Autonomous Systems WS21/22 Final Project
——Search and Rescue

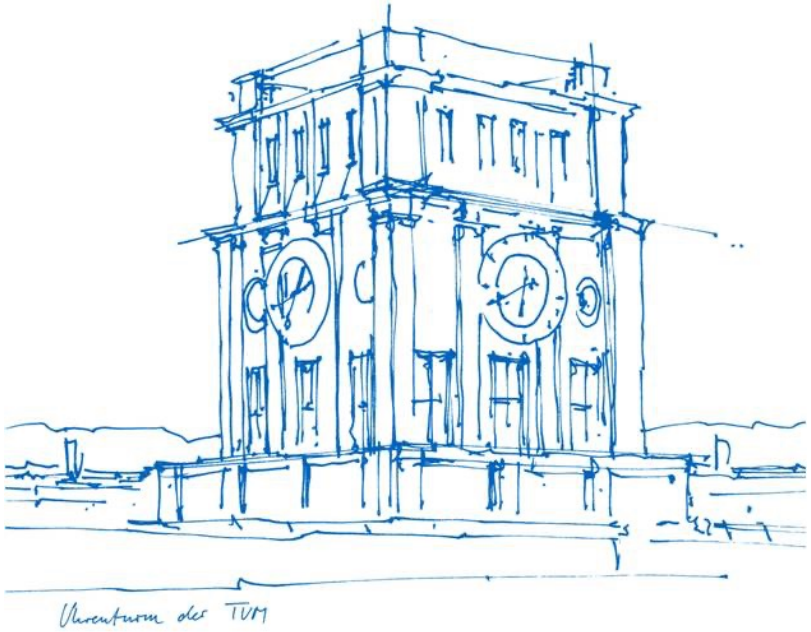
Group Terminus

Dian Yuan, Yinglei Song

Yang Xu, Jianyu Tang

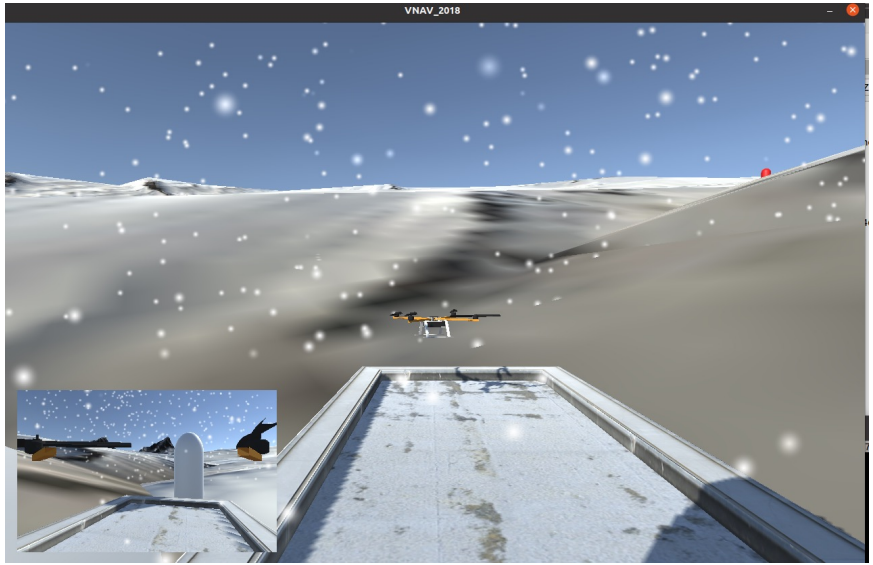
Yubo Min ,Liang Ma

24.03.2022



Introduction

Avalanche Scenario



Victims are randomly separated in the different places.

Number of victims is unknown.

Goal : Fly over the avalanche
Locate all the victims in a short time & distance
Errors of coordinate x, y, z are within ± 1

Challenges: unknown environment
control of UAV's dynamics

Victims are distributed in: $x \in [-110, 70], y \in [0, 70], z \in [1, 10]$

Software Stack

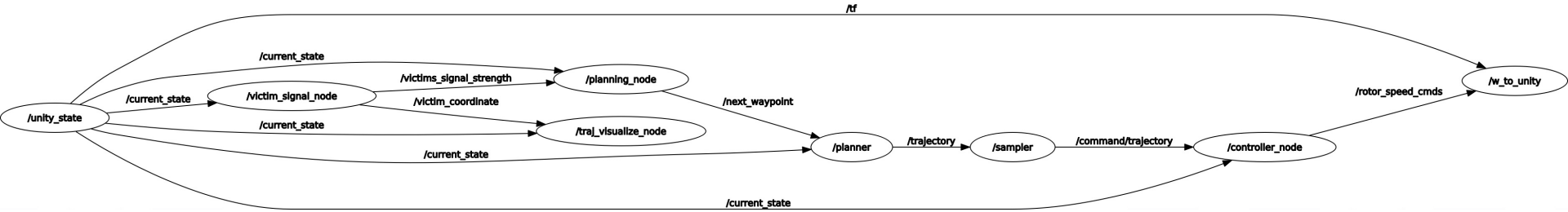
Victims Signal Generation Package

Planning Package

Trajectory Generation Package

Controller Package

Trajectory Visualization Package



Sensor Model

Victims are equipped with signal transmitters.

Signal strength is known, coordinates of victims are not. Strength is depend on the distance between victims and UAV.

Non-logarithmic Model:

$$PR = c_0 * PT/d^n$$

Max Strength $PT = 10000$
Constant $c_0 = 1$
Propagation Factor $n = 1$
Euclidean Distance d

To avoid infinite signal strength when $\mathrm{d} \to 0$, we actually use $PR = 10000/(d + 0.9)$

Sensor Model

Real Sensor with Noise

Noise is Gaussian distributed.

Noise depend on the distance d.

Mean of noise is zero.

When d$\leq 22\mathrm{m}$, variance is relative low. $\Longrightarrow$ Suitable to use signal strength for Calculation.

When d$\geq 22\mathrm{m}$, variance is relative high. $\Longrightarrow$ Unsuitable to use signal strength for Calculation.

Planning Algorithm

To avoid obstacles, the UAV flies at an altitude of 20 meters.

Challenge: Chicken and Egg Problem

We need signal strength of next point for planning before we go.
We need to reach next point before we get the actual signal strength.

An intuitive method: move like a dumb robot vacuum cleaner

Predefined trajectory, which lets UAV reach within 22m of all the victims. Then UAV starts calculation.

Drawback: long distance of travelling

Planning Algorithm —— Observation Points

An Optimal Method:

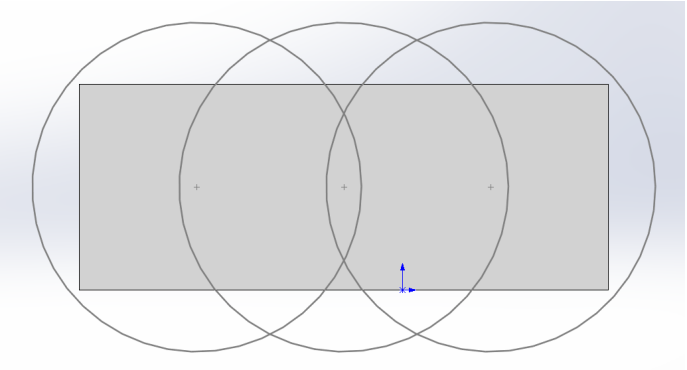
Due to the signal's characteristics, we separate planning task into three parts.

$d \geq 60m$: (We define some threshold value of signal strength to classify three cases.)

Predefined observation points

When UAV flies to these points, it can stay within 60m of victims.

On observation points, UAV may receive signals from several victims.
It chooses the one with strongest signal. $\Longrightarrow$ Shortest distance



Planning Algorithm —— “Gradient Descent”

$22m \leq d < 60m$:

An inspiration from gradient descent in deep learning:

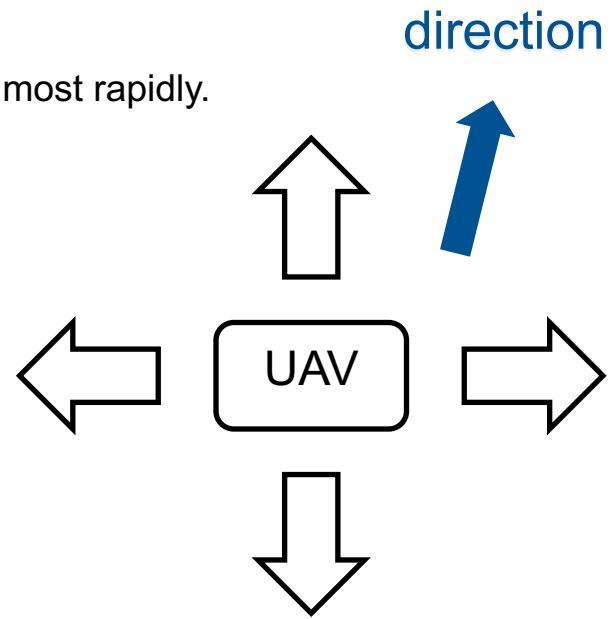
Always choose the direction which makes signal strength increase most rapidly.

How to find the direction?

UAV moves forwards, backwards, leftwards and rightwards.

Collect the signal strength at corresponding points.

Compare signal strength, derive the direction.



Planning Algorithm —— “Gradient Descent”

UAV keeps going on this direction. Meanwhile, it also collects signal of the targeted victim.

When signal strength is decreasing, UAV stops.

Then it moves forwards, backwards, leftwards and rightwards again to find the new direction.

This loop lasts until the signal strength shows that UAV is within 22m of the victim.

For the accuracy of collected data within 22m, we actually use 20m instead of 22m.

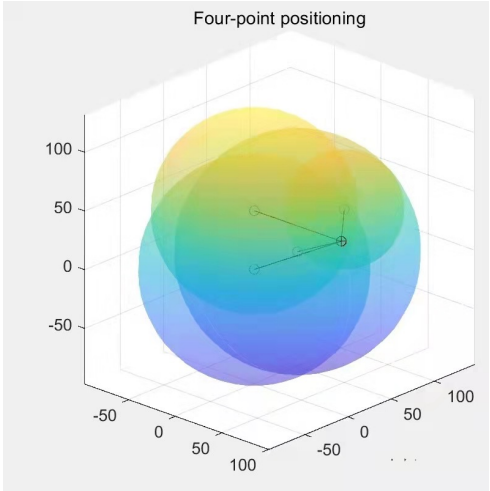
Planning Algorithm —— Four-Points Localization

$d < 22m$:

Signal strength is accurate enough for localization.

Requirement:

These points should not lie on the same line.
Calculation with more than four points is also possible.



$\Longrightarrow$ UAV moves to some locations with predefined displacements compared to current position.

After collecting the data, we use sensor model ($d = \frac{10000}{PR} - 0.9$) to estimate the distance.

Then we calculate the coordinate of targeted victim.

Planning Algorithm —— Four-Points Localization

$$
\begin{aligned}
(x_1-x)^2 + (y_1-y)^2 + (z_1-z)^2 &= d_1^2 \\
(x_2-x)^2 + (y_2-y)^2 + (z_2-z)^2 &= d_2^2 \\
(x_3-x)^2 + (y_3-y)^2 + (z_3-z)^2 &= d_3^2 \\
&\cdots\cdots \\
(x_m-x)^2 + (y_m-y)^2 + (z_m-z)^2 &= d_m^2
\end{aligned}
$$

$$
\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} -2x_1 & -2y_1 & -2z_1 & 1 \\ -2x_2 & -2y_2 & -2z_2 & 1 \\ -2x_3 & -2y_3 & -2z_3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -2x_m & -2y_m & -2z_m & 1 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ d \end{bmatrix} - \begin{bmatrix} d_1^2 - (x_1^2 + y_1^2 + z_1^2) \\ d_2^2 - (x_2^2 + y_2^2 + z_2^2) \\ d_3^2 - (x_3^2 + y_3^2 + z_3^2) \\ \vdots \\ d_m^2 - (x_m^2 + y_m^2 + z_m^2) \end{bmatrix}
$$

$$
\boldsymbol{V}_v = \boldsymbol{B}_v \boldsymbol{X}_v - \boldsymbol{l}_v \qquad \boldsymbol{X}_v = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ d \end{bmatrix} = (\boldsymbol{B}_v^T \boldsymbol{B}_v)^{-1} \boldsymbol{B}_v^T \boldsymbol{l}_v
$$

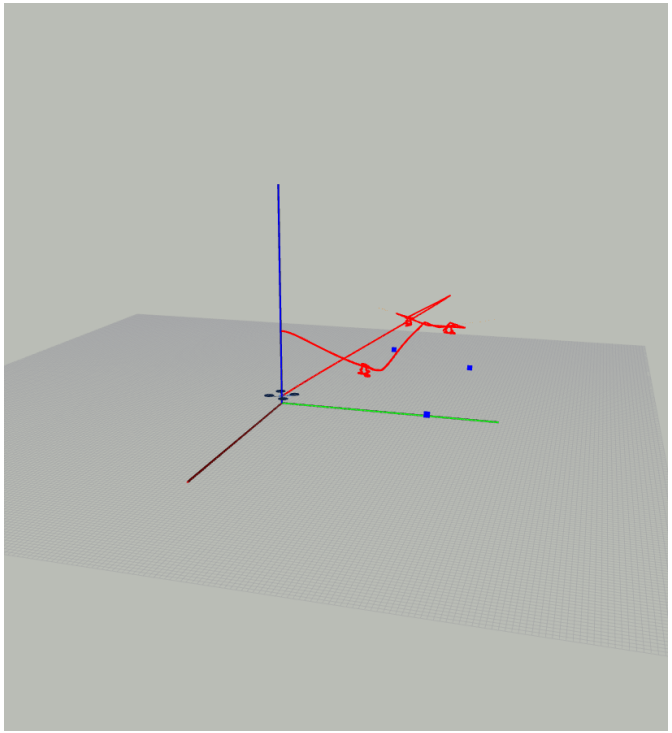
Trajectory Generation & Control

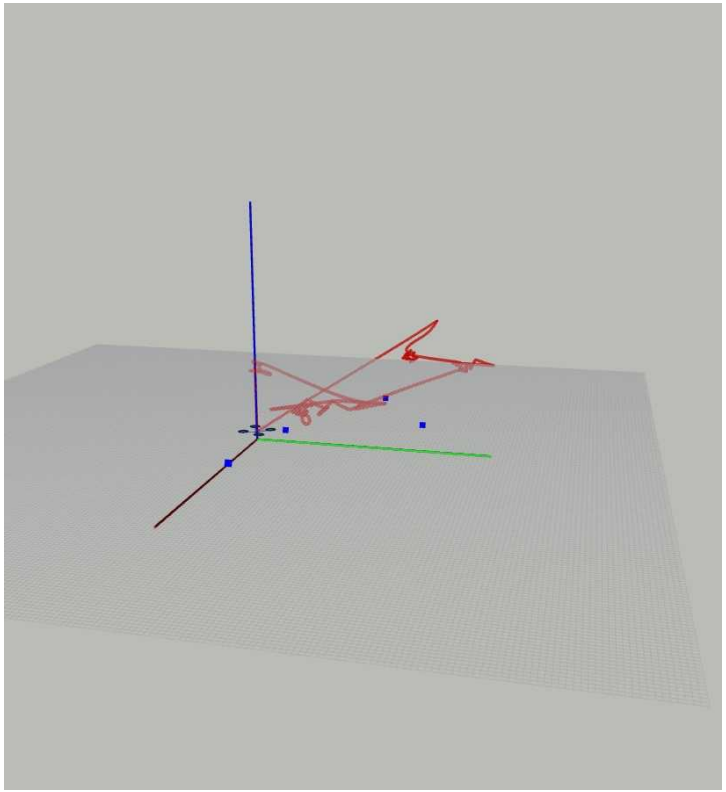
Adapted from code in Lab4 and Lab6.

- No more need of .yaml file $\Longrightarrow$ waypoints are published by planning_node
- Geometric Controller
- For the need of collecting data, the velocity at waypoints is defined as $\vec{0}$.

Trajectory Visualization

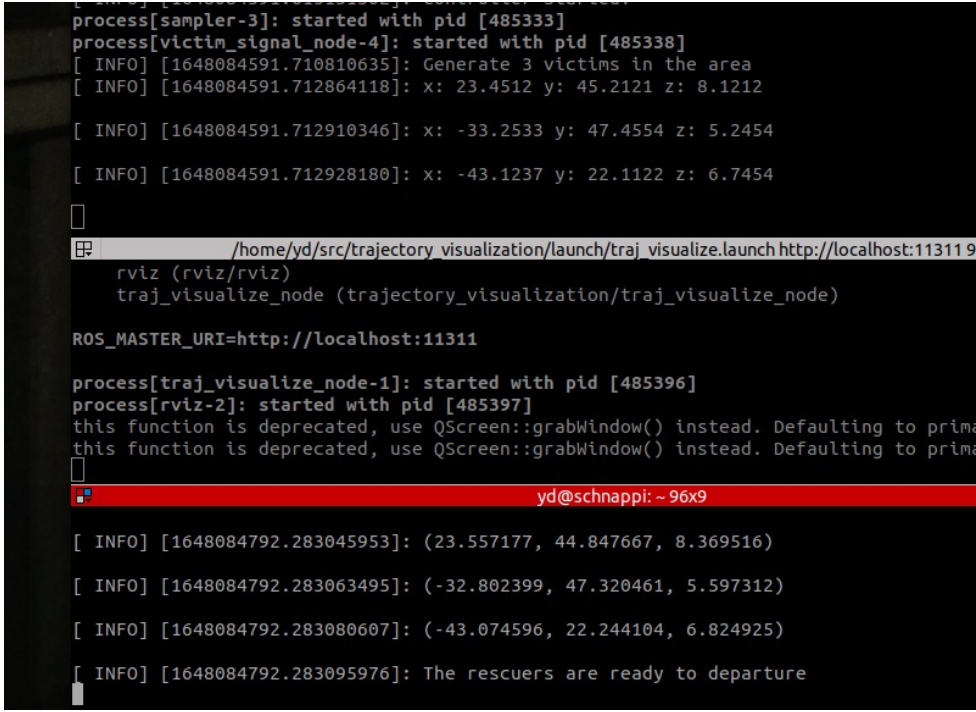
Show the victims and trajectory of UAV in RVIZ





Location Report

Actual	x	y	z
Victim1	23.4512	45.2121	8.1212
Victim2	-33.2533	47.4554	5.2454
Victim3	-43.1237	22.1122	6.7454
Measured	x	y	z
Victim1	23.5572	44.8477	8.3695
Victim2	-32.8024	47.3205	5.5973
Victim3	-43.0746	22.2441	6.8250



Conclusion

Advantages:

After a large number of tests, we insured that the errors of coordinate x, y, z are within ± 1.

Solve the problem with real sensor model.

Relative short search time and distance.

Things still can be improved:

Logic maybe imperfect, we hope to achieve automatic obstacle avoidance

A steadier UAV pose.

Some conditions do not follow the reality.

Thanks for your listening and welcome for questions.