NCSU CentMesh Drones Challenges 2014 – First Round

Preliminaries

General Rules: An rectangular area on the Centennial Oval has been designated as the "flying zone", as detailed in the February tutorial (video available from CentMesh wiki) and also shown in the figure below. A small area adjacent to it is the "staging zone", also shown in the figure. The participants may be in the staging area at any time as long as they wear hardhats and eye protection. Only the judges and other designated organizing team members are allowed to be inside the flying zone at any time. Members of the organizing team will be posted around the flying area to warn off passers-by, and also provide coordination signals (such as raising "flight in progress" flags) that must be honored by all challenge participants.

MCP: In the following, we refer to the "Mission Control Protocol" or MCP – this is a protocol that we have designed specifically for "Mission Control" to send command or coordination messages to the CentMesh drones. The protocol is specified on the CentMesh wiki together with other challenge materials, and sample stub code is also provided.

Geo-fence: The flying zone will be programmed into the drone, and will extend from ground upto an altitude of 400 ft. If the drone attempts to leave this volume of space under programmatic control, a "Geo-fence" will be triggered, and the drone will automatically execute a Return-To-Launch (RTL).

Violations: There are multiple ways in which you can fail any of the challenges. Some of them are considered violations, and will instantly cause the trial to be aborted, with an override recall (either automatically or through judges' intervention). For some violations, we may also disqualify you from re-trying. Violations are:

1. Team members encroaching on the flying zone, or otherwise failing to honor instructions of challenge judges or organizers. (May forfeit retries)
2. Drone under programmatic control attempts to leave flying zone and triggers geo-fence
3. Drone under programmatic control descends and hits ground faster than the pre-programmed "landing descent rate". (May forfeit retries)

Stepping Stones

The following are suggested as easier building blocks that can potentially be incorporated into the challenges later. We provide these simply as suggestions for steps that teams might want to follow; we will not test for these and you need not submit a program to accomplish these tasks.

Stepping Stone: From a landed position, listen for a MCP "Start_Mission" message. Once this message is received, fly vertically up to an altitude of 10 meters, and stationkeep for 2 minutes. Then fly vertically down and land. Vertical speeds should not exceed 2 meters/second.

Stepping Stone: From a landed position, listen for a MCP "Start_Mission" message. Once this message is received, fly vertically up to an altitude of 10 meters, and stationkeep. Your application should listen for a MCP "LAND" message. Once the message is received, fly vertically down and land. Vertical speeds should not exceed 2 meters/second.

Stepping Stone: From a landed position, listen for a MCP "Start_Mission" message. Once this message is received, fly vertically up to an altitude of 10 meters, and stationkeep. Your application should listen for a MCP "WPList" message, containing a sequence of coordinates, which you should fly to in order, holding position for 5 seconds at each. You must fly only vertically or horizontally at any given time. Once at the last coordinate, return to initial stationkeeping position, then land vertically.

Stepping Stone: Same as previous, but you must avoid designated volumes of space during flight – these volumes will be in the shape of rectangular blocks. These will

be communicated to you via a MCP "Obstacles_List" message sent *before* the "WPList" message.

First Round Challenges

Challenge 1: 3D Traveling Salesman. From a landed position, listen for a MCP "Start_Mission" message. Once this message is received, fly vertically up to an altitude of 10 meters, and stationkeep. Your application should listen for a MCP "WP_List" message that will contain a sequence of coordinates. Your application should compute a trajectory that visits the designated points in any order, attempting to minimize the quantity {total horizontal distance traveled + 4 * total vertical distance traveled}. You need not hit the absolute minimum to pass the challenge, but should make a decent attempt. Once at the last coordinate, and assuming you have received a "End_Mission" MCP message by this time, execute a "Return to Launch" (i.e. climb vertically to ceiling, travel horizontally to launch point, land vertically). (Time limit: 5 minutes)

Challenge 2: Catch Me If You Can. From a landed position, listen for a MCP "Start_Mission" message. Once this message is received, fly vertically up to an altitude of 10 meters, and stationkeep. Your application should then listen for a MCP "WP_On" message. Fly to the location included in that message. From that time onwards, while you are within 5 meters of the specified location, you will accumulate 1 point per 100 millisecond. *If you violate the pre-set maximum velocity on the autopilot, you will be judged to have lost the challenge.* Continue to listen for an MCP "WP_Off" message. At this time, the last-specified waypoint has "dried up", and you are no longer accumulating any points by staying in the location. After some time, you will receive another MCP "WP_On" message, and can start accumulating points by going to that location, until the next "WP_Off" message. After a series of such alternative "WP_On" and "WP_Off" messages, you will receive an MCP "End_Mission" message, upon which you must execute a "Return to Launch".

The successive coordinates are guaranteed to be the vertices of a regular polygon, in polygon order (i.e. visiting them in that sequence would result in drawing the polygon, not any of the diagonals). The polygon is guaranteed to be planar, *but the plane is not necessarily horizontal.* The successive MCP messages are not guaranteed to be sent after equal intervals. Remember, *the polygon may be convex or concave.* The polygon degree will not be larger than 8.

Obviously, there is the opportunity of accumulating more points if you can *predict* the next polygon vertex. A correct prediction allows you to use the time between a "WP_Off" message and the succeeding "WP_On" message by making progress toward the next waypoint location in anticipation, thus you start accumulating points faster after the "WP_On" message is issued. An incorrect prediction, on the other hand, can make things worse by leading you to travel in the wrong direction. Even a partial prediction can make a significant difference to the number of points you earn.

If you complete the challenge without violating any conditions, your total points will determine if you passed the challenge. The threshold for passing will be announced after the first few trials. (Time limit: dynamic)

Challenge 3: Running the Maze. From a landed position, listen for a MCP "Start_Mission" message. Once this message is received, fly vertically up to an altitude of 20 meters, and listen for an MCP "Obstacles_List", followed by an MCP "WPList" message containing two waypoints. Once received, fly to the first of those waypoints, then make your way to the second waypoint and land at *the horizontal coordinates of the second waypoint in the list previously received*, while avoiding the specified obstructions. These volumes represent "obstacles" that must be avoided (some of them may not exist in reality). *The entire vertical space above the landing coordinates is not guaranteed to be "clear".* (Time limit: 10 minutes)