a single path (Pythagorean hodograph) and a composite path (clothoid and line segments) are considered. The circular arcs in 2D Dubins paths are approximated with clothoid segments to produce a smooth path. The last part deals with the Pythagorean hodograph (PH) curve known for its rational properties. A procedure is established to derive a PH path of curvature continuity.

Chapter 3 discusses three-dimensional path planning. It extends the principles used in Chapter 2. The Dubins path is produced in 3D using the principles of differential geometry. The clothoid path is not discussed, as the design is similar to that of the 3D Dubins path. The PH path is developed in 3D for use in path planning. The composite version of the Dubins path is generated by finding the common intersecting plane between the initial and final poses with an initial rotation at the start pose. The spatial PH path is developed with quaternion form and the curvature and torsion are met by increasing the tangent vectors at the initial and final poses.

Chapter 4 describes algorithms for detecting and avoiding threats or obstacles. Detection precedes avoidance. The chapter is divided into two major parts. The first part discusses obstacle avoidance in a static obstacle environment, while the second part discusses a strategy for avoiding dynamic obstacles. Obstacle avoidance in both 2D as well as 3D cases are considered. Two methods are used to avoid fixed obstacles: (i) decreasing the curvature, and (ii) generating intermediate waypoints. The methods are simulated for Dubins, clothoid and PH paths. For 3D path planning, a representative building area is considered for the simulation. The area is discretized to a searchable database. The Dijkstra algorithm is used to find the shortest route through the database. Following this, the waypoints are connected with flyable paths.

Chapter 5 presents a path-following algorithm for the Dubins path, which is required by the UAV to follow the feasible path. It develops two forms of guidance algorithm to follow Dubins and PH paths, using a simple linear guidance algorithm and a nonlinear guidance algorithm based on nonlinear dynamic inversion. The chapter then goes on to develop a guidance algorithm to avoid collision with mobile obstacles, which could be other UAVs or aircraft and other flying vehicles not under group control.

Chapter 6 discusses the problem undertaken for path planning, assumptions, problem formulation and solution description for the simultaneous arrival on target of a group of UAVs. An algorithm is used for the path planning. Handling the constraints involved by means of curvature variation is discussed. The flyable paths are tested for safety conditions. A solution is achieved by increasing the lengths over that of a reference path.

## References


Ademoye, T. A., Davari, A. and Cao, W. 2006. Three dimensional obstacle avoidance maneuver planning using mixed integer linear programming. *Proc. Robotics and Applications 2006*, Honolulu, 14–16 August. ACTA Press.

Amin, J. N., Bošković, J. D. and Mehra, R. K. 2006. A fast and efficient approach to path planning for unmanned vehicles. *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, Colorado, 21–24 August. AIAA 2006–6103.

Arámbula Cosío, F. and Padilla Castañeda, M. A. 2004. Autonomous robot navigation using adaptive potential fields. *Mathematical and Computer Modelling*, **40**(9–10), 1141–1156.

Balkcom, D. J. and Mason, M. T. 2000. Extremal trajectories for bounded velocity differential drive robots. *Proc. IEEE Int. Conf. on Robotics and Automation*, San Francisco, pp. 2479–2484.

Beard, R. W., McLain, T. W., Goodrich, M. A. and Anderson, E. P. 2002. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation*, **18**(6), 911–922.

Bender, J. G. 1991. An overview of systems studies of automated highway systems. *IEEE Transactions on Vehicular Technology*, **40**, 82–99.

Bicchi, A. and Pallottino, L. 2000. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, **1**, 3578–3583.

Blyenburgh, van, P. 1999. UAVs: an overview. *Air & Space Europe*, **1**(5/6), 43–47.

Borenstein, J. and Koren, Y. 1991. The vector field histogram – fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, **7**, 278–288.

Bortoff, S. 2000. Path-planning for unmanned air vehicles. *Proc. American Control Conf*., Chicago, pp. 364–368.

Bui, X. N., Boissonnat, J. D., Soueres, P. and Laumond, J. P. 1994. Shortest path synthesis for Dubins non-holonomic robots. *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 2–4.

Chan, Y. K. and Foddy, M. 1985. Real time optimal flight path generation by storage of massive data bases. *IEEE NEACON*, New York, pp. 516–521.

Chandler, P., Rasmussen, S. and Pachter, M. 2000. UAV cooperative path planning. *AIAA Guidance, Navigation, and Control Conf*., Denver. AIAA-2000-4370.

Cheng, P., Shen, Z. and LaValle, S. M. 2001. RRT-based trajectory design for autonomous automobiles and spacecraft. *Archives of Control Sciences*, **11**, 167–194.

Dai, R. and Cochran, J. E. 2009. Path planning for multiple unmanned aerial vehicles by parameterized Cornu-spirals. *American Control Conf*., pp. 2391–2396.

Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**, 269–271.

Dong, J. and Vagners, J. 2004. Parallel evolutionary algorithms for UAV path planning. *AIAA 1st Intelligent Systems Technical Conf*., Chicago, 20–22 September.

Dubins, L. E. 1957. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangent. *American Journal of Mathematics*, **79**, 497–516.

Duleba, I. and Sasiadek, J. 2003. Nonholonomic motion planning based on Newton algorithm with energy constraints. *IEEE Transactions on Control Systems Technology*, **11**, 355–363.

Eagle, J. and Yee, J. 1990. An optimal branch-and-bound procedure for the constrained path, moving target search problem. *Operations Research*, **28**, 110–114.

Eele, A. and Richards, A. 2009. Path planning with avoidance using nonlinear branch-and-bound optimization. *Journal of Guidance, Control, and Dynamics*, **32**(2), 384–394.

Eun, Y. and Bang, H. 2006. Cooperative control of multiple unmanned aerial vehicles using the potential field theory. *Journal of Aircraft*, **43**, 1805–1814.

Fong, T., Thorpe, C. and Baur, C. 2003. Multi-robot remote driving with collaborative control. *IEEE Transactions on Industrial Electronics*, **50**(4), 699–704.

Fowler, J. M. and D'Andrea, R. A. 2003. Formation flight experiment. *IEEE Control Systems Magazine*, 35–43.

Gennery, D. B. 2004. Traversability analysis and path planning for a planetary rover. *Autonomous Robots*, 131–146.

Hebert, J., Jacques, D., Novy, M. and Pachter, M. 2001. Cooperative control of UAVs. *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, Montreal. AIAA-2001-4240.

Howard, A., Parker, L. and Sukatme, G. 2006. Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection. *International Journal of Robotics Research*, **25**(5–6), 431–447.

Iyengar, S., Jorgensen, C., Rao, N. and Weisbin, R. 1986. Robot navigation algorithm using learned spatial graphs. *Robotica*, **4**, 93–100.

Judd, K. and McLain, T. W. 2001. Spline based path planning for unmanned air vehicles. *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, Montreal. AIAA-2001-4238.

Jun, M. and D'Andrea, R. 2003. Path planning for unmanned aerial vehicles in uncertain and adversarial environments. In *Cooperative Control: Models, Applications and Algorithms* (eds Butenko, R., Murphey, R. and Pardalos, P.), pp. 95–110. Kluwer Academic.

Jung, D. and Gupta, K. K. 1996. Octree-based hierarchical distance maps for collision detection. *Proc. IEEE Int. Conf. on Robotics and Automation*, vol. 1, pp. 454–459.

Kambhampari, L. S. and Davis, L. S. 1986. Multi-resolution path planning for mobile robots. *IEEE Journal of Robotics and Automation*, **2**(3), 135–145.

Kanayama, Y. and Hartman, B. I. 1989. Smooth local path planning for autonomous vehicles. *Proc. IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, pp. 1265–1270.

Kavraki, L. E., Svestka, P., Latombe, J.-C. and Overmars, M. 1996. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, **12**, 566–580.

Khatib, O. 1985. Real time obstacle avoidance for manipulators and mobile robots. *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 500–505.

Komoriya, K. and Tanie, K. 1989. Trajectory design and control of a wheel-type mobile robot using *B*-spline curve. *IEEE–RSJ Int. Conf. on Intelligent Robots and Systems*, Tsukuba, pp. 398–405.

Koren, Y. and Borenstein, J. 1991. Potential field methods and their inherent limitations for mobile robot navigation. *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1398–1404.

Latombe, J. C. 1991. *Robot Motion Planning*. Kluwer Academic.

Laumond, J. P. 1986. Feasible trajectories for mobile robots with kinematic and environment constraints. *Proc. Int. Conf. on Intelligent Autonomous Systems*, Amsterdam, pp. 346–354.

LaValle, S. 1998. Rapidly-exploring random trees: a new tool for path planning. Technical Report.

LaValle, S. M. 2006. *Planning Algorithm*. Cambridge University Press.

Lipschutz, M. 1969. *Schaum's Outline of Differential Geometry*. McGraw-Hill.

Liscano, R. and Green, D. 1989. Design and implementation of a trajectory generator for an indoor mobile robot. *IEEE–RSJ Int. Conf. on Intelligent Robots and Systems*, Tsukuba, pp. 380–385.

Massink, M. and Francesco, N. 2001. Modelling free flight with collision avoidance. *Proc. IEEE Int. Conf. on Engineering of Complex Computer Systems*, pp. 270–279.

McGee, T. G., Spry, S. and Hedrick, J. K. 2005. Optimal path planning in a constant wind with a bounded turning rate. *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, San Francisco.

McLain, L. and Beard, R. W. 2000. Trajectory planning for coordinated rendezvous of unmanned air vehicles. *AIAA Guidance, Navigation, and Control Conf*., Denver. AIAA-2000-4369.

Mitcheel, I. M. and Sastry, S. 2003. Continuous path planning with multiple constraints. *IEEE Conf. on Decision and Control*, pp. 5502–5507.

Mittal, S. and Deb, K. 2007. Three dimensional offline path planning for UAVs using multiobjective evolutionary algorithms. *Proc. IEEE Congress on Evolutionary Computation, CEC 2007*, Singapore, 25–28 September.

Murray, R. M. 2007. Recent research in cooperative control of multi-vehicle systems. *Journal of Dynamic Systems, Measurement and Control*, **129**(5), 571–583.

Nelson, W. L. 1989. Continuous curvature paths for autonomous vehicles. *IEEE Int. Conf. on Robotics and Automation*, Scottsdale, AZ, vol. 3, pp. 1260–1264.

Nikolos, I., Valavanis, K., Tsourveloudis, N. and Kostaras, A. 2003. Evolutionary algorithm based offline/online path planner for uav navigation. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, **33**, 898–912.

NMAB and ASEB (National Materials Advisory Board, and Aeronautics and Space Engineering Board) 2000. *Uninhabited Air Vehicles: Enabling Science for Military Systems*. National Academy Press.

Oliveira, P., Pascoal, A., Silva, V. and Silvestre, C. 1998. Mission control of the MARIUS AUV: system design, implementation, and sea trials. *International Journal of Systems Science*, **29**, 1065–1080.

OSD (Office of the Secretary of Defense) 2005. *Unmanned Aircraft Systems Roadmap 2005–2030*. http://ftp.fas.org/irp/program/collect/uav_roadmap2005.pdf

Patcher, M. and Chandler, P. R. 1998. Challenges of autonomous control. *IEEE Control Systems Magazine*, pp. 93–97.

Pettersson, P. O. and Doherty, P. 2004. Probabilistic roadmap based path planning for an autonomous unmanned aerial vehicle. In *Workshop on Connecting Planning and Theory with Practice* at *14th Int. Conf. on Automated Planning and Scheduling, ICAPS*.

Piazzi, A. and Bianco, C. L. 2000. Quintic $G^2$ splines for trajectory planning of autonomous vehicles. *IEEE Intelligent Vehicles Symp*., Dearborn, MI, pp. 193–200.

Polymenakos, L. C., Bertsekas, D. P. and Tsitsiklis, J. N. 1998. Implementation of efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, **43**, 278–283.

Qu, Z. 2008. *Cooperative Control of Dynamical Systems*. Springer.

Rabbath, C., Gagnon, E. and Lauzon, M. 2004. On the cooperative control of multiple unmanned aerial vehicles. *IEEE Canadian Review*, 8–15.

Reeds, J. A. and Shepp, R. A. 1990. Optimal paths for a car that goes both forward and backward. *Pacific Journal of Mathematics*, **145**(2), 367–393.

Ren, W., Beard, R. W. and McLain, T. W. 2004. Coordination variables and consensus building in multiple vehicle systems. In *Cooperative Control* (eds Kumar, V., Leonard, N. and Morse, A. S.), pp. 171–188. Springer.

Richards, A. and How, J. 2002. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. *American Control Conf*., pp. 1936–1941.

Richards, A., How, J., Schouwenaars, T. and Feron, E. 2001. Plume avoidance maneuver planning using mixed integer programming. *AIAA Guidance, Navigation, and Control Conf*.. AIAA-2001-4091.

Robb, M., White, B. A., Tsourdos, A. and Rulloda, D. 2005. Reachability guidance: a novel concept to improve mid-course guidance. *American Control Conf*., pp. 339–345.

Roberts, G. C., Ramana, M. V., Corrigan, C., Kim, D. and Ramanathan, V. 2008. Simultaneous observations of aerosol–cloud–albedo interactions with three stacked unmanned aerial vehicles. *Proceedings of the National Academy of Sciences of the USA*, **105**(21), 7370–7375.

Sasiadek, J. and Duleba, I. 2000. 3D local trajectory planner for UAV. *Journal of Intelligent and Robotic Systems: Theory and Applications*, **29**, 191–210.

Scheuer, A. and Fraichard, T. 1997. Collision-free continuous curvature path planning for car-like robots. *IEEE Int. Conf. on Robotics and Automation*, Albuquerque, NM, pp. 867–873.

Schouwenaars, T., Moor, B., Feron, E. and How, J. 2001. Mixed integer programming for multi-vehicle path planning. *European Control Conf*., Porto, Portugal.

Schouwenaars, T., Feron, E. and How, J. 2006. Multi-vehicle path planning for non-line of sight communication. *American Control Conf*., Minnesota, 14–16 June.

Segovia, S., Rombaut, M., Preciado, A. and Meizel, D. 1991. Comparative study of the different methods of path generation for a mobile robot in a free environment. *Proc. Int. Conf. on Advanced Robotics*, pp. 1667–1670.

Shamma, J. S. (ed.) 2007. *Cooperative Control of Distributed Multi-Agent Systems*. John Wiley & Sons, Ltd, Chichester.

Shanmugavel, M., Tsourdos, A., Żbikowski, R. and White, B. A. 2005. Path planning of multiple UAVs using Dubins sets. *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, San Francisco. AIAA-2005-5827.

Shanmugavel, M., Tsourdos, A., Żbikowski, R., White, B. A., Rabbath, C. A. and Lechevin, N. 2006a. A solution to simultaneous arrival of multiple UAVs using Pythagorean hodograph curves. *Proc. American Control Conf*., Minneapolis, 14–16 June.

Shanmugavel, M., Tsourdos, A., Żbikowski, R. and White, B. A. 2006b. 3D Dubins sets based coordinated path planning for swarm of UAVs. *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, Keystone, CO. AIAA-2006-6211.

Shanmugavel, M., Tsourdos, A., Żbikowski, R. and White, B. A. 2007a. Path planning of multiple UAVs with clothoid curves. *17th IFAC Symp. on Automatic Control in Aerospace*, Toulouse, France.

Shanmugavel, M., Tsourdos, A., Żbikowski, R. and White, B. A. 2007b. 3D path planning for multiple UAVs using Pythagorean hodograph curves. *AIAA Guidance, Navigation, and Control Conf. and Exhibit*, Hilton Head, SC.

Shanmugavel, M., Tsourdos, A., White, B. A. and Żbikowski, R. 2007c. Differential geometric path planning of multiple UAVs. *AMSE Journal of Dynamic Systems, Measurement and Control*, **129**(5), 620–632.

Shanmugavel, M., Tsourdos, A., White, B. A. and Żbikowski, R. 2009. Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs. *Control Engineering Practice*, **17**, in press.

Shima, T., Rasmussen, S. and Sparks, A. 2005. UAV cooperative multiple task assignments using genetic algorithms. *Proc. American Control Conf*., Oregon, vol. 5, pp. 2989–2994.

Shkel, A. and Lumelsky, V. 1996. On calculation of optimal paths with constrained curvature: the case of long paths. *IEEE Int. Conf. on Robotics and Automation*, pp. 3578–3583.

Shladover, S. E. 1991. Automatic vehicle control developments in the PATH program. *IEEE Transactions on Vehicular Technology*, **40**, 114–130.

Smith, T. R., Hansmann, H. and Leonard, N. E. 2001. Orientation control of multiple underwater vehicles. *Proc. 40th IEEE Conf. on Decision and Control*, pp. 4598–4603.

Takahashi, A., Hongo, T. and Ninomiya, Y. 1989. Local path planning and control for AGV in positioning. *IEEE–RSJ Int. Conf. on Intelligent Robots and Systems*, Tsukuba, pp. 392–397.

Tompkins, P., Stentz, A. and Wettergreen, D. 2004. Global path planning for Mars rover exploration. *IEEE Aerospace Conf*., vol. 2, pp. 801–815.

Tsitsiklis, J. N. 1995. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, **40**, 1528–1538.

Uny Cao, Y., Fukunaga, A. and Kahng, A. 1997. Cooperative mobile robotics: antecedents and directions. *Autonomous Robots*, **4**, 7–27.

Vian, J. L. and More, J. R. 1989. Trajectory optimization with risk minimisation for military aircraft. *AIAA Journal of Guidance, Control and Dynamics*, **12**, 311–317.

Wilson, J. 2007. UAV worldwide roundup 2007. *Aerospace America*, pp. 30–37.

Wu, X., Feng, Z., Zhu, J. and Allen, R. 2007. GA-based path planning for multiple AUVs. *International Journal of Control*, **80**(7), 1180–1185.

Yang, G. and Kapila, V. 2002. Optimal path planning for unmanned air vehicles with kinematic and tactical constraints. *IEEE Conf. on Decision and Control*, pp. 1301–1306.

Yuh, J. 1995. Developments in underwater robotics. *Proc. IEEE Conf. on Robotics and Automation*, pp. 1862–1867.

Yuh, J. 2000. Underwater robotics. *Proc. IEEE Conf. on Robotics and Automation*, pp. 932–937.

Zabarankin, M., Uryasev, S. and Pardalos, P. 2002. Optimal risk path algorithms. In *Cooperative Control and Optimization* (eds Murphey, R. and Pardalos, P.), pp. 271–303. Kluwer Academic.

Zhang, D., Wang, L. and Yu, J. 2008. Geometric topology based cooperation for multiple robots in adversarial environments. *Control Engineering Practice*, **16**, 1092–1100.

Zheng, C., Li, L., Xu, F., Sun, F. and Ding, M. 2005. Evolutionary route planner for unmanned air vehicles. *IEEE Transactions on Robotics*, **21**, 609–620.