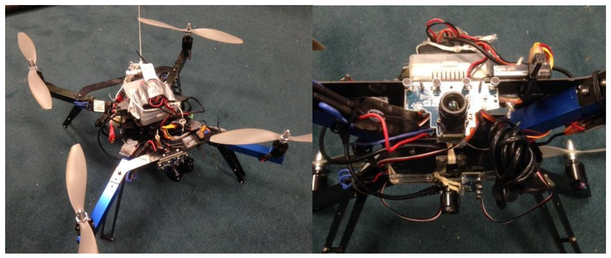
Trajectory Following in GPS Denied Environments for UAVs using Receding Horizon Control

Sam Zeng, M. Talha Agcayazi Advisor: Drew Bagnell

THE ROBOTICS INSTITUTE

MOTIVATION

PLATFORM



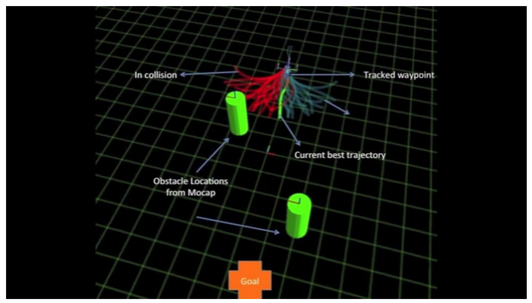
REACTIVE CONTROL

[1]

Pros:
- No Mapping

Cons:
- Myopic

DELIBERATIVE PLANNING

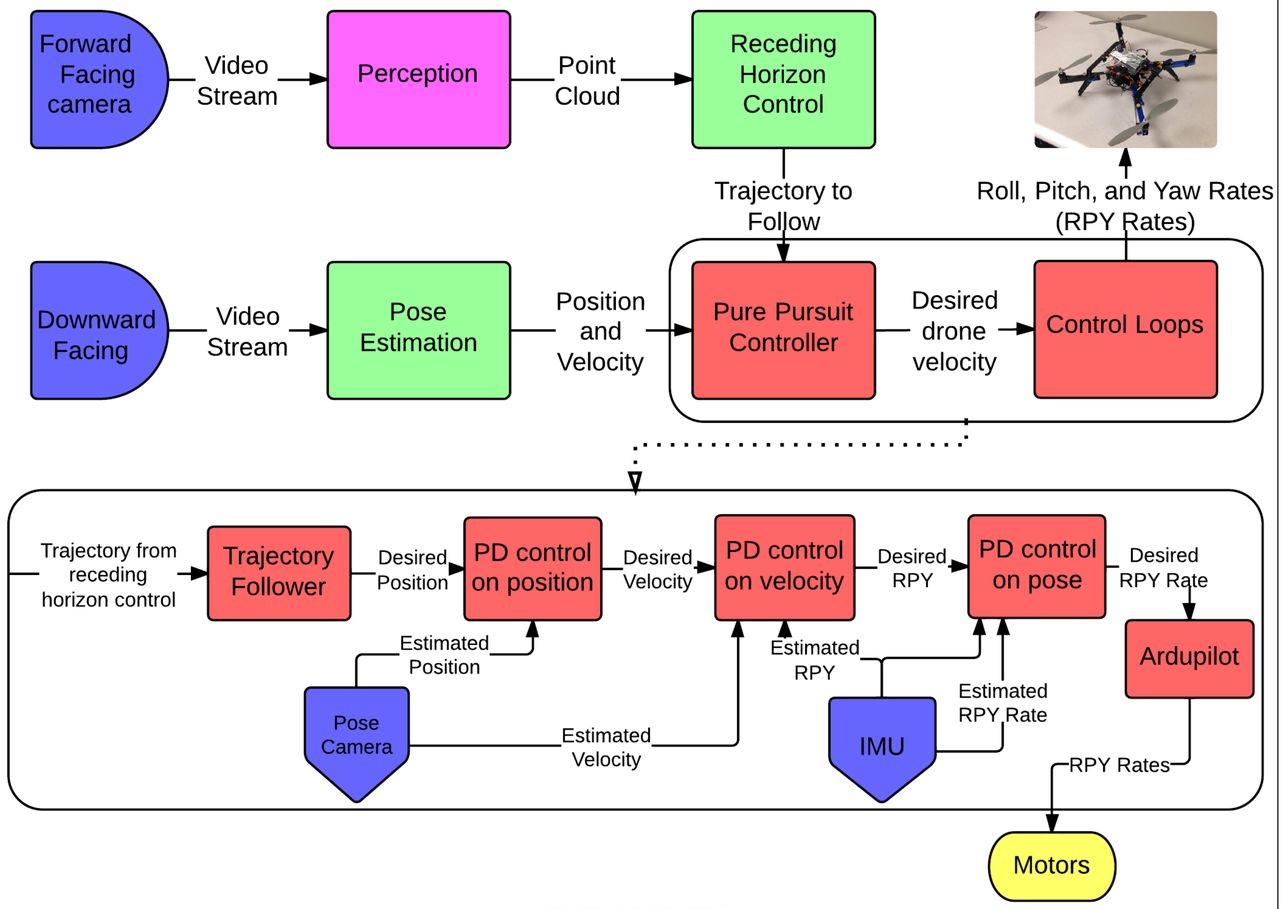


[2]

Pros:
- Goal Planning

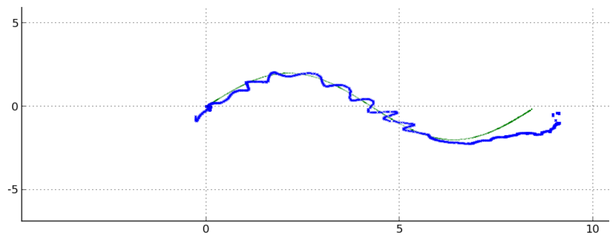
Cons:
- Computationally more expensive.

APPROACH

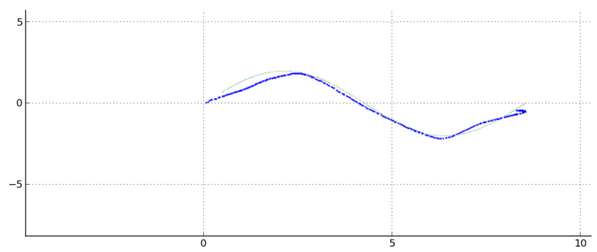


RESULTS

- Initially, the drone oscillated significantly around its desired trajectory as shown below:



- Tuning several parameters allowed us to rapidly improve trajectory following performance.

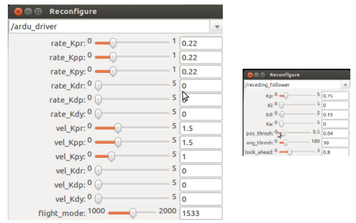


FUTURE WORK

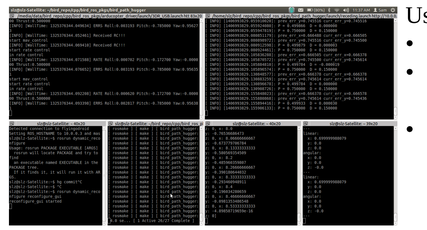
- We plan to test the whole receding horizon pipeline in a dense forest setting once the vision side of the pipeline is working.

CHALLENGES & SOLUTIONS

Tuning:
- Hard to change gains during program runtime.

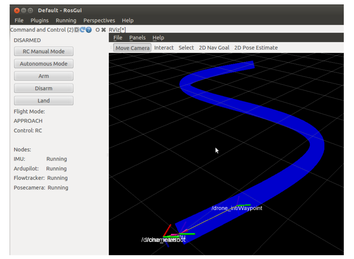


Command and Control Interface:



Using Terminals:
- Confusing
- No software kill switch
- Hard to track running nodes

Unified GUI
- Easy to use
- Safety buttons implemented
- Node status menu
- Ability to populate other plugins in the same window.



Altitude hold mode:
- A new flight mode that uses sonar and barometer sensors was implemented for altitude control.

ACKNOWLEDGEMENTS

- We would like to express our gratitude towards Dey, Matt, Shaurya, Drew and Martial for their guidance and support.
- Our thanks are extended to all those who made RISS possible.

REFERENCES

- [1] Stephane Ross, Narek Melik-Barkhudarov, Kumar Shaurya Shankar, Andreas Wendel, Debadeepta Dey, J. Andrew (Drew) Bagnell, and Martial Hebert, "Learning Monocular Reactive UAV Control in Cluttered Natural Environments," IEEE International Conference on Robotics and Automation, March, 2013.
- [2] "BIRD MURI." RobotWhisperer RSS. N.p., n.d. Web. 05 Aug. 2014.

Closing the Control Loop Around Visual Odometry

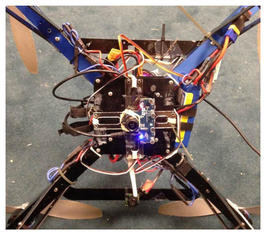
RISS: Sam Zeng, M. Talha Agcayazi
Advisor: Drew Bagnell
Carnegie Mellon University, Robotics Institute

MOTIVATION

- Getting a quadcopter to fly through dense forest using receding horizon control.
- This requires a set of controllers that would enable the quadcopter to follow trajectories.

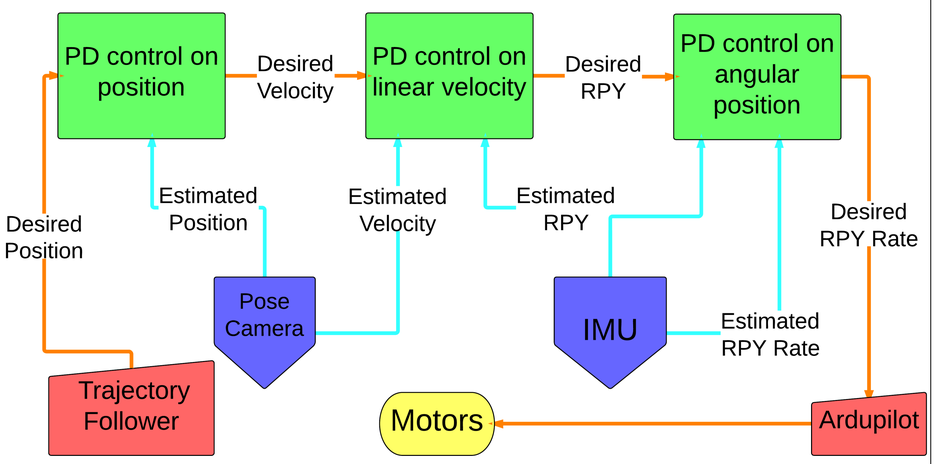
PLATFORM

- IMU
- PSEye
- Odroid XU
- Ardupilot



Method

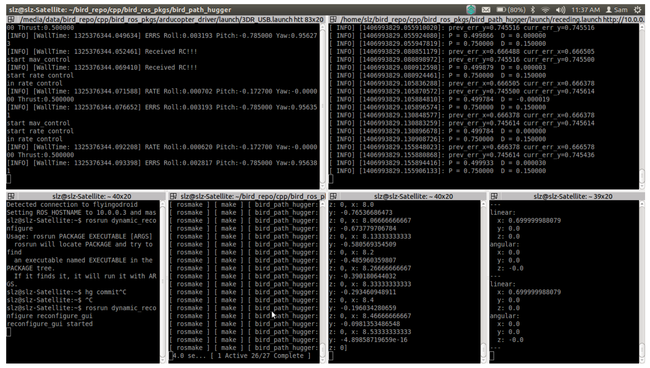
- A simple cascading PID control loop should allow the drone to follow trajectories. We will implement this in three steps.
  1. Build and tune PD controllers that allow us to achieve command velocities.
  2. Build and tune PD controller to generate command velocities to get the drone to hover around a point in the world frame.
  3. Integrated this with existing code so the drone follows trajectories by attempting to “hover” around a constantly moving waypoint.



CHALLANGES

User Interface

- Hard to keep track of running ROS nodes with Terminal screens.
- Need a software kill switch.
- Visualizations implemented.



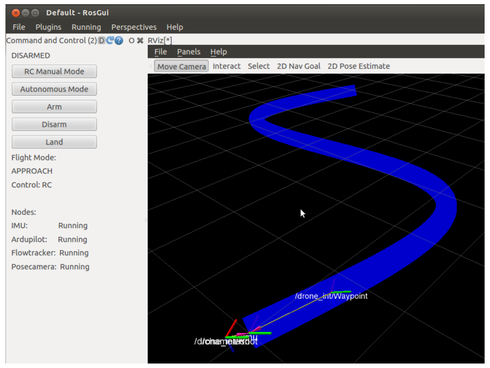
Tuning

- Hard to change gains during program runtime.

SOLUTIONS

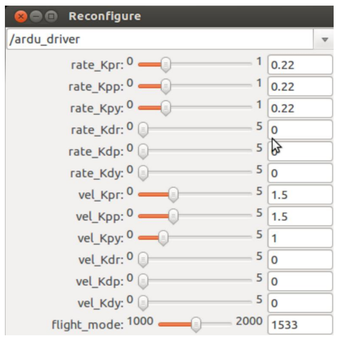
Command and Control Interface

- Status Bar to show Node status
- Simple commands like land and kill button implemented.
- Easy to add different ROS GUI plugins like RVIZ to visualize on the same window.



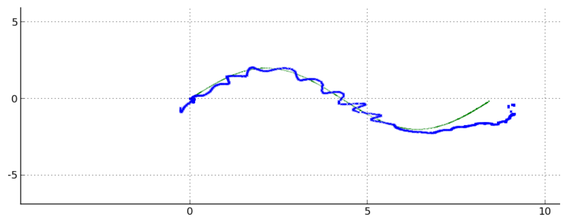
Tuning

- Dynamic Reconfigure was implemented.

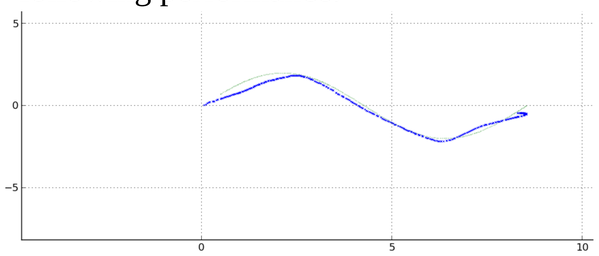


RESULTS

- Initially, the drone oscillated significantly around its desired trajectory as shown below:



- Tuning several parameters through dynamic reconfigure allowed us to rapidly improved trajectory following performance.



CONCLUSION and FUTURE WORK

- Our current implementation of these three layers of PD controllers has so far demonstrated consistent and robust control of the drone.
- We plan to also compare the drone’s trajectory following with respect to ground truth pose estimation with Vicon motion capture.
- We plan to test the whole receding horizon pipeline in a dense forest setting once the vision side of the pipeline is working.

ACHNOWLEDGEMENTS

- We would like to express our gratitude towards Dey, Matt, Shaurya, Drew and Martial for their guidance and support.
- Our thanks are extended to all those who made RISS possible.