Search and Rescue Simulation in Avalanche

Dian Yuan
Technical University of Munich
Mechatronics and Robotics
Zhejiang, China
dian.yuan@tum.de

Yinglei Song
Technical University of Munich
Mechatronics and Robotics
Zhejiang, China
ge97zoj@mytum.de

Yang Xu
Technical University of Munich
Mechatronics and Robotics
Shanghai, China
ge23ged@mytum.de

Jianyu Tang
Technical University of Munich
Robotics, Cognition, Intelligence
Guangdong, China
ge28kop@tum.de

Yubo Min
Technical University of Munich
Mechatronics and Robotics
Jilin, China
ge23bah@mytum.de

Liang Ma
Technical University of Munich
Mechatronics and Robotics
Henan, China
ge23miq@tum.de

***Abstract*—Unmanned aerial vehicle (UAV), i.e. a vehicle that does not require real-time manual control, has a role to play in many situations. In this report, we try to use ROS and Unity to simulate a search and rescue mission using a UAV in an avalanche scenario. We build our sensor model, ways of mission planning and will generate a trajectory for UAV. Our goal is that our UAV can find all victims under very limited known conditions and to identify their coordinates as precisely as possible.**

I. INTRODUCTION

Avalanches kill and injure hundreds of people every year. As victims of avalanches are often buried in snow and cannot be seen by the naked eye during search and rescue, many skiers choose to carry signal transmitters or wear ski clothing with built-in signal transmitting chips to protect themselves.

To work with the search and rescue teams to locate the victims as quickly as possible, we tried to use a UAV that had been programmed to carry out a search mission. The ultimate goal was to find and return the coordinates of all the victims as quickly as possible.

In section 2, preliminaries will introduce how are the victims be generated and our sensor model is set concerning reality. Afterward, section 3 presents the ways of mission planning, which includes the order to detect all the victims and the method of determining precise coordinates for each victim. Section 4 is about trajectory generation and location reports. Section 5 is the conclusion.

II. PRELIMINARIES

Our overall design idea is that each victim sends their id number and a signal of a specific strength to the outside world. This signal is expressed as a numerical value that decreases as the distance to the UAV increases, and if it goes beyond the range in which the signal can be communicated, the UAV will not receive any signal from that victim. Also, to simulate a realistic environment, we have added noise to this signal that fits a Gaussian distribution, thus preventing us from directly inverting an accurate result from a known signal attenuation model.

A. Generation of Random Victims

We define two attributes for each victim, one is their id number and one is their coordinates. Where the id number is what will be sent to us along with the signal strength value, the id is a positive integer starting from 1. The id is set to make the search more orderly and we can distinguish which victims have been found by their id. Their coordinates are generated randomly and are also not known to the UAV. The pseudo-code for generating a victim can be found in Alg. 1: As the scene provided is too large, we have selected suitable upper and lower bounds on the X, Y, and Z axes as the range of random points, for X it is [-110,70], for Y it is [0,70], for Z it is [1,10]. Function of $frand$ is to generate a random number within this interval given an upper and lower bound.

Algorithm 1 Victim Generation

Input: X_{low}, X_{up}, Y_{low}, Y_{up}, Z_{low}, Z_{up}, id number n
Output: A victim with its id and coordinate v

1: $v.id \leftarrow n$
2: $x \leftarrow frand(X_{low}, X_{up})$
3: $y \leftarrow frand(Y_{low}, Y_{up})$
4: $z \leftarrow frand(Z_{low}, Z_{up})$
5: $v.coordinate \leftarrow [x, y, z]$
6: return v

B. Sensor Model

For the sensor model between victims and the UAV, We have taken into account the parameters of relevant products on the market. The finalized parameters are as follows:

- **Capability**: Our transmitters have an effective signal transmission range of 60m, beyond which the UAV will not detect the victim's message. At the same time, we consider a radius of 0.1m, with the sender as the center of the circle, to be the specific area where the victim is located, i.e. the signal strength is maximum in this area. 0.1m also corresponds to the maximum permissible deviation from the actual coordinates of the last transmission.

- **Transmitting Model**: The essence of the transmission model is to enable the UAV to pick up larger signal values in areas close to the victim, and only smaller values in areas far from the victim. Therefore we have chosen the corresponding non-logarithmic model directly because it is easy to compute:

$$PR = c_0 * PT/r^n \tag{1}$$

Where PT is the transmitted power of the signal, i.e. the maximum value, which we currently set at 10000, PR is the power that the UAV will receive, c_0 is a constant related to the antenna parameters and the transmitting frequency, n is the propagation factor and for simplicity we both take 1 for them. And r is the euclidean distance of the UAV from that victim.
- **Noise**: To increase the uncertainty of the model, all signals received by the UAV have a noise that fits a Gaussian distribution with a mean value of zero. We also specify that the variance of this noise increases as the distance between the UAV and the victim increases.

It should be noted that although when calculating the Euclidean distance between the UAV and the victim, we use the coordinate points of the victim, that is because under our definition this attenuation model is objective. If the UAV is within transmitter range and such a value which is consistent with that model, must be detected. However, the UAV will not know the exact coordinates of the victim at first.

The whole sensor model can again be summarised as follows: each victim carries a transmitter that continuously sends its id to the outside world, and the UAV receives the victim's id, and the corresponding current signal strength, as soon as it enters the signal range of that victim. What we do next is to get the coordinates of this victim while knowing only these two pieces of information.

III. Mission Planning

A. Task Description

Having identified the sensor model above, the problems we think that the UAV may meet are as following:

- **Detect all the Victims**: UAV will have a workspace as we will not use the full provided scenario. The victims are placed randomly in this workspace and the UAV does not know how many victims there would be in the workspace. We need to make sure it can find all of them.
- **How to locate**: If a victim is to be found, since we only know the strength of the signal emanating from him, even if we can deduce the approximate distance between the UAV and the victim at this point, we still do not know the exact direction. We need an algorithm to help us determine the victim's position and eventually calculate its coordinates.
- **Order to find all the Victims**: The process of finding and locating victims also requires a certain amount of planning, and we hope that we will be able to find all the victims as quickly as possible using relatively little time and traveling relatively short distances.

B. Detection

Since the number of victims is not known in advance, we have to make it possible for UAV to scan the entire workspace as a way of ensuring that no victim is missed.

We have set several observation points. If a ball of radius 60m is drawn with each observation point as the centre, the space contained in these balls should add up to include all the workspace.

The general idea is that the UAV will go to each observation point in sequence, find all the victims that can be observed at the current observation point, then fly to the next observation point and repeat the process. In order to cope with the situation where a victim can be observed by more than one observation point at the same time, we stipulate that once the victim has been observed for the first time and its coordinates determined, all subsequent observation points will ignore the information it sends.

C. Location

Once we reach an observation point, we receive information once about the victims within the detection range of that observation point. At this point, it comes down to the question of how we derive the exact coordinates of the victim from the known conditions, i.e. the strength of the received signal and the sensor model.

We propose an algorithm that mixes the idea of gradient descent and the four-point localisation method:

The idea of gradient descent is used to find the most likely direction from the UAV to the victim. Each time this gradient-descent-like algorithm is run, the UAV will fly a certain distance forward, backward, left, and right at the current position, measuring the respective signal strengths at specific locations in four different directions. Since the four directions are two opposite each other, we should be able to get two directions that make the signal strength increase. We plan to take the weighted average direction of these two directions as the expected flight direction of the UAV, which can be described as:

$$D = (d_1 * R_1 + d_2 * R_2)/(R_1 + R_2) \tag{2}$$

Where D is our predicted fly direction in vector, d_1 and d_2 are two directions at which the signal strength increases. R_1 and R_2 are the signal gains in their directions. UAV will fly in this predicted direction. If the UAV flies within a spatial distance of 20m from the victim, then it will automatically enter the four-point localization algorithm and start solving the victim's coordinates. At the same time we set up an error checking mechanism: for every 100 cycles, the planning algorithm compares the signal strength before and after one cycle. If the signal strength becomes lower, we abort the flight in the current direction and again use the method described above to determine a new flight direction until we are within 22m of the victim and can switch to the four-point localization algorithm.

Four-point localization method is best known for its use in satellite positioning, where it can be applied to solve equations to find a more accurate solution than the approximate solution when using gradient descent. There are several methods for solving this quadratic equation, and the one we ended up using is referred to in this article [1]. Our method of picking points to list equations is then we design a specific non-congruent trajectory, and once we enter the 20m range, we will have the UAV fly this trajectory once, sample it and solve it.

In practice, according to our definition, because the variance of the noise becomes relatively small within 22m of the victim, the gains from using the four-point localization method to directly solve the equation for more accurate coordinates are greater at this point. Whereas within 22m to 60m of the victim, the variance of the noise is relatively large, and we would use a gradient descent-like approach to get the UAV into a range of distances where points can be taken and solved first. However, since there is still in fact some error within 22m, and since some points actually beyond 22m may also be mistakenly considered to be within 22m because of the noise, we take two measures to try to avoid being affected: One is to use mean filtering to eliminate as much as possible Gaussian error with a mean of 0. The other is that the UAV does not shift the method at 22 m from the victim, but at around 20 m to ensure that the UAV is within the specified range before sampling and solving the equations.

D. Planning Order

We want the process of finding all victims to be as short as possible and with as few travel paths as possible. Our planning order can be shown as below 1:

The general criteria for the search can be summarised as identifying the close ones first (those with high signal strength).

As we have already mentioned, we will select the right number of observation points based on the defined workspace. Each observation point corresponds to an area and we will rescue the victims in these areas in a set order. And when searching an area, we introduce a $searchlist$ to hold the victims whose coordinates are to be determined. This list will only decrease and not increase when performing an area search and is used to avoid detecting victims that should belong to the next observation area when we set up sub-observations.

And the reason we introduced the concept of sub-observation points is that if we had used the process of finding one victim and then returning to the observation point to find the next victim, this process would have lost a lot of valuable time on the way. For example, if the current observation point is (0,0,0) and the three victim coordinates A, B and C, are (0,0,40), (0,0,30) and (0,0,-35). Without the concept of sub-observation point, we will search them as A-C-B, the UAV has to do two tosses and turns. But now our search order will be A-B-C, saving time with only one folding run.

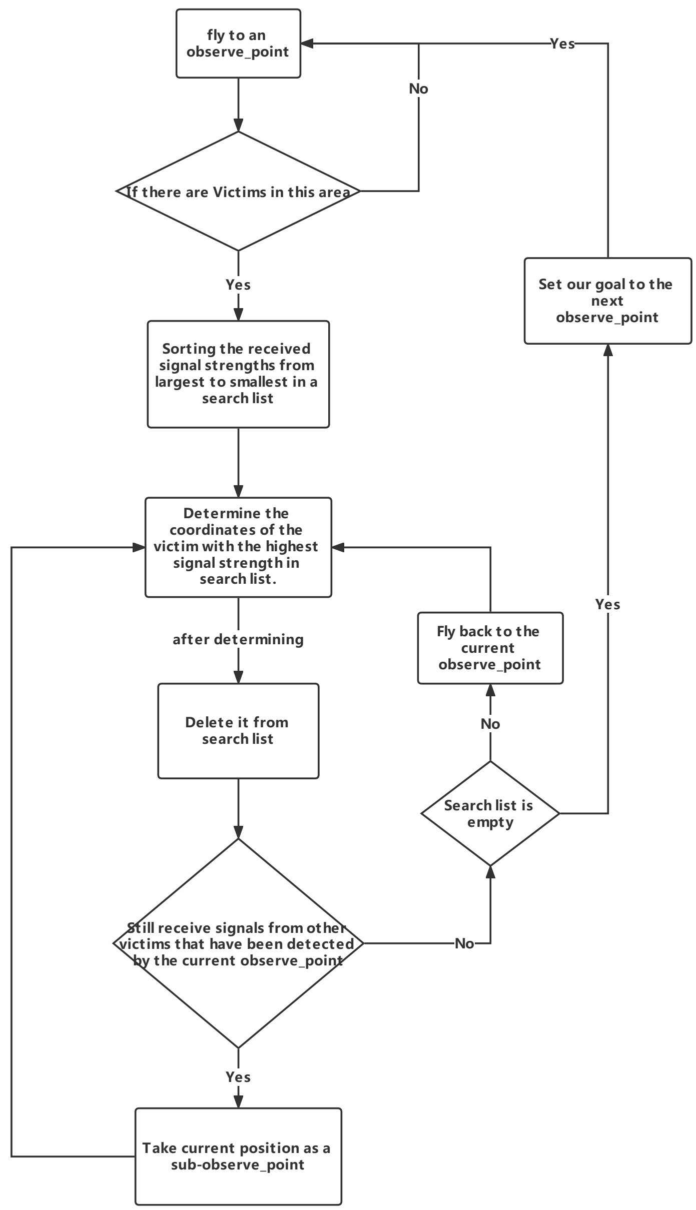

Fig. 1. Process diagram to show our planning order

We hope that if a particular victim is found, we will be able to drop by and carry out a rescue if there are other victims in his vicinity. Of course, if we can't detect any other victims in the vicinity of a particular victim, we will fly directly to the next observation point if the search list has been emptied. However, if the search list has not been emptied, this means that the two victims may be in opposite directions, so we need to go back to the current observation point and start searching from a new direction.

IV. Trajectory Generation and Location Report

Having done the above, we used rviz to show the generated trajectories, and the victims we added to the workspace. We experimented with the case when the victims were 3 or 4 and the generated trajectories are shown below 2 3.

And as we have mentioned, the drone will fly at a certain altitude to avoid hitting the mountain and will not land on the same plane as the victim. The few places where the trajectory entangles are the sampling steps when using the four-point localization method. A comparison of the final solved points with the actual generated points can be seen in the following tables IV IV.

Where we can see, for the 3-victims case, the point with the largest error from the actual coordinates of the victim is 0.5877m. For the 4-victims case, the point with the largest error from the actual coordinates of the victim is 0.7149m.

These errors appear to us to be within acceptable limits. Since avalanche signal transmitters are usually worn around a person's waist, if we take the average height of a person to be around 1.8m, it is acceptable for subsequent arriving rescue teams to dig through a 1m to 2m square of snow after the UAV has determined the location of the signal.

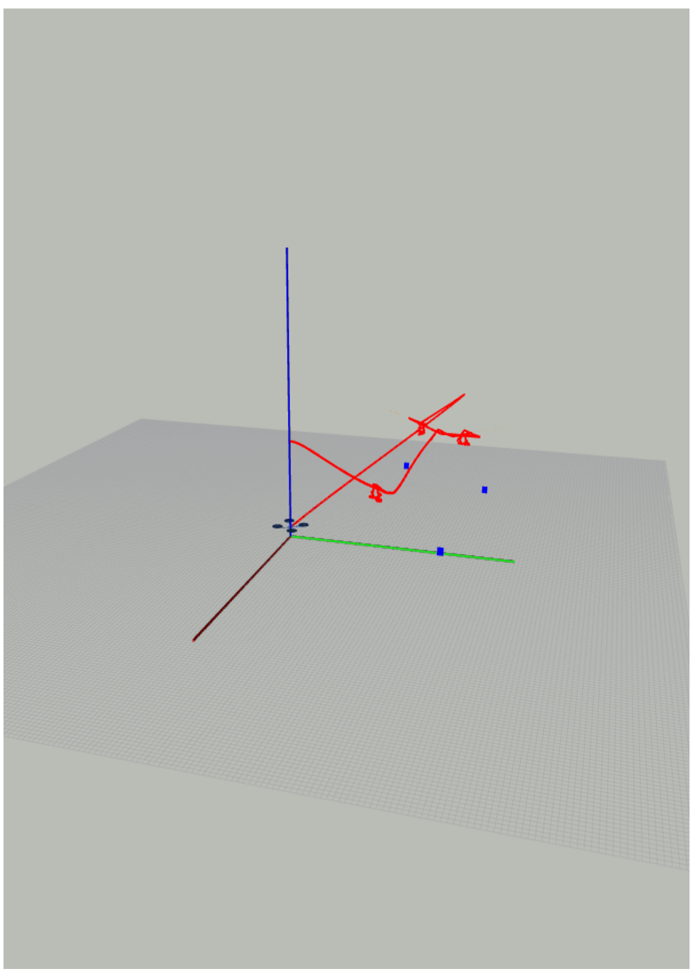

Fig. 2. Trajectory to find 3 victims

The 3-Victims Case	
Real	Solved
[23.4512,45.2121,8.1212]	[23.5572,44.8477,8.3695]
[-33.2533,47.4554,5.2454]	[-32.8024,47.3205,5.5973]
[-43.1237,22.1122,6.7454]	[-43.0746,22.2441,6.8249]

The 4-Victims Case	
Real	Solved
[-2.2984,6.9649,2.2997]	[-1.7875,6.6212,2.6363]
[-12.7364,40.792,2.9575]	[-12.4068,41.1838,3.2172]
[-30.949,27.0487,4.6028]	[-30.3996,27.1815,5.0406]
[46.7243,10.682,9.8419]	[46.3415,10.4582,9.7225]

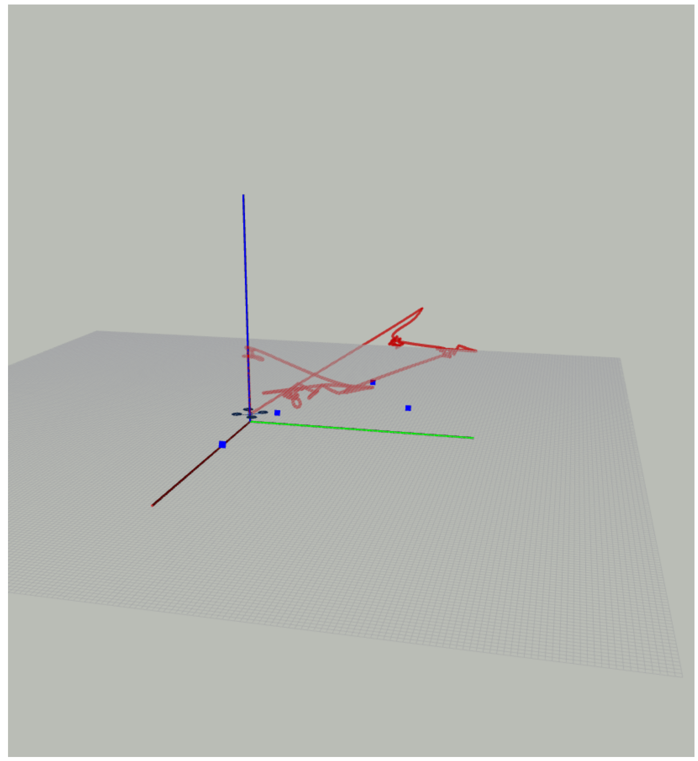

Fig. 3. Trajectory to find 4 victims

V. Conclusion

After testing, our algorithm has a certain degree of feasibility, and with very little information received, we have been able to get the coordinates of all the victims more accurately. We were also able to obtain the coordinates of the victims by solving them without having to fly to their location, which was satisfactory in terms of search efficiency.

The downside is that in the actual writing of the algorithm, our logic may still have imperfections and the code implementation is more difficult. Moreover, we have set some conditions that do not necessarily follow the reality exactly to facilitate the writing of the algorithm, and we have not taken into account the interference of inter-victim signals at all. These are all areas to be optimized in the future.

References

[1] Y. Xu, J. Zhou, and P. Zhang, "Rss-based source localization when path-loss model parameters are unknown," *IEEE communications letters*, vol. 18, no. 6, pp. 1055–1058, 2014.