Inter IIT Tech Meet 11.0 | High Prep

Drona Aviation: The Pluto Drone Swarm Challenge

Task 2 Report

Primary Team ID: 17
Secondary Team ID: 34

Contents

1. Drone Tracking using Computer Vision

Aruco Detection

First the video is captured by the camera frame-wise and **aruco marker** is detected. A box is formed around the **aruco marker** upon detection in the initial frame. The corners of the aruco marker detected are initialized as the corners of the bounding box for **CSRT** to track.

Channel and Spatial Reliability Tracker (CSRT)

- **CSRT tracker** was used to track the motion of the drone accurately even when the drone moved with increased velocity
- The input to the tracker was the output obtained through **aruco detector**
- The bounding box is set as the base position, and it follows the drone as it moves with the **aruco marker** attached to it
- **CSRT tracker** works by training a correlation filter with compressed features (HoG and Colornames)
- The filter is used to search the area around the last known position of the object in successive frames
- The tracker stops tracking once the object goes out of the frame

*Position tracking of Drone using CSRT

2. Control

Introduction

- The **class** `Control` implements **PID control** on roll, pitch and throttle values so that the drone can achieve the desired position
- The `positionhold()` function is responsible for making the drone hold its position at any point in the 3D coordinates which is visible in the camera
- The **z coordinates** come from the **real sense camera pipeline** and the **x, y (pixel)** values come from the camera frame and are transformed to the corresponding x, y values in the real-world coordinates using the z values

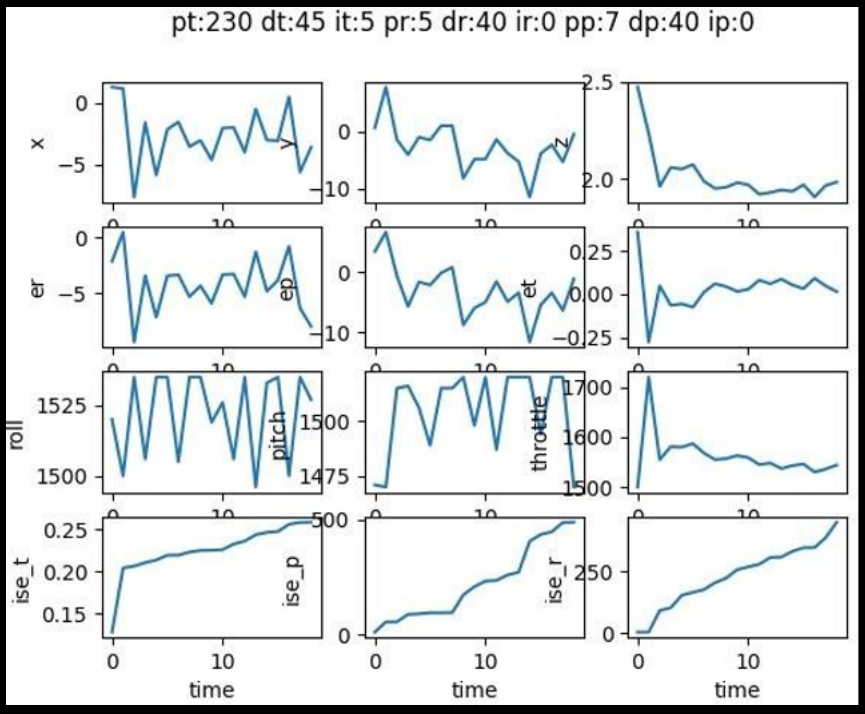
Setting Roll,Pitch and Throttle using PID control

- The **error value** is defined by the difference between the current position and the desired position
- The **ierror** refers to the integrated error and the **derror** refers to the derivative error.
- The **roll, pitch** and **throttle** values are set by summing the reference values with the sum of the errors multiplied by their respective constants
- In the case of throttle, the reference value varies as the drone battery drops to a particular value. Before setting the values, they are limited between two values so that there is less overshooting from the desired position

*Hovering of Pluto Drone using camera

Hovering And Rectangle Formation of Pluto Drone

- In case of **hovering** the drone, a **waypoint** just above the initial position of the drone is passed to the `positionhold()` function and `positionhold()` function is repeatedly called
- In the plot shown, the left column shows **x, roll error, roll** and **integral square error of roll** respectively
- Similarly, the middle column shows **y, pitch error, pitch** and **integral square error of pitch**
- Finally, the last column shows **z, throttle error, throttle** and **integral square error of throttle**



*Variation of drone's parameters and error terms with time

For making a **rectangle,** `positionhold()` is called on four points of the **rectangle** as **waypoints,** and the code is written such that the drone hovers for some time on all four points.

3. Distributed Memory-Caching

- When the tracking and control was done serially, it was observed that there was a time lag due to extra computation time. To reduce this time lag, the detection and control scripts are run **parallelly** using **distributed computing**
- **Memcached** is a general-purpose distributed memory-caching system. The tracking script stores the values of control values **(roll, pitch and throttle)** into a memory cache, and the control script reads these values from the cache in real time
- Initially, **memcache clients** are initialised at a specific localhost port in each of the scripts. Then the values of control values are set in the tracking script using the `set_multi()` function. The control script reads these values using the `get_multi()` function