Multi agent planning to detect obstacles and navigate in an unknown environment

Group Names

***Abstract*— Rescue operations for individuals lost in a forest requi sre drones to quickly scan a large unknown evironment and quickly provide relief. We propose to work on this problem by breaking it into 2 stages. 1) multiple drones cooperatively scan the entire area with unknown obstacles communincating with each other to create a single map. 2) Once a person is found, another drone can be sent from the base to their location to provide relief by finding an optimal path.**

I. INTRODUCTION

Objective:

1) Multiple drones plan and scan the unknown environment. The unknown environment is broken down into multiple grids and the task of the drones is to visit each grid in a cooperative manner.
2) The drones detect obstacles around them which are then transformed to the initial frame and updated on a common map.
3) When a person is found, a different drone using the common map calculates the optimal path to reach their location and provide relief.

Environment:

1) We will use PyBulletDrone environment for this project where cylinders are used to represent trees.
2) The environment spawns these trees at random locations throughout the map which is unknown to the planning algorithm. These obstacles are sensed by the robot as they come within the sensing range of onboard sensors.
3) A person is said to be detected when the drone is close enough to their location.

Procedure:

1) Our workspace is R^3. Configuration space of the quadcopter is $R^3 \times SO(3)$.
2) Start with a vanilla RRT^* algorithm that generates the trajectory in a static environment.
3) Sampling is done from a 12D C-space (Position, Velocity) taking the kinodynamics of the rotor into account.
4) Then make the quadcopter follow this path using an off the shelf control algorithm.
5) Implement different flavours of the RRT algorithm for dynamic environments.
6) Compare these algorithm on the basis of their compute time and the path length.

Questions: Should we search for the optimal path in the 3D World space or the 6D C-Space of the robot or 12D Position + Velocity space?

II. ROBOT MODEL

We have decided to use Quadrotor as our robot.

A. Working of Quadrotors

The quadrotor is a highly non-linear, six degree-of-freedom and under-actuated system. A quadrotor has two sets of counter-rotating propellers, therefore neutralizing the effective aerodynamic drag. It has four principal modes of operation: Vertical movement is controlled by simultaneously increasing or decreasing the thrust of all rotors. Yaw moment is created by proportionally varying the speeds of counter-rotating parts to have movement with respect to quadrotor's z-axis. Roll can be controlled by applying differential thrust forces on opposite rotors of the quadrotor to have movement with respect to quadrotor's x-axis. Pitch can be controlled by applying differential thrust forces on opposite rotors of the quadrotor to have movement with respect to quadrotor's y-axis.

B. Quadrotor Model

1) Model of a rotor: Each rotor rotates with angular velocity ω and generates a lift force F and moment M. Moment is acting opposite to the directing of rotation.

The lift Force F and moment M of ith rotor can be calculated by:

$F_i = k_f * \omega_i^2, \qquad k_f = k_T * \rho * D^4$
$M_i = k_m * \omega_i^2, \qquad k_m = k_Q * \rho * D^5$

where:
k_T is thrust coefficient
k_Q is torque
rho is fluid density
D is diameter of propeller

2) Equations of Motion: Total thrust and moment is the sum of individual ones in each of the 4 rotors.

Thrust: $F = F_1 + F_2 + F_3 + F_4 - mga_3$

Here, F_x are individual lift forces by the propellers and $m * g$ is the one by gravity.

Moment: $M = r_1 * F_1 + r_2 * F_2 + r_3 * F_3 + r_4 * F_4 + M_1 + M_2 + M_3 + M_4$

Here, $r_x * F_x$ are the moments created by forces in quadrotor's centre of gravity and M_x are the individual moments created by the propellers.

3) Newton-Euler Equations for Quadrotor: **Linear Dynamics** Applying Newton's Second Law for system of particles, we get (in inertial frame);

$F = mass * acceleration$

$acceleration(\ddot{r}) = d\dot{r}/dt$, where $\dot{r} = [u, v, w]^T$ (3.3)

Since, w is the yaw-axis in which we calculate thrust, we get;

$$mass{*}\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -m * g \end{bmatrix} + R_{\psi}\phi\theta \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}$$

Rotational Dynamics Applying Euler's rotation equations, we get (in body frame);

$M_c =^A dH_c^B/dt =^B dH_c^B/dt +^A \omega^B \times H_c^B$

where, H_c is the angular momentum and ${}^A\omega^B$ is angular velocity of body B in frame A which is given by $p.b_1 + q.b_2 + r.b_3$

General vector form of Euler's equation is; $M_c = I\dot{\omega} + \omega \times (I\omega)$

For Quadrotor, after rearranging the general vector form;

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Let $\gamma = k_M/k_F$, $M_i = \gamma F_i$, we get;

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & L & 0 & -L \\ -L & 0 & L & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

Final equations using Linear and Rotational dynamics equations, we get;

$$\begin{bmatrix} T \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ 0 & Lk_F & 0 & -Lk_F \\ -Lk_F & 0 & Lk_F & 0 \\ k_M & -k_M & k_M & -k_M \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

III. MOTION PLANNING

1/2-1 page

IV. RESULTS

1 page

V. DISCUSSION

1/2-1 page

REFERENCES

[1] Example, Example bibliography, The bibliography: A great example, 1st ed. vol. 1, Delft, 2019, pp. 1-2