Marker Based Localization of a Quadrotor

Akshat Agarwal & Siddharth Tanwar

Objective

Introduction

Objective: To implement a high level control pipeline on a quadrotor which could autonomously take-off, hover over a marker and **land on it with high precision**.

- Quadrotors have Vertical TakeOff and Landing (VTOL) ability
- Limited flight time because of battery technology
- In any autonomous deployment, quads must be able to find a suitable landing pad and land on it
- In any long term deployment, quads need ability to land on a charging platform and dock with it, autonomously - Highly precise!

Paper Followed: Yang, Shuo, et al. "Precise quadrotor autonomous landing with SRUKF vision perception." *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.

What makes landing difficult?

- Ground effect: When a quad flies close to the ground, the air being pushed downward by rotors has no place to go, build up in air pressure. Gives non-linear lifting forces, making landing much more unstable.
  Solution: Use landing platform above the ground
- Mechanical docking of chargers needs extremely high precision.
  Mitigating solution: Use guiding mechanical structures, like a cone

Target: Perform landing with error < 5cm

Hardware Setup

Nayan Quadrotor

- A high performance quadrotor by Aarav Unmanned Systems (AUS)
- Uses an Odroid-XU4 on-board computer running Lubuntu 14.04 (ARM Octa-core, 2GB RAM)
- Twin ARM Cortex M4 Processor with a RTOS (Real-time OS) for the flight controller (HLP + LLP)

Image Source: http://www.aus.co.in/

Sensor Setup on Nayan

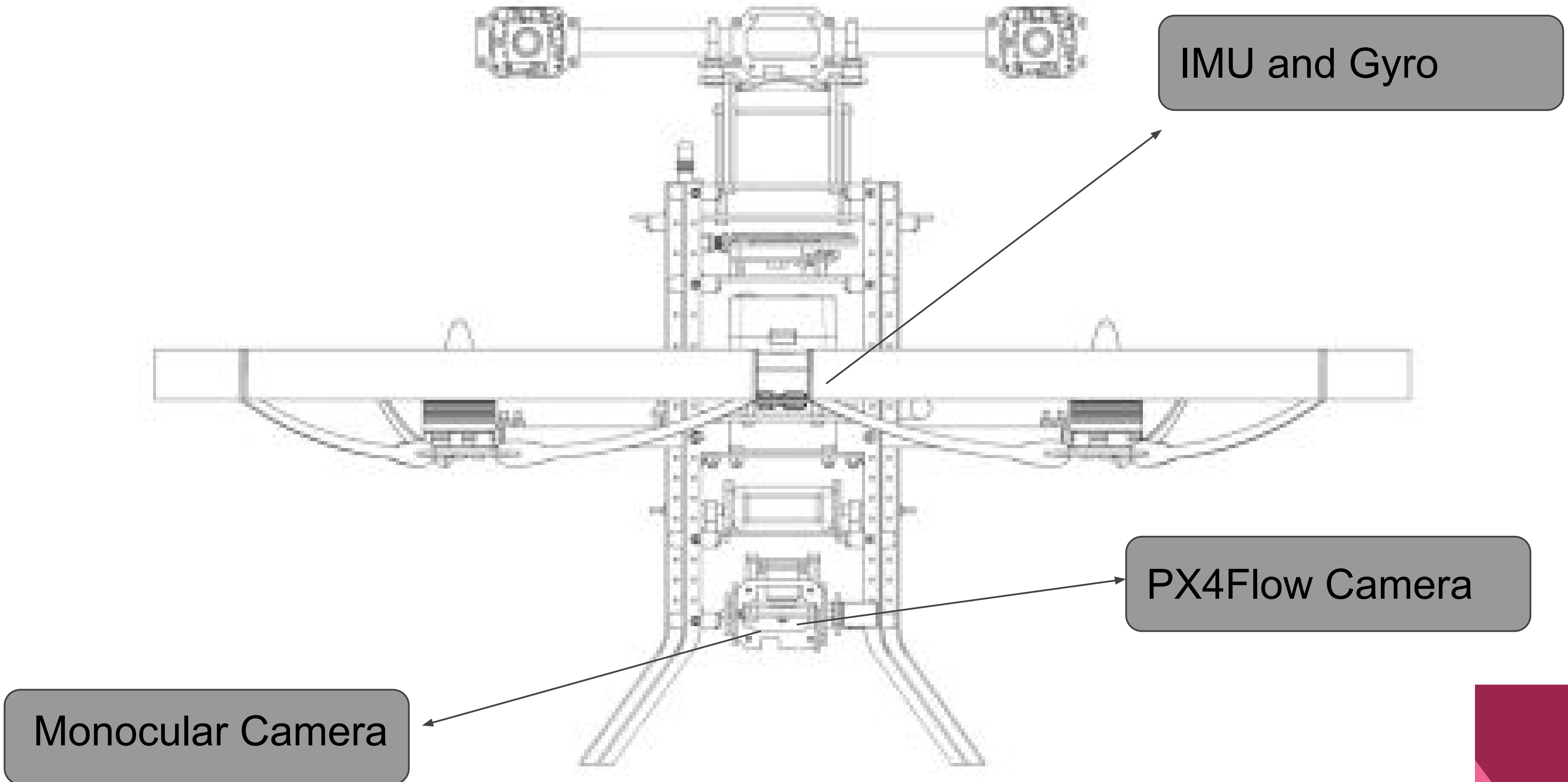


Image Source: http://www.aus.co.in/

Monocular Camera

- Matrix Vision Bluefox USB 2.0 MLC (high quality gray scale CMOS) camera
- Resolution : 752 x 480
- Max. frame rate [Hz] : 90
- Adjustable exposure and gain for adapting to low lighting conditions, but performs much better in well lit environments
- The package - Bluefox Driver on ROS

Image Source: http://www.aus.co.in/

PX4Flow (Optical Flow sensor)

Optical flow is the pattern of apparent motion of objects, surfaces and edges in a scene caused by the relative motion between an observer and the scene

- Optical Flow processing (gives x-,y- velocities) @ 400 Hz
- Installed facing downwards
- Supposed to work in both indoor and outdoor low-lighting conditions
- A supplementary sonar sensor gives distance to ground
- The package - PX4Flow Driver on ROS

Image Source: http://www.aus.co.in/

IMU and Gyroscope

- Provides linear acceleration (-8G to +8G) and angular rates (max 2000 deg/sec) to flight controller
- Linear acc using accelerometers, and changes in pitch/roll using gyroscopes
- An absolute reference frame (towards North)
- Installed on the flight controller board - Body frame
- Used to estimate orientation of the quad

Image Source: https://pixhawk.org/

Architecture

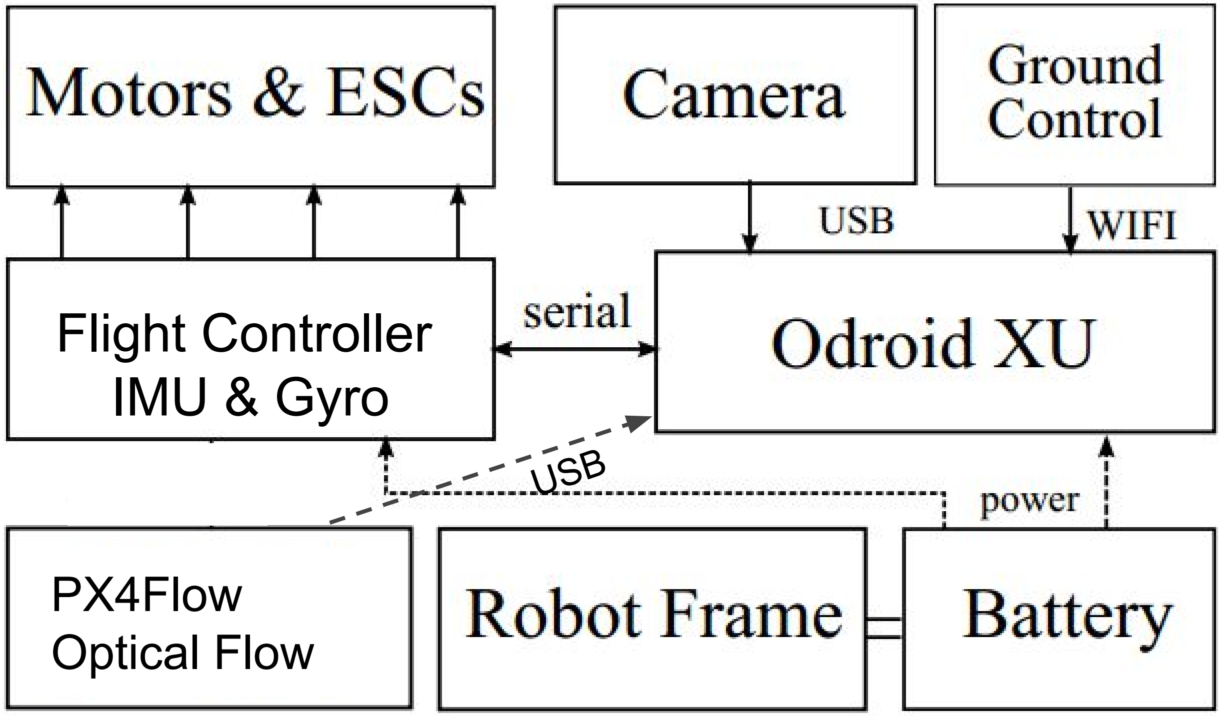


Image Source : Yang, Shuo, et al. "Precise quadrotor autonomous landing with SRUKF vision perception." *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015

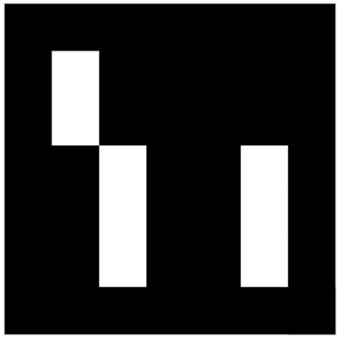
Libraries

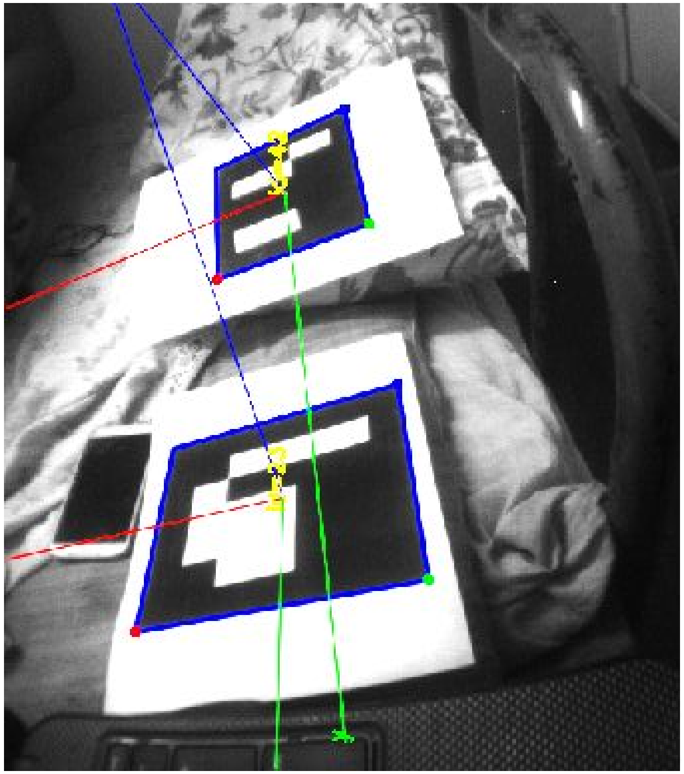
ROS - Robot Operating System

- A flexible framework for writing robot software and managing inter-process communication
- ROS offers a huge community, all sensors have robust ROS packages
- Used ROS-Indigo on Nayan to integrate all sensors and observe from ground station

ArUco Markers and Library

- Provides a library to generate markers that are easily detectable via camera
- Comes with an image processing pipeline as well to obtain the pose of the marker in Camera Frame
- Integrates with ROS through the aruco_ros package





Explored Frameworks

Marker Detection

- The marker detection and identification steps are:
  - Image segmentation
  - Contour extraction and filtering
  - Marker Code extraction
  - Marker identification and error correction
- Obtain the corner points of detected marker
- Since real size of marker is known, correspondence easily established
- PnP problem - solved to get R,t!

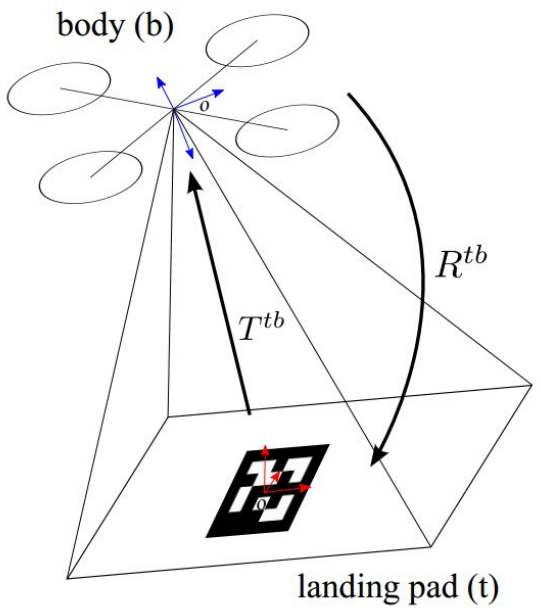


Image Source : Yang, Shuo, et al. "Precise quadrotor autonomous landing with SRUKF vision perception." *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015

Rotation Compensation

- IMU rotation better than the rotation from PnP solver (ArUco), especially in pitch and roll components
- Decomposes rotation into two components: $R = R_{tilt}R_{torsion}$, where $R_{torsion}$ only involves rotation around yaw axis, while R_{tilt} contains the rotation on pitch and roll axes.
- The torsion component of PnP solver rotation is multiplied with tilt component of IMU rotation, to get a stable, precise rotation measure

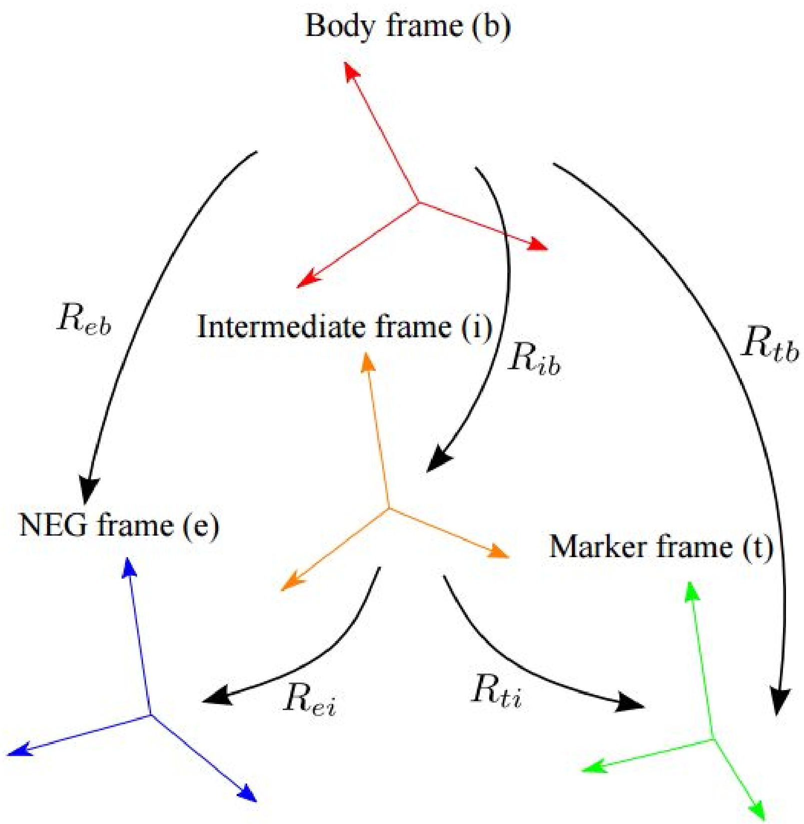


Image Credits : Yang, Shuo, et al. "Precise quadrotor autonomous landing with SRUKF vision perception." *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015

SRUKF (Square Root - UKF)

- UKF: Most computationally intensive step is calculating new set of sigma points at each time update
- In SRUKF, the square root S of P is propagated directly
- For state-space formulation, it has time complexity O(L^2) unlike UKF which has time complexity O(L^3)
- Is also numerically more stable and guarantees PSD-ness of the state covariances

$$\boldsymbol{\mathcal{X}}_{k-1} = [\hat{\mathbf{x}}_{k-1} \quad \hat{\mathbf{x}}_{k-1} + \gamma \mathbf{S}_k \quad \hat{\mathbf{x}}_{k-1} - \gamma \mathbf{S}_k]$$

$$\boldsymbol{\mathcal{X}}^*_{k|k-1} = \mathbf{F}[\boldsymbol{\mathcal{X}}_{k-1}, \mathbf{u}_{k-1}]$$

$$\hat{\mathbf{x}}^-_k = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{X}^*_{i,k|k-1}$$

$$\mathbf{S}^-_k = \mathrm{qr}\left\{\left[\sqrt{W_1^{(c)}}\left(\boldsymbol{\mathcal{X}}^*_{1:2L,k|k-1} - \hat{\mathbf{x}}^-_k\right) \quad \sqrt{\mathbf{R}^{\mathbf{v}}}\right]\right\}$$

$$\mathbf{S}^-_k = \mathrm{cholupdate}\left\{\mathbf{S}^-_k \,,\, \mathcal{X}^*_{0,k} - \hat{\mathbf{x}}^-_k \,,\, W_0^{(c)}\right\}$$

$$^5\, \boldsymbol{\mathcal{X}}_{k|k-1} = [\hat{\mathbf{x}}^-_k \quad \hat{\mathbf{x}}^-_k + \gamma \mathbf{S}^-_k \quad \hat{\mathbf{x}}^-_k - \gamma \mathbf{S}^-_k]$$

$$\boldsymbol{\mathcal{Y}}_{k|k-1} = \mathbf{H}[\boldsymbol{\mathcal{X}}_{k|k-1}]$$

$$\hat{\mathbf{y}}^-_k = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_{i,k|k-1}$$

Image Source : Yang, Shuo, et al. "Precise quadrotor autonomous landing with SRUKF vision perception." *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015

Models

$$x_i = \begin{bmatrix} T^{tb}{}_i \\ q^{tb}{}_i \\ v^t{}_i \\ \omega^b{}_i \end{bmatrix} = \begin{bmatrix} T^{tb}{}_{i-1} + v^t{}_{i-1}\Delta t \\ q^{tb}{}_{i-1} \times q(\omega^b{}_{i-1}\Delta t) \\ v^t{}_{i-1} + a^t{}_{i-1}\Delta t \\ \omega^b{}_{i-1} \end{bmatrix}$$

Motion Model

$$y_i = \begin{bmatrix} T^{tb} \\ q^{tb} \\ \frac{1}{w_1+w_2}[w_1(v^t{}_i + a^t\Delta t) + w_2\frac{T^{tb} - T^{tb}{}_i}{\Delta t}] \\ \omega^b \end{bmatrix}$$

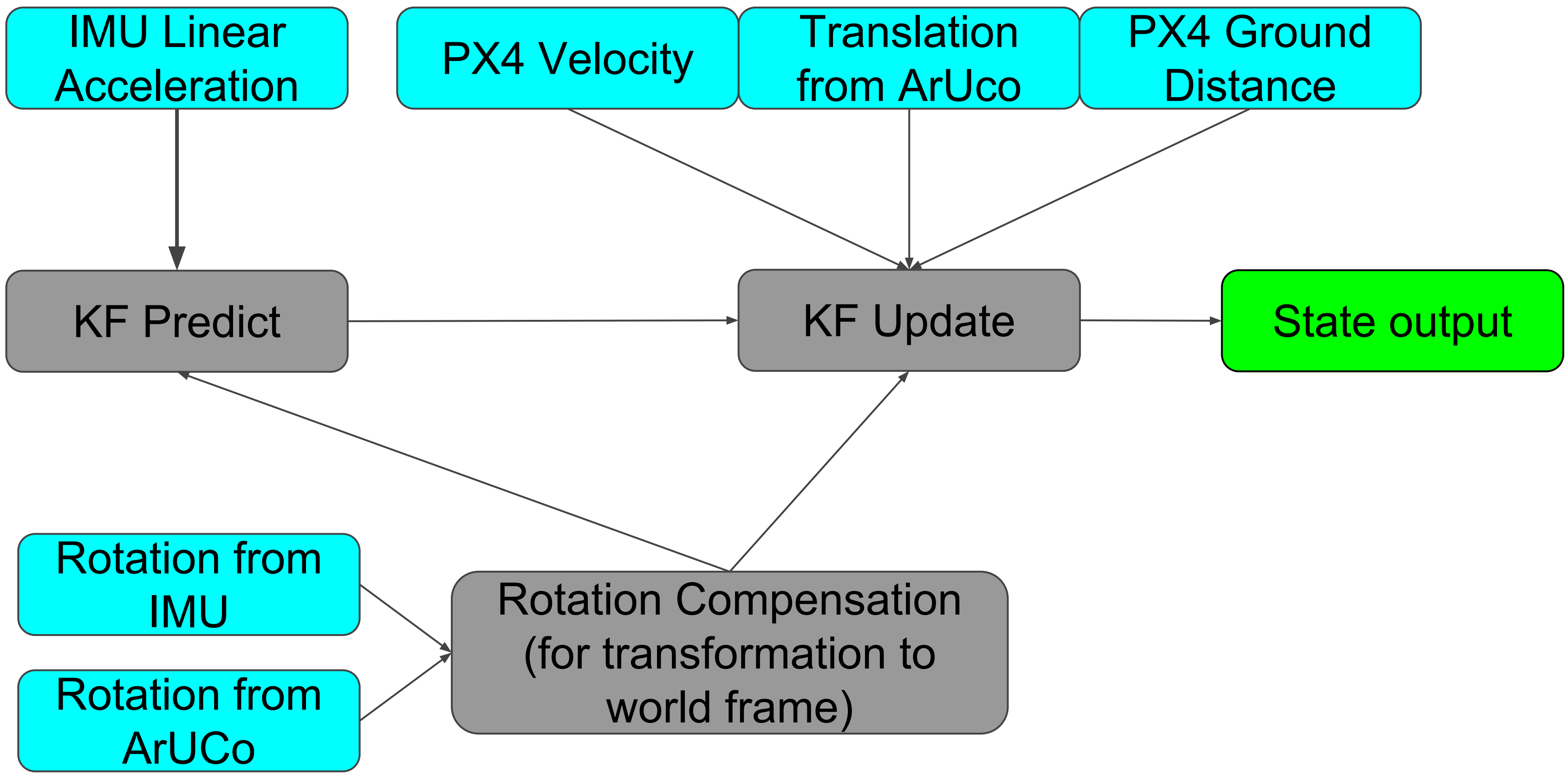
Measurement Model

UKF and KF

- After trying with previous model using UKF, we decided to use a linear model and a simple Kalman Filter on it
- State $\mathbf{x} = [x;y;z;x';y';z']$
- Linear motion model where $\mathbf{x_i = x_{i-1} + x'_{i-1} * dt}$
- The action $\mathbf{u = [a_x;a_y;a_z]}$

- Measurement model observes translation vector from camera, velocity from optical flow sensor
- Height from ground also observed using sonar

Data Flow

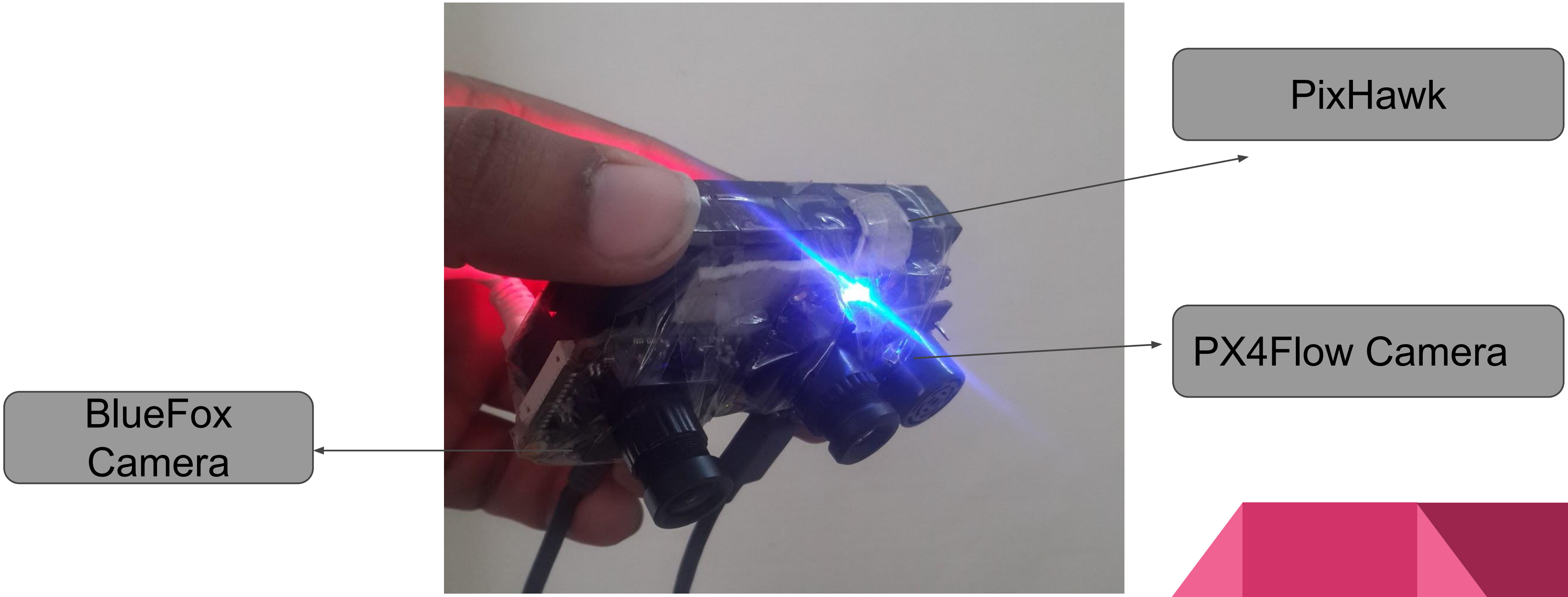


IMU Linear Acceleration
PX4 Velocity
Translation from ArUco
PX4 Ground Distance
KF Predict
KF Update
State output
Rotation from IMU
Rotation from ArUCo
Rotation Compensation (for transformation to world frame)

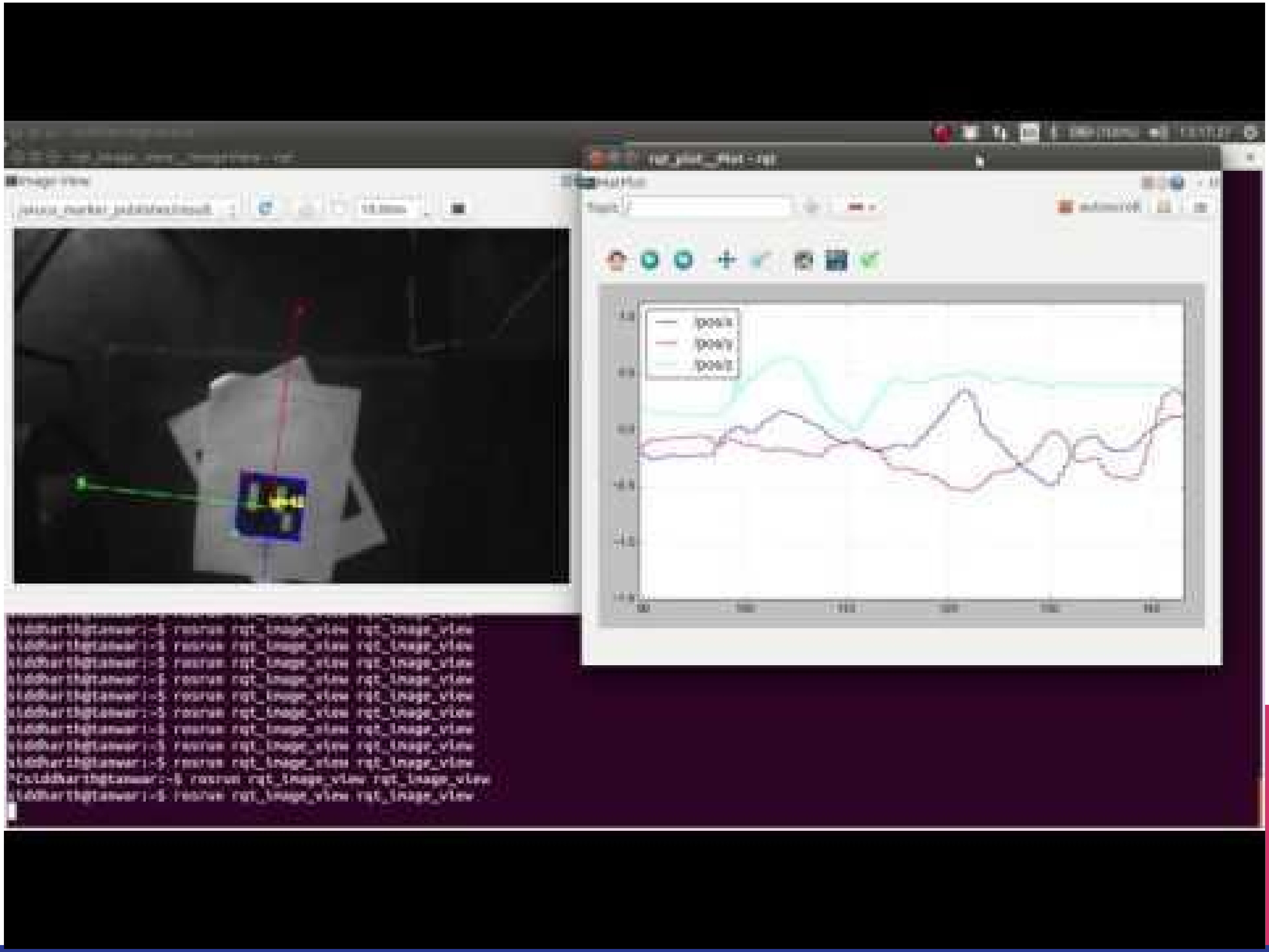
What did we accomplish?

Testing: Sensor Set

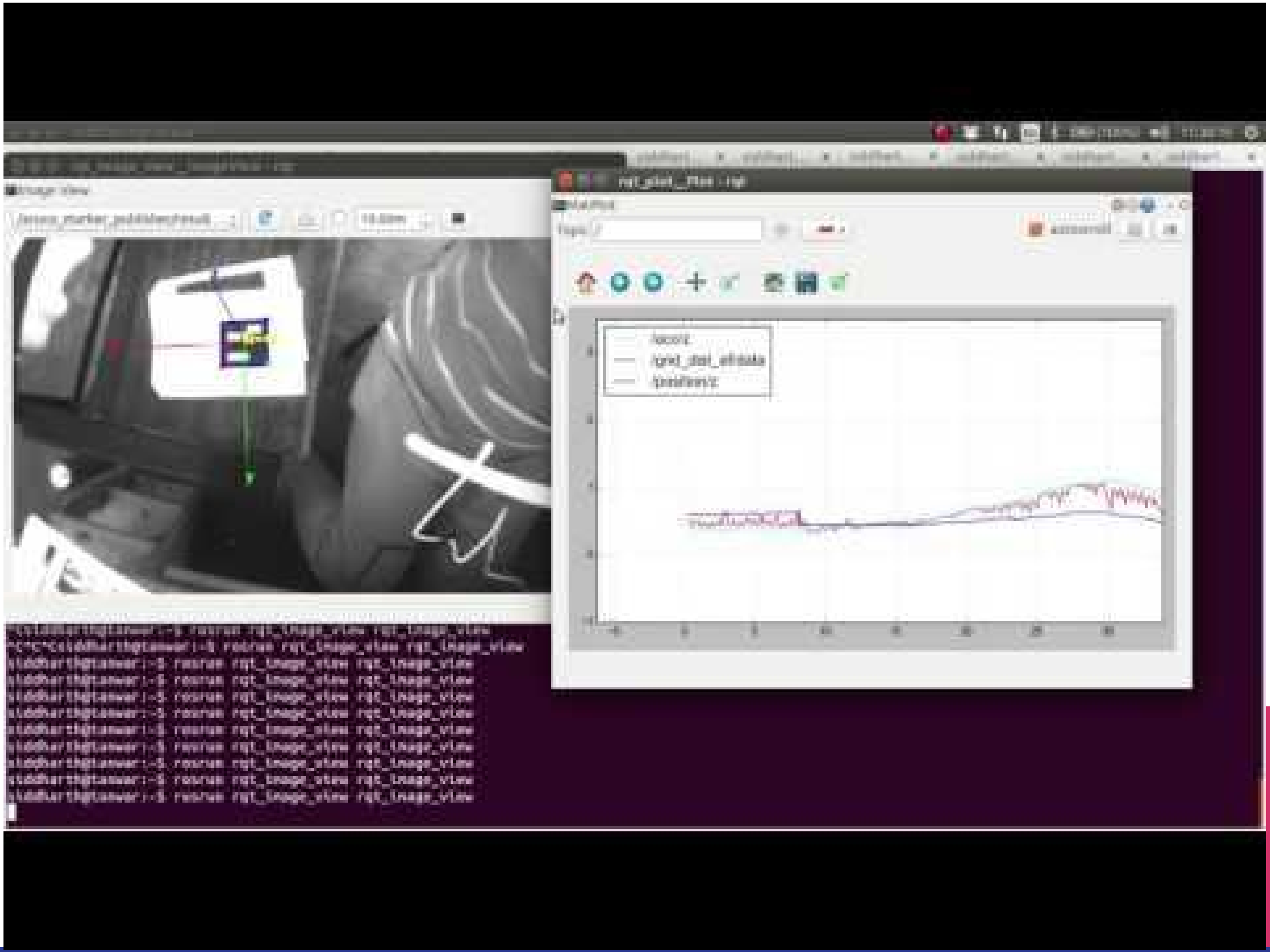
- Problems with Nayan
- Used Pixhawk + Camera + PX4Flow camera to make a Rosbag File



ArUco Marker Detection



A Localization Example



Conclusion and Future work

- We built an architecture for implementing SRUKF, UKF or KF on the quadrotor
- Tracked pose of the quadrotor using the ArUco markers
- Tracked Pose of the quadrotor using variants of Kalman Filter

- Use the UKF/SRUKF/KF on Nayan Platform
- Send controls to the quad's flight controller and observe its performance, tuning the filter accordingly to get precision landing

Acknowledgements

- Thanks to Mr. Krishna Raj Gaur for helping us setup the testing module in the absence of the quadrotor
- Thanks to Mr. Shakti and Mr. Radhe Shyam, Intelligent Systems Lab for their support

Thank You

Questions ?