Robotica (2016) volume 34, pp. 823–836. © Cambridge University Press 2014
doi:10.1017/S0263574714001878

Optimal cooperative path planning of unmanned aerial vehicles by a parallel genetic algorithm

Hamed Shorakaei†*, Mojtaba Vahdani‡, Babak Imani§ and Ali. Gholami¶

†*Department of mechanical engineering, Asadabad Branch, Islamic Azad University, Asadabad, Iran*
‡*Imam Hossein University, Tehran, Iran*
§*Department of mechanical engineering, Harsin Branch, Islamic Azad University, Harsin, Iran*
¶*Imam Hossein University, Tehran, Iran*

(Accepted June 15, 2014. First published online: July 24, 2014)

SUMMARY
The current paper presents a path planning method based on probability maps and uses a new genetic algorithm for a group of UAVs. The probability map consists of cells that display the probability which the UAV will not encounter a hostile threat. The probability map is defined by three events. The obstacles are modeled in the probability map, as well. The cost function is defined such that all cells are surveyed in the path track. The simple formula based on the unique vector is presented to find this cell position. Generally, the cost function is formed by two parts; one part for optimizing the path of each UAV and the other for preventing UAVs from collision. The first part is a combination of safety and length of path and the second part is formed by an exponential function. Then, the optimal paths of each UAV are obtained by the genetic algorithm in a parallel form. According to the dimensions of path planning, genetic encoding has two or three indices. A new genetic operator is introduced to select an appropriate pair of chromosome for crossover operation. The effectiveness of the method is shown by several simulations.

KEYWORDS: Optimal cooperative path planning; Parallel genetic algorithm; Probability map; Unmanned aerial vehicle / UAV; Cost function.

1. Introduction

Some of the robots and autonomous vehicles are deployed to perform missions in hazardous environments, such as operations in nuclear power plants, and space explorations and to monitor and spy enemy forces on the battlefield. The production of a smart unmanned aircraft vehicle (UAV) is important for reducing human losses and injuries. One main challenge to the development of UAVs is path planning in hostile environments. In a path planning method, finding a path in which there is no collision with obstacles or other UAVs proves to be difficult.

The path planning for autonomous systems is performed based on two different domains the most common of which is based on the geometry and kinematics of the systems. In this category, the path planning is performed using the environment (obstacles) and the main goal is usually to find the shortest path without collision that is of significance in the geometry of the system. Another area of the path planning is the dynamics of the system, so that dynamic equations incorporate the system as a constraint into path planning equations and the obtained path should be met in the dynamic equations, as well. In this case, it is also possible to enter the engine output torque limits into dynamical equations. The aim of path planning in this issue is to obtain a path with minimum energy input, minimum kinetic energy, or minimum jerk. The first type is usually used for real-time applications.

* Corresponding author. E-mail: h.shorakaei@basu.ac.ir

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

One of the main targets for UAV path planning in hostile environments is to achieve an ideal target point. If there is some information on the presence of environment threats, a safe path could be always planned. In ref. [1], a path planning algorithm based on the threat probability map is proposed. The cost function of the proposed algorithm has two terms: one is concerned with path length and the other is related to the potential path threat. The authors of[1] optimized the cost function using Bellman-Ford algorithm, and also used the changed probability map of one of UAVs in potential conflict points for the path planning of Multi-UAVs. In ref. [2], there is proposed a real-time path planning based on an algorithm capable of working with limited information on workplace for fully autonomous UAVs in hostile environments. The proposed algorithm is composed of two main parts that consider the effect of different assumptions on data and threats; the direct collocation that approximates states and controls piecewise polynomials is extensively utilized in space applications and manned aircrafts. However, it has limited applications in UAV. In ref. [3] the optimal path planning for some UAVs using direct collocation and nonlinear programming is proposed. The authors of[3] successfully applied the algorithm to a UAV path in order to provide the minimum observation time for a video camera mounted on the UAV. They considered some UAVs to provide the minimum sensor coating time using a combination of the UAVs. Shanmugavel *et al.* also described a path planning for a group of UAVs.[4] Their main goal was the simultaneous arrival of some UAVs on the target point. The problem was formulated to generate possible paths (flyable and safe) and solved in three phases. In the first phase, the flyable paths were generated; in the second phase, the additional constraints were added to the problem to produce safe paths so that none of the UAVs collide others or observable obstacles in the environment, and also in the third phase some paths were produced for simultaneous arrival. The paths were generated by using differential geometry rules in the first phase. The paths, in the second phase, were manipulated to increase safety using meeting safety constraints. Finally, in the third phase, the simultaneous entry can be achieved by creating paths with equal length.[4] In ref. [5], a cooperative attack path planning problem for UAVs swarm located in grid model and Bi-level programming was solved. They used Verona map for the geometric modeling of the space in which flight environment was divided into rectangular spaces with equal sizes. They then proposed a Bi-level programming scheme and divided the cooperative attack path planning into two layers: path planning layer and cooperative design layer. At last, the optimal candidate paths were obtained by ACO algorithms.[5] In ref. [6], a strategy for gaining cooperation time among the groups of vehicles is offered. The initial paths are constructed based on Voronoi diagram and based on that, the smoothing paths are designed for the Waypoint Path Planner (WPP). Then, it considers the cooperative timing problem and proposes a particle swam optimization algorithm for simultaneous attack. In ref. [7] a Co-evolutionary improved Genetic Algorithm (CIGA) for global path planning of multiple mobile robots is proposed. This algorithm employs a co-evolution mechanism together with an improved genetic algorithm (GA). This improved GA proposes a new genetic modification operator and is compared with conventional gas. The use of a co-evolution mechanism takes into full account the cooperation between populations, which avoid collision between mobile robots and are conductive for each mobile robot to obtain an optimal or near-optimal collision-free path.[7] But this algorithm is not applicable for a 3D application and does not consider probability map. In refs. [8] and [9], there is good information for real time application. Falomi *et al.*[9] is based on determination of wheel–rail contact points and develops a method which is reliable in terms of precision and can be implemented on-line, assuring a calculation time consistent with real-time calculations of multi body dynamics. An algorithm to determine the shortest trajectory of a fixed-wing UAV in scenarios with no-fly areas is proposed in ref. [10] in which the flight performance and the navigational capabilities are considered. Proposed algorithm in ref. [10] is based on a discretization of the airspace by a specific network. But this algorithm obtains a local minimum path. However, this algorithm has a little chance to obtain global minimum path. In ref. [11] the flight of an unmanned aerial vehicle under complicated conditions and hazards has been considered. Hazards are given in terms of a 2D relief. To find the optimal 2D path, the risk of the given constraints on flight time and velocity is minimized. The spatial risk distribution is described by the probability density of a Gaussian random variable. The found 2D path is used to construct the polynomial approximation of the 3D path taking the local relief into account. Due to use of the polynomial, search space to find minimum path is small and the 3D path is a local path rather than an appropriate path.[13] The performance of a real-time multi body dynamics simulation using linearized constraint equations is discussed in ref. [14]. The comparison of a linearized vs. non-linearized multi body vehicle model for a real-time simulation is performed in ref. [14].

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

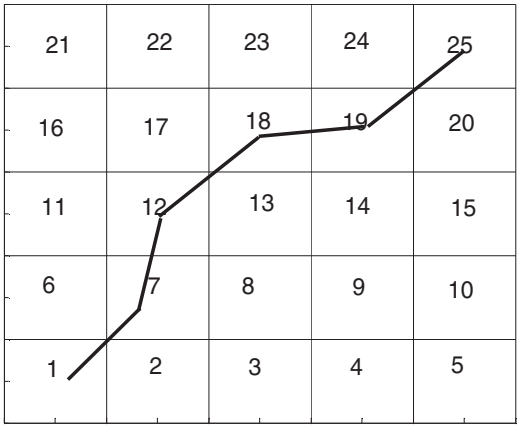


Fig. 1. An example of path.

In this paper, an optimal cooperative path planning (OCPP) algorithm for some UAVs based on parallel genetic algorithm (PGA) is shown. The inputs of this algorithm are the probability map and the target of each UAV. The probability map is graded based on the probability of non-neutralizing UAV and obstacles. The cost function of each UAV is a linear combination of two terms one of which is about path length and the other is a function of the probability of non-neutralizing UAV. The most significant advantage of this algorithm is that each UAV can have individually a target and a different cost function. Hence, the optimizations of the UAVs are parallel. Among other advantages of this algorithm is its applicability for real-time applications, yet simple with a high efficiency, a low computational cost and also the ability to run in two dimensions and three dimensions.

2. Probability Map

The problem input is the probability map of environment and the aim is to design a safe and short path to achieve a target point. The environment probability map is graded with a specific scale, while considering equipment and devices in each cell, a specific value of it is allocated. This value is the probability of non-neutralizing UAV in the cell. The basic concept in designing the probability map is similar to occupancy maps based on grid in refs. [11] and [12]. The probability value of each cell is obtained considering its occupancy amounts. The occupancy values for each cell are determined based on reading sensors and by applying the conditional probability of occupancy using Bayes' rule. These values are determined by the sensor characteristics, the location of the sensors, the measurement methods, etc.[1] Figure 1 shows an example of the celled environment and also a traversed path. The traversed path corresponding to Fig. 1 passes through 1,7,12,18,19,25.

Assume that the region R is considered for the mission and R is composed of n cells. The following events are defined as follows:

D_i: event in which the UAV in cell "i" is detected by the adversary.
ε_i: event in which the adversary is in cell "i".
S_i: event in which the UAV in cell "i" is neutralized by the adversary.

The complement of event A is denoted by $\bar{A}$. If X_i represents ith sensor reading, the probability of occupancy of cell i by the formula can be expressed as:

$$P(\varepsilon_i|X_1, ..., X_k) = 1 - \left(1 + \frac{P(\varepsilon_i|X_1)}{1 - p(\varepsilon_i|X_1)} \prod_{j=2}^{k} \frac{P(\varepsilon_i|X_j)}{1 - P(\varepsilon_i|X_j)} \frac{1 - p(\varepsilon_i)}{P(\varepsilon_i)}\right)^{-1} \quad (1)$$

The probability for which the UAV in cell i is neutralized by the adversary is

$$P(S_i) = P(S_i|D_i)P(D_i) + P(S_i|\bar{D}_i)P(\bar{D}_i) = P(S_i|D_i)P(D_i) \quad (2)$$

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

Where we assume the probability of non-neutralizing UAV is zero when it will not be detected by the adversary. Probability $P(D_i)$ can be expressed as follows:

$$\begin{aligned} P(D_i) &= P(D_i(\varepsilon_i|X_1, ..., X_k))P(\varepsilon_i|X_1, ..., X_k) + P(D_i(\bar{\varepsilon}_i|X_1, ..., X_k))P(\bar{\varepsilon}_i|X_1, ..., X_k) \\ &= P(D_i(\varepsilon_i|X_1, ..., X_k))p_i + P(D_i(\bar{\varepsilon}_i|X_1, ..., X_k))(1 - p_i) \end{aligned} \tag{3}$$

We have

$$P(S_i) = P(S_i|D_i)\{P(D_i(\varepsilon_i|X_1, ..., X_k))p_i + P(D_i(\bar{\varepsilon}_i|X_1, ..., X_k))(1 - p_i)\} \tag{4}$$

We assume that we can estimate or calculate the probability $P(S_i|D_i)$, p_i, $P(D_i(\varepsilon_i|X_1, ..., X_k))$ and $P(D_i(\bar{\varepsilon}_i|X_1, ..., X_k))$ from prior information of the adversary. The probability $P(\bar{S})$, i.e. the probability for which UAV is not neutralized by the adversary when it follows path $(a_1, a_2, \ldots, a_l)$ can be expressed as:

$$P(\bar{S}) = \prod_{i \in (a_1, a_2, ..., a_l)} P(\bar{S}_i) \tag{5}$$

The objective is to find the path $(a_1, a_2, \ldots, a_l)$ that maximizes the probability $P(\bar{S})$ when cell a_1 is the origin of the mission and cell a_l is the target of the mission.

3. Proposed Algorithm

In this study, the input of the problem is probability map and the target point of each UAV, and the objective of the problem is to develop an optimal path for navigation of UAV from its current point (as start point) to the target point without any conflict with other obstacles or other UAVs considering the cost function of each UAV. Then, in the proposed UAVs path-planning algorithm, the used cost function will be first described, and then the given cost function will be optimized by parallel genetic algorithm for which, in turn, an optimal path will be obtained for each UAV. This operation is explained in 2D and 3D states and simulated for a 2D state.

3.1. Cost function

The used cost function in the study has three terms two of which belong to each UAV, and third term belongs to the distance between UAVs. The first two terms are related to the path length and the safety of the path. First, the manner to obtain the cost of each UAV is explained and then total cost is described.

The safety of the path is determined by a logarithmic function that depends on the specified probabilities of the probability map.[1] Since each cell has a longitude and latitude coordinate in the map and also a specific probability, the probability map is defined by map function in (6).

$$map(x, y) = P(\bar{S}) \tag{6}$$

Figure 2 shows an example of probability map from a given environment. The colors indicate that whatever cells are concentrated from light blue to darker red, the probability of danger is increased, i.e. probability of non-neutralizing is increased. Here, six different levels of danger are represented and shown by six colors and every color has a value between zero and one. Also, the obstacles are represented by dark color. As example: $map(10, 18) = p_4$

The danger function for each cell is determined by (7). The cost function for a path is equal to the summation of the danger function of cells that the UAV has passed. The cost function is defined by (8).

$$\text{danger} = -\log(map(x, y)) \tag{7}$$

$$\text{cost}_{\text{safe}} = \sum_{k=1}^{s} (-\log(map(x_k, y_k)) + c) \tag{8}$$

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

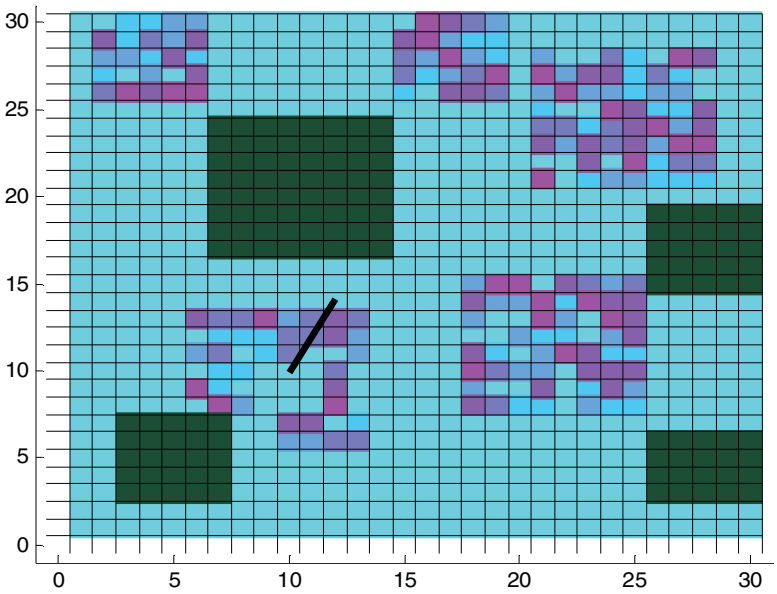


Fig. 2. The probability map of an assumed environment.

where *(x,y)* is the longitude and latitude coordinate of the cell, c is constant, and s is the number of cells that the path passes through them. For example, for the path shown in Fig. 2, because the path passes though the cells with the positions (10, 10), (10, 11), (11, 11), (11, 12), (11, 13), (12, 13), (12, 14), the cost function related to the safety of the path is $\text{cost}_{\text{safe}} = -(3\log(p_1) + 3\log(p_4) + \log(p_5))$.

3.1.1. Footprint of a line. To obtain a footprint of a line and realize cells that line passes through, we should have two points on it. In this work, each part of the path is a straight line and both start point and end point are clear. If $r_i = (x_i, y_i, z_i)$ and $r_f = (x_f, y_f, z_f)$ are the coordinates of initial and final points, respectively, then the unit vector of the path is according to (9):

$$uv = \frac{r_f - r_i}{\sqrt{\left(r_f - r_i\right)^2}} = \frac{(x_f - x_i)\hat{i} + (y_f - y_i)\hat{j} + (z_f - z_i)\hat{k}}{\sqrt{(x_f - x_i)^2 + (y_f - y_i)^2 + (z_f - z_i)^2}} \tag{9}$$

If the unit vector (9) is used, the cells that one line passes through can be obtained from (10)

$$r_j = round\left(r_{j-1} + \delta l \times unique_vector\right) \tag{10}$$

δl has a small amount and indicate move step. Equation (10) starts in $r_1 = r_i$ and continues until $r_j \geq r_f$. The flowchart of the described method is shown in Fig. 3. Number of repetitive r_i represents the number of steps that have been taken through the cell. Therefore, if the number of iterations r_j is equal to K, then the path length passed through the cell will be (11).

$$L = k \times \delta l \tag{11}$$

According to defined equations, the improved cost function is according to (12).

$$\text{cost} = \sum_{k=1}^{s} \left((-\log\left(map(x_k, y_k)\right) + c)\, L_{(x_k, y_k)} + wL_{(x_k, y_k)}\right) \tag{12}$$

Where w is the weight of path length and is used to adjust the effect of length and safety of the route to the desired value. $L(x_k, y_k)$ is the passed length in each cell. Here, to consider the obstacles, it is assumed that map function is very small and approximately zero when there is an obstacle cell.

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

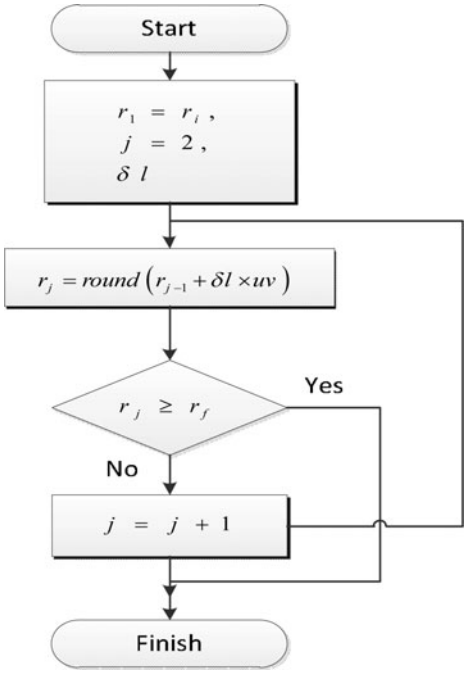


Fig. 3. Algorithm flowchart of footprint a line.

In other words, the probability of non-neutralizing UAV is very small in these cells. So, if there is an obstacle cell in the path, it will high costs and is eliminated automatically.

The manner to obtain the cost of each UAV is explained. Since an evolution algorithm (genetic algorithm) is used for the optimization in a parallel form, the total cost function depends on the minimum cost of each UAV until algorithm is present at that stage. *The default answers at each of iterations are equal to the paths that have minimum cost for each UAV.* Then, the paths have to be investigated in motion. As the velocity of UAVs is clear, the time of their arrival to each point is obtained through their direction. Now, the time of simultaneous passage of two UAVs from a specified range is considered.

For this, a constant C is defined as the minimum distance of two UAVs to avoid collision. *dc* is a $num \times 1$ matrix where *num* is number of UAVs, and *j*th array of *dc* is number of collisions *j*th UAV with other UAVs.

According to the topics the total cost function in each step is defined as 13.

$$\text{COST} = \sum_{i=1}^{num} \left(\text{cost}_i + e^{dc_i}\right) \tag{13}$$

3.2. Parallel genetic algorithm

3.2.1. Encoding. The used genetic algorithm performs optimization for all UAVs, simultaneously and in a parallel form. The defined codes are multidimensional, i.e. a dimension of code is developed for each UAV. Every code is defined as a $m \times n$ matrix where m is the number of UAVs and n is the length of the code that indicates a path. Each cell has a longitude and latitude in a 2D mode; therefore, two indices are needed to develop a path. *row* index is considered to be the longitudinal position of the path cells and *col* index as latitudinal position of the path cells. So, every chromosome or individual in population has two indices, *row* and *col*. Figure 4 shows an example of encoding for three UAVs implemented in Fig. 5.

GA needs an initial population. First, some chromosomes are randomly generated as the initial population, and then the first and last bits of *row*s and *col*s are considered to be the same for each UAV, because the first and last bits of the start and target cells related to each UAV are similar.

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

$$\text{row} = \begin{bmatrix} 1 & 2 & 5 & 10 & 13 & 16 & 19 & 21 \\ 10 & 10 & 14 & 16 & 20 & 23 & 26 & 30 \\ 3 & 4 & 5 & 8 & 16 & 18 & 24 & 26 \end{bmatrix}$$

$$\text{col} = \begin{bmatrix} 1 & 7 & 14 & 16 & 18 & 20 & 20 & 30 \\ 8 & 13 & 13 & 18 & 22 & 26 & 29 & 30 \\ 25 & 20 & 18 & 16 & 16 & 15 & 12 & 10 \end{bmatrix}$$

Fig. 4. An example of encoding for three UAVs.

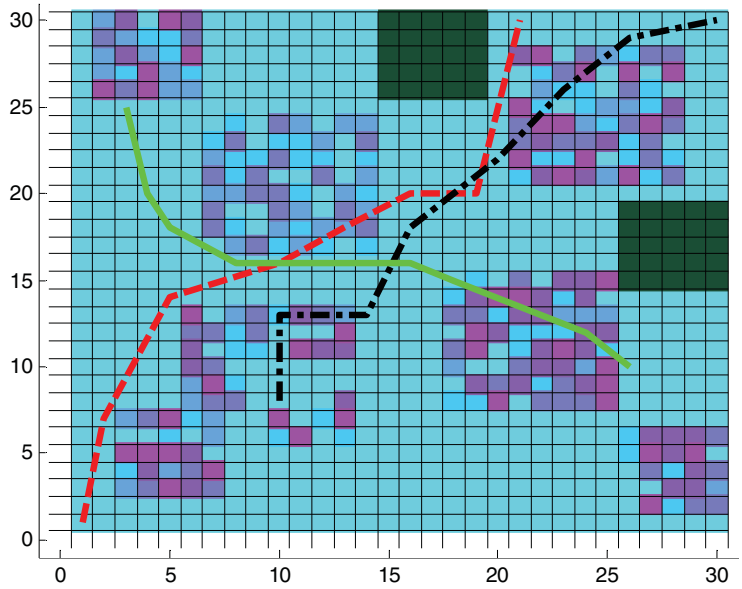


Fig. 5. The implementation of code of Fig. 4 in probability map.

3.2.2. Selection operator. The present chapter aims to select the appropriate chromosomes considering their cost. The more appropriate a chromosome, The higher the probability of its selection. To select pair chromosomes, some chromosomes with low cost are first selected. Assuming l chromosomes are selected, the number of repetitions of each chromosome corresponds to (14). Because minimum costs are important, new function, namely *newcost* is defined according to (15).

$$repeat_i = \frac{\text{newcost}_i \times 50l}{\sum_{j=1}^{l} \text{newcost}_j} \tag{14}$$

$$\text{newcost}_i = \frac{1}{\text{cost}_i} \tag{15}$$

Now, in a virtual house with $50l$ rooms, each chromosome is duplicated to the number of its own *repeat* and occupies the rooms. Assuming that the number of crossovers equal to n, and then $2n$ rooms are selected randomly. Thus, $2n$ chromosomes in the rooms are selected for crossover operation. It should be noted that the factor of "*50*" which raises the number of choices and thus enhances the accuracy of selection of chromosomes is used, and the factor is considered as an arbitrary number. For example, it is assumed that $n = 2$, $l = 4$ and Table I shows the cost and the indices of each chromosome.

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

Table I. An example of repeat function.

i (chromosome index)	1	2	3	4
Cost	134	43.2	486	219
newcost	0.0075	0.0231	0.0013	0.0046
repeat	41	127	7	25

Table II. Selected pair chromosomes.

Selected pair chromosome		
i_1	2	2
i_2	4	1

random selection	1	⋯	41	42	⋯	168	169	⋯	175	176	⋯	200
						⇓	⇓					
i	1	1	1	2	2	2	3	3	3	4	4	4

Fig. 6. Selection of pair chromosomes.

$$row\{1\}:\begin{Bmatrix}1&2&5&10&13&16&19&21\\1&6&7&7&10&15&21&21\end{Bmatrix}\xrightarrow[r=2]{\text{CO}}\begin{Bmatrix}1&2&7&7&10&15&21&21\\1&6&5&10&13&16&19&21\end{Bmatrix}$$

$$col\{1\}:=\begin{Bmatrix}1&6&10&11&13&20&26&30\\1&9&12&17&16&23&27&30\end{Bmatrix}\xrightarrow[r=2]{\text{CO}}\begin{Bmatrix}1&6&12&17&16&23&27&30\\1&9&10&11&13&20&26&30\end{Bmatrix}$$

$$row\{2\}:=\begin{Bmatrix}10&11&8&11&14&24&25&30\\10&10&18&14&15&17&19&30\end{Bmatrix}\xrightarrow[r=4]{\text{CO}}\begin{Bmatrix}10&11&8&11&15&17&19&30\\10&10&18&14&14&24&25&30\end{Bmatrix}$$

$$col\{2\}:=\begin{Bmatrix}8&17&13&16&16&29&29&30\\8&9&10&21&20&20&27&30\end{Bmatrix}\xrightarrow[r=4]{\text{CO}}\begin{Bmatrix}8&17&13&16&20&20&27&30\\8&9&10&21&16&29&29&30\end{Bmatrix}$$

$$col\{3\}:=\begin{Bmatrix}3&4&5&8&16&18&24&26\\3&7&9&10&16&17&22&26\end{Bmatrix}\xrightarrow[r=5]{\text{CO}}\begin{Bmatrix}3&4&5&8&16&17&22&16\\3&7&9&10&16&18&24&26\end{Bmatrix}$$

$$col\{3\}:=\begin{Bmatrix}25&20&18&16&16&15&12&10\\15&21&19&17&13&12&10&10\end{Bmatrix}\xrightarrow[r=5]{\text{CO}}\begin{Bmatrix}25&20&18&16&16&12&10&10\\25&21&19&17&13&15&12&10\end{Bmatrix}$$

Fig. 7. An example of crossover operator for three UAVs.

Now, according to Fig. 6, considering the number of the *repeat* functions of the chromosomes, they devote some rooms to themselves. 4 (or $2n$) are selected among the rooms. Finally, according to Table II, chromosomes associated with i_1 and i_2 do crossover together.

3.2.3. Crossover operator. The pairs of chromosomes (paths) as selected by the selection operator in the previous step are combined by the single-point crossover operator in this part, and two new chromosomes are produced. In single-point crossover, a point is chosen randomly between 1 to n (n is the number of chromosomes bits) and then two selected chromosomes exchange the point of their genes. This work is done for both *row* and *col* as the same. Figure 7 shows an example of single-point crossover.

3.2.4. Mutation operator. After the crossover, the mutation occurs on some of the new chromosomes. When mutation occurs, one bit of the *row* and *col* for each UAV is randomly selected and its content randomly changed. Figure 8 shows an example of mutation.

After the operators are applied on the chromosomes, the new chromosomes are needed to be evaluated and the cost of each one is calculated. Then, the previous steps must be repeated. Figure 9 shows how to calculate total cost, and Fig. 10 represents the flowchart of algorithm mentioned. The

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

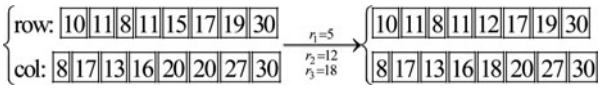


Fig. 8. An example of mutation.

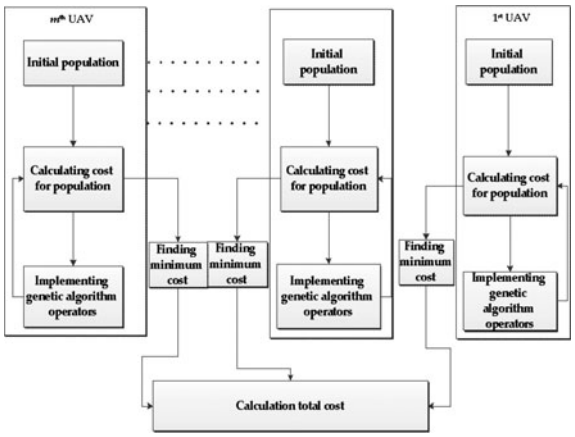


Fig. 9. Calculating total cost.

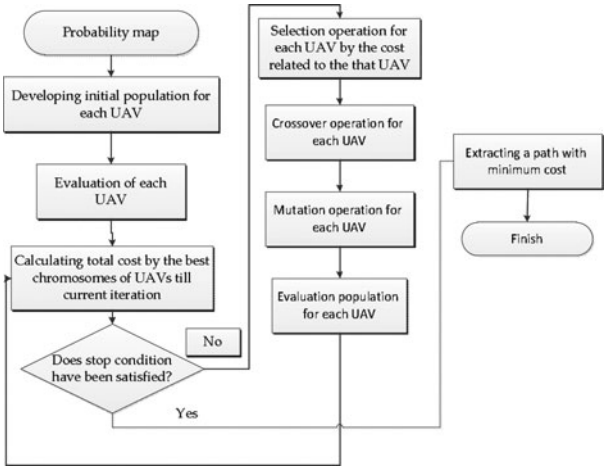


Fig. 10. The used algorithm flowchart for simultaneously group path planning of UAVs.

described algorithm is applied to a two-dimensional motion. To transform the above algorithm into a three-dimensional algorithm, it is sufficient to add another index into encoding part, so height could be taken into account. We add an index called *height*. In three-dimensional motion, danger areas are similar to two-dimensional areas, with the difference that such areas have also height.

In this work, the height of all danger areas is considered equal to each other. Since aerial vehicles fly usually in fixed heights and alter their height only when an obstacle appears in the way, the obstacles have their own special and real height.

3.3. Real-time algorithm for one UAV

As the running of the algorithm may be a bit longer, the algorithm is somehow different for the real time problems. The stages of real-time path planning are presented as follows:

1. Path planning from the start point to the targeted one, to achieve a local optimal path without any collision at the first moments of flight and the movement of UAV at the direction of the path planned.
2. The movement of UAV as much as t second is planned at the direction of the path, so as the velocity of the coordinate is specified, it is possible to obtain UAV.

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

Table III. Design parameters for second simulation.

Marker and color in figure	Velocity (m/sec)	Status	Target position	Start position	UAV
• (red)	70	w = 0	(10,30)	(10,8)	1
◇ (black)	90	w = 3	(27,10)	(3,25)	2
□ (white)	110	w = 3	(22,30)	(1,1)	3

Table IV. Design parameters for third simulation.

Marker and color in figure	Velocity (m/sec)	Status	Target position	Start position	UAV
• (red)	70	w = 0	(30,30)	(10,8)	1
◇ (black)	90	w = 0	(27,10)	(3,25)	2
□ (white)	110	w = 0	(22,30)	(1,1)	3

Table V. Design parameters for forth simulation path.

Marker and color in figure	Velocity (m/sec)	Status	Target position	Position	UAV
• (red)	80	w = 3	(10,30)	(10,8)	1
◇ (black)	90	w = 3	(30,30)	(30,1)	2
□ (white)	120	w = 3	(22,30)	(1,1)	3
⋆ (dark cyan)	85	w = 3	(30,30)	(10,27)	4

3. Afterwards, the coordinate of t second is regarded as a new starting point and then it is run to find a new path based on this algorithm.
4. If prior to the end of t second the algorithm is stopped, the path obtained is the optimal path and UAV must pass the reminder of the way in this path.
5. If, by ending t second, the algorithm is not stopped, the last optimal path obtained is considered as the final path in the memory of planning, and algorithm is stopped, otherwise it returns to stage 2.

Such stages are continued until the algorithm has the sufficient time to run; otherwise it is continued until the targeted point.

4. Simulation

A 30× 30 environment is considered for simulation. The environment is composed of six levels of dangerous areas and obstacles that are distributed randomly. First a problem is solved for one UAV by the cost function (12). Then, the simulations are performed for three and four UAVs. According to the algorithm, each UAV is optimized by its own cost function and using (13) the paths without collision are obtained. To show the elapsed time in the obtained path of each UAV, the small markers on the path are used. Each marker represents one second elapsed. The target points of UAVs for three different cooperative simulations are represented in Tables III–V, respectively.

Figures 11 and 12 show an optimal two- and three-dimension path for one UAV using (12) with $w = 4$. In this problem, the height obstacles are clear and the path passes on the obstacles.

Figure 13 shows the optimized paths for three UAVs which have been parameterized in Table III. According to Fig. 13, in the cells with coordinates (15, 10), (10, 5), (15, 24) and (18, 21) two UAVs have passed through a cell, and their distance is relatively reduced around those points; however it does not indicate that there is a collision, because the time of the arrival of each UAV to the mentioned points is varied. As can be seen from the figure, every small marker on the paths represents a second. As an example, pink UAV passes through the point (15, 10) at 20th second, while white UAV passes through the point at the 16th second. These times indicate that a contact does not occur. This explanation also holds true for other locations.

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

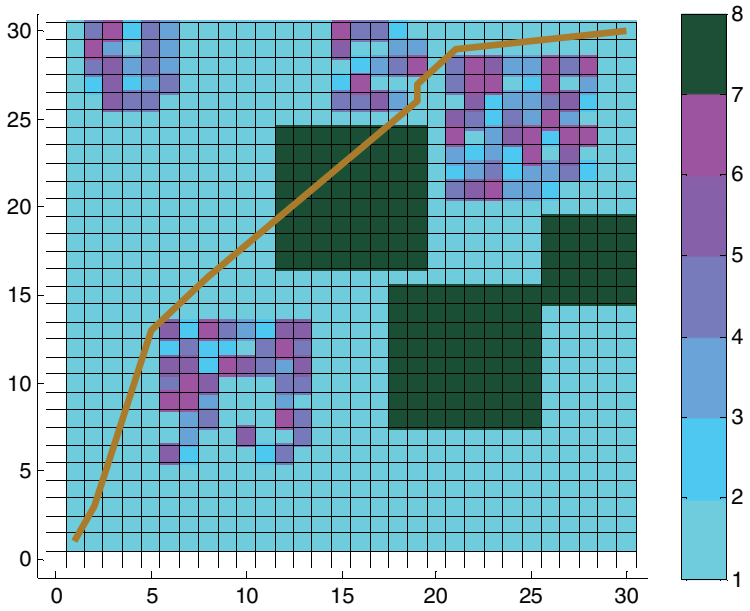


Fig. 11. Optimal path in surface.

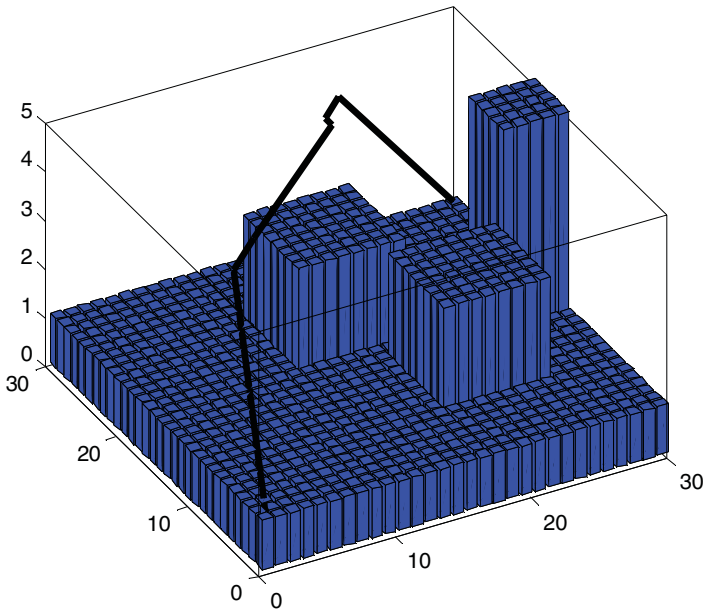


Fig. 12. Optimal path in space.

Figure 14 represents the optimized paths for three UAVs with the path safety term ($w = 0$) considered. As shown in the figure, the obtained path for all UAVs is completely safe and it has not entered the danger zones as much as possible. However, not only the path of each UAV but also their speed must be taken into account. The points on which the paths intersect each other have not happened at the same time at all.

In the fourth simulation for all UAVs, $w = 3$ is assumed. Figure 15 presents the designed paths for four UAVs. Here, the obstacles are increased relative to the previous problem, and UAVs avoiding collision arrive the target point. Like previous simulations, the same parts of paths occur at different times.

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

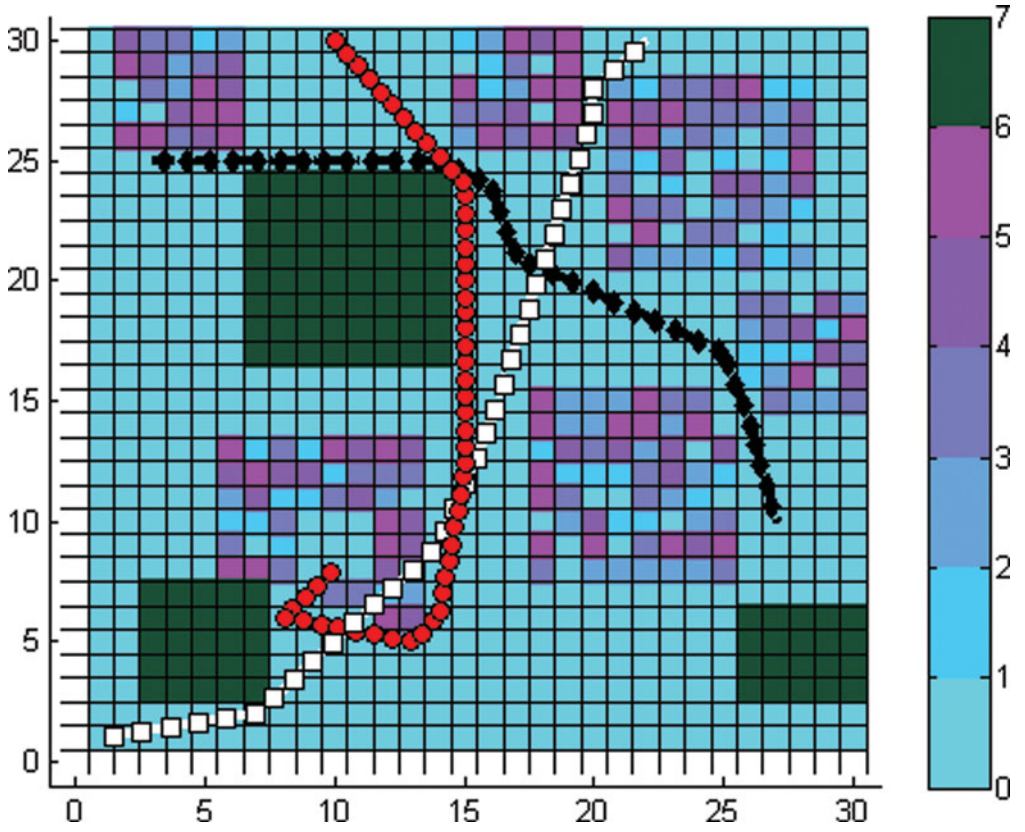


Fig. 13. Obtained paths for three UAVs (second simulation).

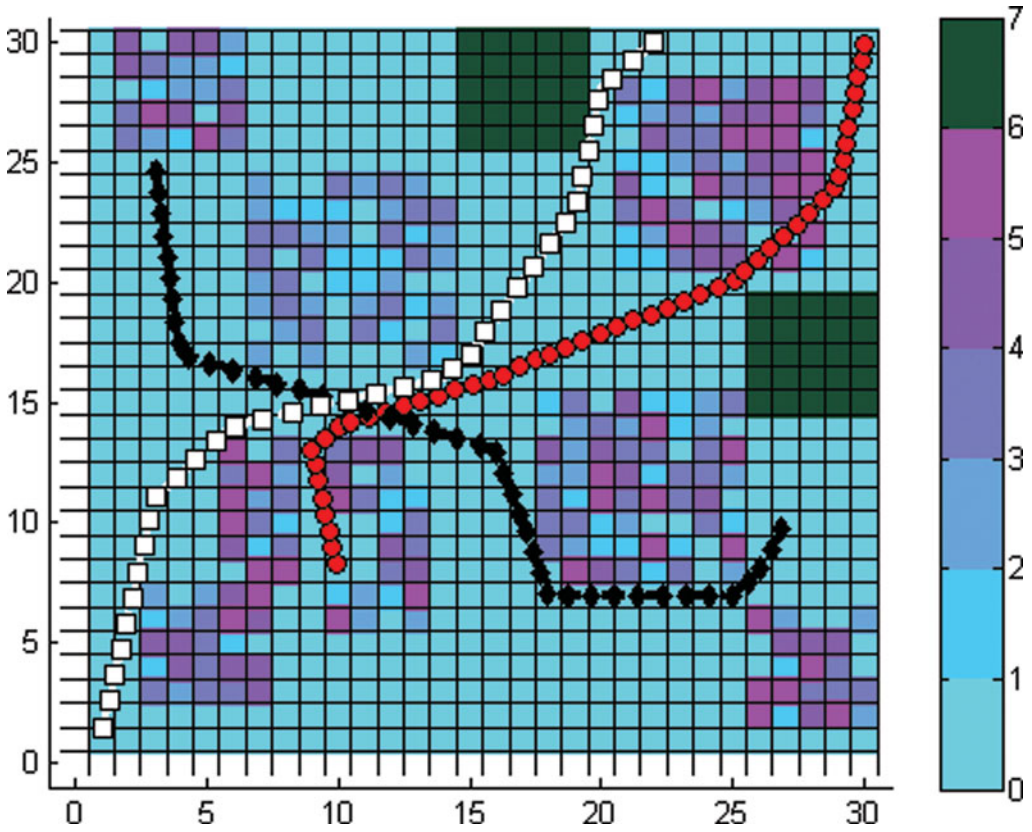


Fig. 14. Obtained paths for three UAVs (third simulation).

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

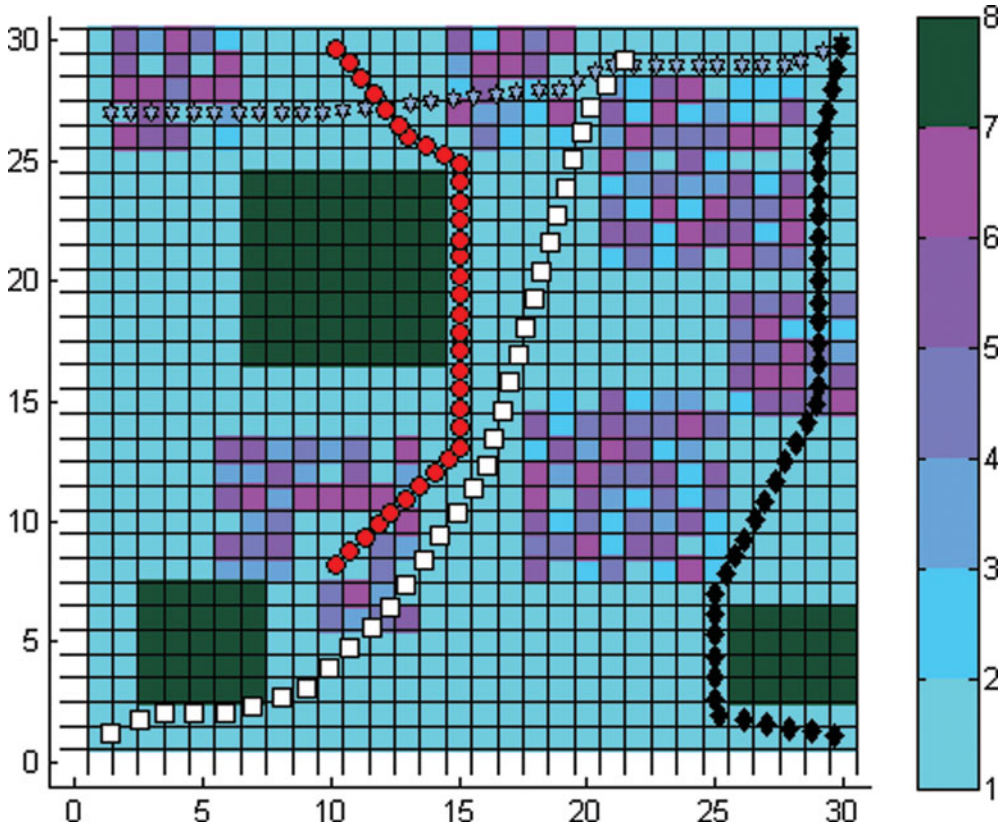


Fig. 15. Obtained paths for three UAVs (forth simulation).

5. Conclusion

In this paper, a simple but applicable method was presented using PGA for UAVs group path planning. Firstly, how to generate the probability map using definition of three events was explained, and after generating the probability map the cost function was expressed. The cost function is a linear combination of the probability of non-neutralizing UAVs, path length and an exponential term to consider their distance. Genetic algorithm used in this work, whether the path planning is two or three dimensional, it offers proportional two or three-dimensional codes and each direction have its own special code.

In this study, four different simulations were performed in which the quantity of w, probability map, environment obstacles and the number of UAVs were considered as variables. As it is clear from the tables and maps, whatever value of w is higher the shortness of the path is prioritized over its safety. In the first simulation a three-dimensional path planning with $w = 4$ is performed. The second problem obtains the path to three UAVs with different cost functions. This simulation shows the high flexibility of the algorithm for different cost functions and proves that there is no obligation to the same cost function of UAVs. In Fig. 14 in which the w is zero for each three UAVs, UAVs come close to the danger celles to reduce the path length; however they do pass the danger zone. Figure 15 is related to the path planning in the case that the effect of path length is considered three times of the path safety. Therefore, given the high significance of the path length, UAV has passed the less dangerous areas but it has not entered the cells with obstacle. This indicates that the applied algorithm is highly capable of solving different path planning problems of UAVs with a variety of conditions.

References

1. M. Jun and R. D'Andrea, "Path planning for unmanned aerial vehicles in uncertain and adversarial environments," *Cooperative Control: Models, Applications and Algorithms* (2002) pp. 95–111.
2. Y. Kim, *et al.*, "Real-time path planning with limited information for autonomous unmanned air vehicles," *Automatica* **44**, 696–712 (2008).

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press

3. R. G. Brian, *et al.*, "Optimal Path Planning of UAVs Using Direct Collocation with Nonlinear Programming," *American Institute of Aeronautics and Astronautics* (2006).
4. M. Shanmugavel, *et al.*, "Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs," *Control Engineering Practice* **18**, 1084–1092 (2010).
5. H. Zhonghua, *et al.*, "Cooperative attack path planning for unmanned air vehicles swarm based on grid model and bi-level programming," *Information & Computational Science* **8**(4), 671–679 (2011).
6. H. Tong, "Path planning of UAV based on voronoi diagram and DPSO," *Procedia Engineering* **29**, 4198–4203 (2012).
7. H. Qu and K. Xing, "Takacs Alexander, An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing* **120**, 509–517 (Nov. 23, 2013).
8. E. Melia, M. Malvezzia, S. Papinia, L. Pugia, M. Rinchia and A. Rindia, "A railway vehicle multibody model for real-time applications," *Int. J. Veh. Mech. Mobility* **46**(12), 1083–1105 (2008).
9. S. Falomi, M. Malvezzi, E. Meli and A. Rindi, "Determination of wheel–rail contact points: Comparison between classical and neural network based procedures," *Meccanica* **44**(6), 661–686 (Dec. 2009).
10. L. Babel, "Flight path planning for unmanned aerial vehicles with landmark-based visual navigation," *Robot. Auton. Syst.* **62**(2), 142–150 (Feb. 2014).
11. M. A. Andreev, A. B. Miller, B. M. Miller and K. V. Stepanyan, "Path planning for unmanned aerial vehicle under complicated conditions and hazards," *J. Comput. Syst. Sci. Int.* **51**(2), 328–338 (2012).
12. T. Shiiba, K. Obana and N. Machida, "Comparison of linearized vs. non-linearized multibody vehicle model for real-time simulation," *Int. J. Non-Linear Mech.* **53**, 32–40 (Jul. 2013).
13. A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Trans. Robot. Autom.* **3**, 249–265 (1987).
14. J. P. Hespanha, H. H. Kizilocak and Y. S. Ateskan, "Probabilistic Map Building for Aircraft-Tracking Radars," *American Control Conference*, Arlington, VA (2001) pp. 4381–4386.

https://doi.org/10.1017/S0263574714001878 Published online by Cambridge University Press