UAV-UGV-Human
Autonomous Collaboration

Austin Cassill, Sujay Bajracharya, William Heeter, Lakiel Wade and Dr. Shiqi Zhang

Abstract—

1. Approach

Our approach to execute a collaborative system between an aerial and a ground robot started with the implementation of the open source middleware Robot Operating System (ROS). ROS is a collection of tools, libraries, and conventions that aim to provide greater access to a wide variety of robotic hardware in order to allow for more robust and complex robotic behavior. ROS was developed in the early 2000s by multiple researchers at multiple universities for multiple different robot platforms. The idea was that through collaborative robotics research, all can advance their knowledge and development quicker and more thoroughly as opposed to separately. We used ROS version Kinetic (released in May 2016) as it was the newest version available when we began our research. We used ROS Kinetic on computers running the Linux operating system build Ubuntu 16.04.

Constructing a system in which multiple robots operating independently requires the utilization of multiple ROS masters. While there are multiple approaches to this within the ROS community, we found them to be too robust and computationally heavy for our purposes and chose to implement our own. Since our collaboration is based in message communication between the robots, rather than implementing a ROS multi-master system, we opted to run individual ROS masters on two separate computers (one for each robot) and develop a TCP server-client model that can pass simple messages between the two computers over Wi-Fi. The robots can then send messages to the clients and the server can pass the messages between the computers so that each robot can observe the state of the other. The Wi-Fi network is provided by the aerial robot itself, the Parrot Bebop 2 Power drone, and has a range of approximately 300 meters.

Each robot operates according to its own task while constantly updating a topic in ROS that informs the other robot of its current state (i.e. moving, stationary, landing, etc.). Based on one robots state, the other robot may adjust its execution of a task to better accommodate the other robot or maintain safety during operation. For example, if the UGV is currently moving, and the UAV is currently performing the task of landing, the UAV may wait or follow the UGV until it is stationary again and then proceed to land safely.

The ground robot, whose primarily responsible for navigating the environment, uses a Simultaneous Localization and Mapping (SLAM) algorithm to create and utilize a map of the given space. Localization for the ground robot is achieved using Adaptive Monte Carlo Localization (AMCL).

The aerial robot needs to be able to take off and land safely on the ground robot and this is achieved by tracking an Augmented Reality (AR) marker. The camera on the UAV tracks the marker, measuring its orientation to the marker in quaternions. A Python script then translates those measurements into Euler headings, and directs the UAV to approach the marker in such a way as to line itself up lock onto the marker and then start its descent. If at any point in the UAV loses its lock on the AR marker, it stops its descent, readjusts its position in order to relock, and then continues its landing procedure.

Autonomous identification of safe landing and takeoff zones is achieved through the implementation of artificial potential fields. Artificial potential fields programmatically generate an attractive or repulsive force to objects or voids within an area. The ground robot is repulsed by potential obstacles that could impair a safe drone landing.

2. Introduction

We set out to investigate the possibilities of creating a collaborative system between a UAV (unmanned aerial vehicle) and a UGV (unmanned ground vehicle). We decided to focus on the landing-takeoff problem, trying to develop a system for the UAV to safely and reliably take off from and land on the ground robot. We also focused on developing a method for the robots to autonomously identify safe areas for takeoffs and landings of the UAV.

2.1. Another Subsection

2.1.1. Subsubsection Heading Here. Subsubsection text here.

3. Experiments

4. Conclusion

The conclusion goes here.

Acknowledgments

The authors would like to thank...

References

[1]