Intro

Unmanned Aerial Vehicles have seen an acceleration in interest for solutions to problems within environments that are cluttered with obstacles. In most cases not all obstacles are known at the initial path-planning stage, or that accurate representation of all known obstacles in memory is not feasible. This constraint gives rise to the need for an online path-planning methodology where the vehicle can sense its environment in and modify the flight path in real time.

For this project we considered the application of autonomous package delivery within an urban environment containing obstacles that are unknown before the mission begins. The vehicle begins at a specified start location, and must navigate the environment to reach the target delivery location without colliding with an obstacle. The overall strategy implemented to achieve this follows. At a specified frequency, the control loop first senses the environment to look for any new obstacles. If an obstacle is found that intersects the current planned path, a new path is computed. Once a path has been chosen, the control inputs to optimally track the path towards the goal are found. The inputs are then applied to the control until the the next period begins.

Environment Sensing: Simulated Sensor

For the purposes of this short project, the sensor was assumed to be very simplistic. A radius of effective sensing from the vehicle's current location is defined in the simulation parameters. If any portion of an obstacle lies within this radius away from the vehicle's position, then the entire obstacle is sensed, and saved to the list of known obstacles. The sensor neglects any visual obstruction from other obstacles. For the simulations performed in this report, the sensing radius was assumed to be 10 meters.

Path Planning: Rapidly-exploring Random Trees

The problem of path planning within a cluttered environment is far from trivial. The union of the numerous constraints imposed by the obstacles non-convex,. This makes finding a true optimal solution, especially in spaces of 3 or more dimensions, very computationally expensive. Instead, we applied a suboptimal path planning algorithm coded by Clifton, et. al. [#]. The algorithm is based on the Rapidly-exploring Random Trees algorithm, which is a highly popular member of the random sampling class of motion planning algorithms. This particular implementation utilizes two randomly-exploring trees, one expanding from the start node, and the other expanding from the goal node. As the trees expand, they look to connect a node from one tree to the other without a collision with an obstacle. If such a connection is made, then a connection from the start to finish has been found, and the search algorithm exits.

Random sampling methods greatly improve the speed of computing a feasible solution, but often produce results that are far from optimal. A variety of smoothing techniques exist to shorten the path found by the RRT search. The RRT Matlab software also includes a smoothing stage. The smoothing algorithm takes advantage of the requirement that all obstacles are defined as planes in 3-D space. The point where the RRT crosses the extended planes is used to form an initial smoothing. Then those points are moved along the planes to the obstacle boundary if removing the point is not feasible.

The smoothed curve is returned by the RRT planner as a series of line segments from the start to the goal. The final path that is returned is seen to be acceptably good for practical application on a consistent basis.

The results simulated in this project neglect many important factors that must be accounted for before the method will be suitable for use on a physical vehicle. The first, and biggest, factor is the time delay incurred by computation time of the RRT and MPC algorithms. The code will need to be optimized, parallelized, and ported to a faster language such as C. Some time delay will be inevitable, but when accounted for with a sufficiently small delay and accurate dynamic model, this factor will become manageable. For a good dynamics model, the dynamics of the low-level controls (PID, LQG, etc.) will need to be characterized. In this simulation, the low-level control was assumed to exist and have acceptable performance, but was not developed or modeled here.

@inproceedings{clifton2008evaluating,

title={Evaluating performance of multiple RRTs},

author={Clifton, Matthew and Paul, Gavin and Kwok, Ngai and Liu, Dikai and Wang, D-L},

booktitle={Mechtronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on},

pages={564--569},

year={2008},

organization={IEEE}

}

[#] Matthew Clifton, Gavin Paul, Ngai Kwok, Dikai Liu, Da-Long Wang, "Evaluating Performance of Multiple RRTs", IEEE conference on Mechatronic and Embedded Systems and Application, 2008