# Autonomous Flight of Micro Air Vehicles 2015 - Group 6

M. Bevernaegie,[*] C. Cheung,[†] Y.I. Jenie,[‡] S.H. Lee,[§] P. Lu,[¶] M. Siddiquee,[‖]
Delft University of Technology, Delft, The Netherlands

## Abstract

**This should be done at the end**

## 1 Introduction

1. Overall introduction of the project

2. Equipment - Opti-track, AR.Drone 2.0

3. Software - Paparazzi

## 2 Vision algorithm

### 2.1 *Vision-based Navigation in the Literature*

Here we talk about the vision algorithm used in the literature

### 2.2 *Own Vision Algorithm*

Here we talk about our own method and the reasons for the choice

## 3 Flight plan

### 3.1 *Flight Plan I - Moving Waypoints*

Seong's flight plan

### 3.2 *Flight Plan II - Fixed Waypoints*

Peng's flight plan

### 3.3 *Comparison and Choice of Flight plan*

Comparison and discussion.

## 4 Simulation

### 4.1 *Linear Simulation*

Yazdi's method and results

### 4.2 *MATLAB & Paparazzi Simulation*

Seong's method and results

## 5 Competition Result and Discussion

After tuning the control and vision threshold parameters through trial-and-error in real tests, it was confirmed that both vision algorithm and flight plan works very well with adequate sensitivity towards the distance from the orange poles and black wall. Also, it was observed that the drone could maintain a stable flight during the maneuvers, such as stopping and turning of the heading. The slow speed of the drone, together with the intermittent turning strategy, allowed the drone to stop and turn with only a marginal overshoot whenever an obstacle is detected or it goes outside the flight area. Regrettably, however, the tuning was done in the environment where the obstacles were rather spaced out widely, and in particular, the black wall was positioned near and parallel to one of the boundaries of the flight area.

During the actual competition, the black wall was positioned adjacently near one of the corners, and there was another orange pole next to it. What happened in the first half of the competition was: the drone moved and made a few turns, and eventually went to near the corner. After it realized that it was out of bounds, it made $180^o$ turn and attempted to come back inside. However it saw the adjacent black wall, and turned to the other side. Then it saw the orange wall and turned back to the previous heading. And it saw the black wall again, and so on. After drifting slowly towards the black wall, it touched the black wall and had to be restarted due to the low battery.

In the second half of the competition, the drone was started after being repositioned in the home position, so it did not go into the corner and did not get trapped again. As a result, the drone covered the traveled distance of well more than 100 m. Although this is not quite a high result compared to the other teams, our drone had the fewest collisions with the obstacle, and our team won the 6th place out of 13 teams in total.

In the following list, the main reasons are addressed as to why it could not be foreseen that the drone may get trapped near the boundaries/corners:

- In the Paparazzi simulation, the vision algorithm was ”mimicked” by a random number generator. Hence, there was never a possibility of a drone getting trapped in a place.

- During the real tests for tuning, the black wall was located parallel to one of the boundaries and further way

[*] M.Bevernaegie@student.tudelft.nl
[†] C.Cheung@student.tudelft.nl
[‡] Y.I.Jenie-1@tudelft.nl
[§] S.H.Lee-2@student.tudelft.nl
[¶] P.Lu-1@tudelft.nl
[‖] M.Siddiquee-1@student.tudelft.nl

from the corner. So, the drone could not be possibly trapped by the ”blocking” of the wall.

To solve this problem in the future, the following strategies may be used to improve the flight plan:

- ”Breakthrough Maneuver”: Incorporate a maneuver that when a drone makes more than two consecutive turns back and forth, turn the heading only $45^o$ and fly forward for a certain period of time while suppressing command from the vision result.

- ”Temporary Blinding”: In case the displacement of the drone is too low during a fixed time period, flexibly adjust the vision threshold parameters (e.g. the width threshold for the pole detection) such that the drone becomes less sensitive towards the obstacles nearby.

- ”Circular Flight Area”: Define the flight area as a circle, instead of a square. This will prevent the drone from going too close into a corner.

## 6 Conclusion

## References

## Appendix A: Data

## Appendix B: More data