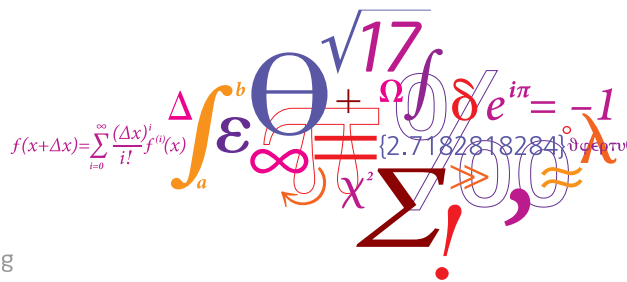
Self-Contained Autonomous Quadrotor

Søren Krogh Andersen

DTU Electrical Engineering, Automation and Control, Technical University of Denmark (DTU)



DTU Electrical Engineering
Department of Electrical Engineering

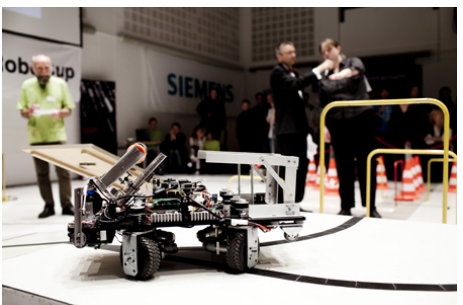
Outline

- Introduction
  - Goal
  - Motivation
  - Specifications
  - Solution
- Visual Odometry
  - GPFVO
  - Transformation Estimation
  - Hiding Latency
- Hardware
  - Block Diagrams
- Controller
  - Quadrotor Dynamics
  - Attitude and Altitude
  - Position
- Results

Goal I

Long Term

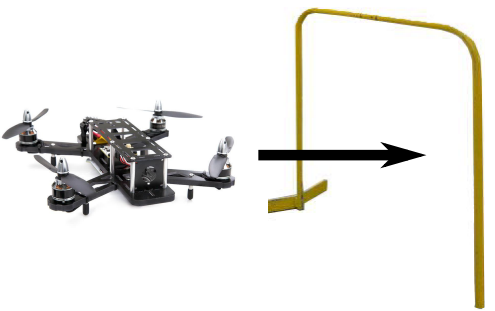
Creating a quadrotor, that can participate in the annual Robocup competition.



Goal II

Sort Term (this project)

Constructing an autonomous quadrotor, capable of navigating through an opening 45 cm wide and 50 cm



Motivation

Why quadrotor?

- Cheap
- Mechanically simple, makes construction easy
- Availability of components
- “Hot” topic

Why Aimed at Robocup?

- Specific
- Fun
- Similar tasks in the industry:
  - Inventory counting
  - In-house transport system

Motivation

Why quadrotor?

- Cheap
- Mechanically simple, makes construction easy
- Availability of components
- “Hot” topic

Why Aimed at Robocup?

- Specific
- Fun
- Similar tasks in the industry:
  - Inventory counting
  - In-house transport system

Specifications

Requirements

- Self-contained
- Autonomous
- Safe
  - Protection against propellers
  - Manual overwrite

Success Criteria

- Hover inside an area of length 45 cm × width 45 cm for at least 30 s without leaving the area.
- Fly through an opening of width 45 cm × height 50 cm without hitting the sides.
- Manual controller where inputs to the drone are: **Altitude rate, yaw rate, pitch, roll.**

Specifications

Requirements

- Self-contained
- Autonomous
- Safe
  - Protection against propellers
  - Manual overwrite

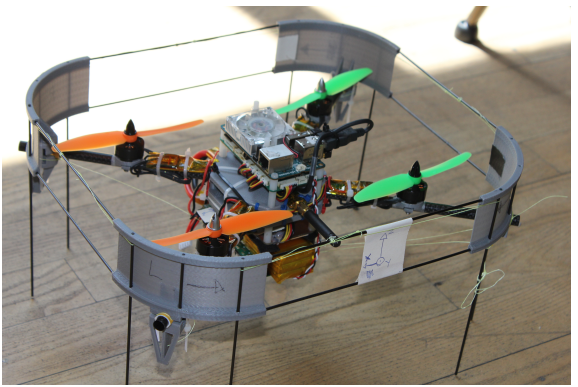
Success Criteria

- Hover inside an area of length $45\,\text{cm} \times$ width $45\,\text{cm}$ for at least $30\,\text{s}$ without leaving the area.
- Fly through an opening of width $45\,\text{cm} \times$ height $50\,\text{cm}$ without hitting the sides.
- Manual controller where inputs to the drone are: **Altitude rate, yaw rate, pitch, roll.**

Solution I

Small Scale Quadrotor

A quadrotor of size $30\,\text{cm} \times 35\,\text{cm}$ is constructed. Uses Visual Odometry for navigation. On-board Linux computer for computation.



Solution II

Sub-Tasks

- Mechanical construction
- Visual Odometry
- Controller
  - Attitude, altitude
  - Position

Ground Projected Features Visual Odometry

Theory of Operation

- EKF for state estimation
- Uses IMU to propagate state estimate
- Uses distance sensor to update altitude estimate
- Uses camera to estimate motion and update estimate

State Representation

- Current pose
- IMU bias
- Pose at time of previous image

$$\boldsymbol{x}_k = \left[{}^{I}_{G}\boldsymbol{q}^T, {}^{G}\boldsymbol{p}_I^T, {}^{G}\boldsymbol{v}_I^T, \boldsymbol{b}_g^T, \boldsymbol{b}_a^T, {}^{I}_{G}\boldsymbol{q}_p^T, {}^{G}\boldsymbol{p}_{I_p}^T, {}^{G}\boldsymbol{v}_{I_p}^T \right]^T$$

Ground Projected Features Visual Odometry

Theory of Operation

- EKF for state estimation
- Uses IMU to propagate state estimate
- Uses distance sensor to update altitude estimate
- Uses camera to estimate motion and update estimate

State Representation

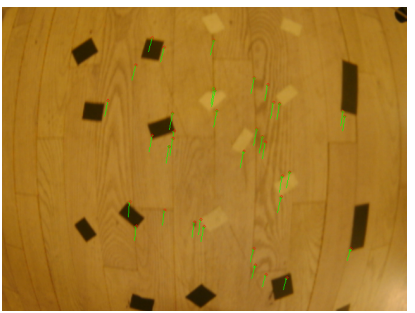
- Current pose
- IMU bias
- Pose at time of previous image

$$\boldsymbol{x}_k = \left[{}^{I}_{G}\boldsymbol{q}^T, {}^{G}\boldsymbol{p}_I^T, {}^{G}\boldsymbol{v}_I^T, \boldsymbol{b}_g^T, \boldsymbol{b}_a^T, {}^{I}_{G}\boldsymbol{q}_p^T, {}^{G}\boldsymbol{p}_{I_p}^T, {}^{G}\boldsymbol{v}_{I_p}^T \right]^T$$

Transformation Estimation I

Feature Tracking

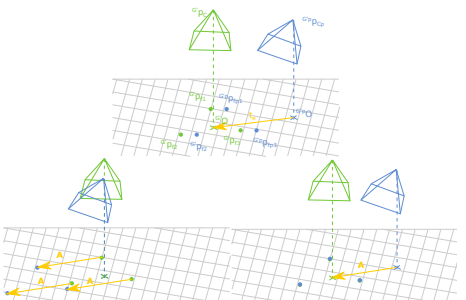
- Detect interesting points in image with GFTT
- Track feature movement with optical flow



Transformation Estimation II

Transformation Estimation

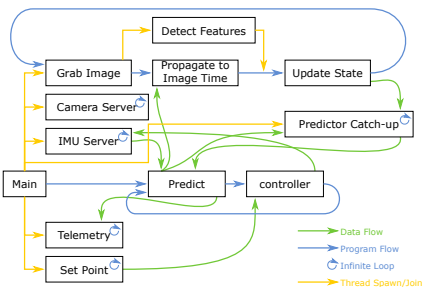
- Project features on ground
- Estimate feature movement



Hiding Latency and Speedup I

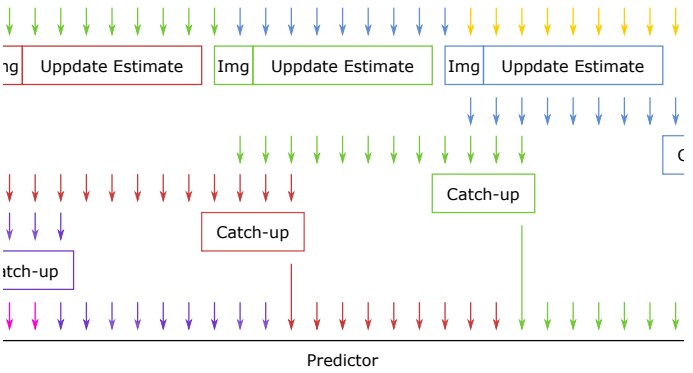
Multi-Threading

- Propagate state in parallel with feature tracking
- Predict state with IMU data to hide latency



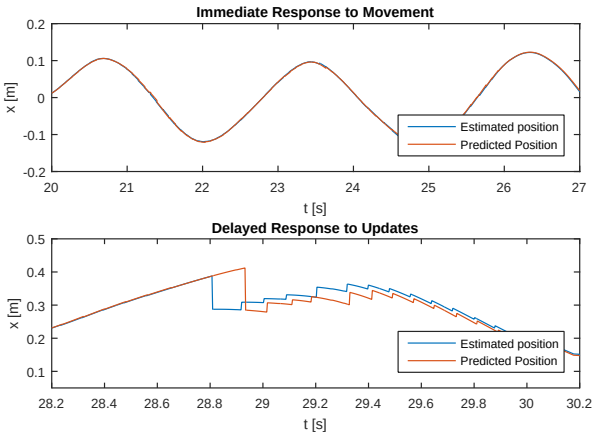
Hiding Latency and Speedup II

Prediction Timing I



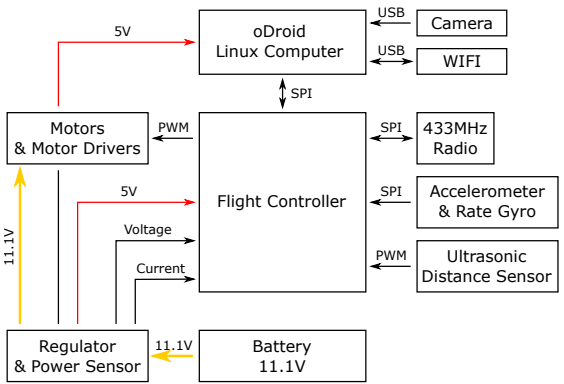
Hiding Latency and Speedup III

Prediction Timing II



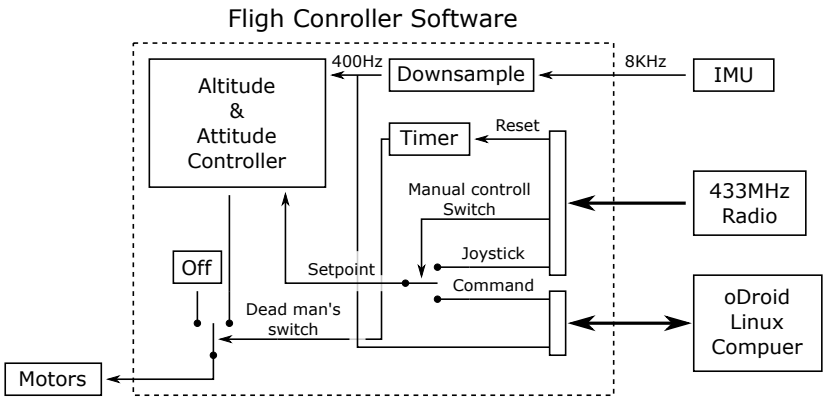
Block Diagrams I

Quadrotor Components



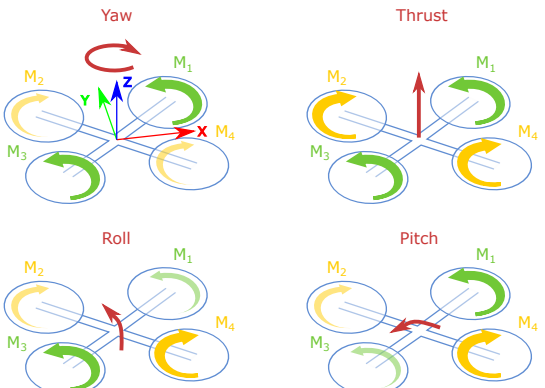
Block Diagrams II

Flight Controller



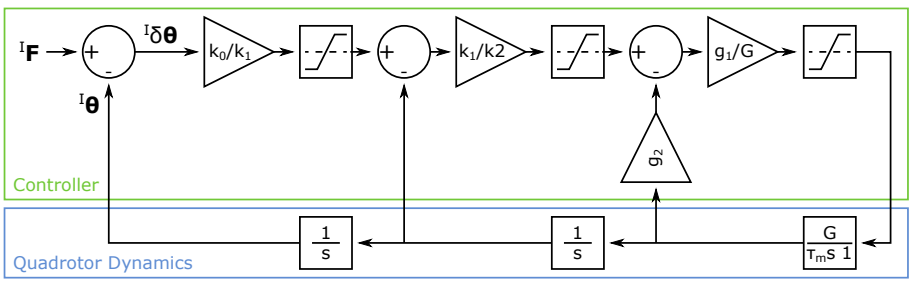
Quadrotor Dynamics

How a Quadrotor Steers



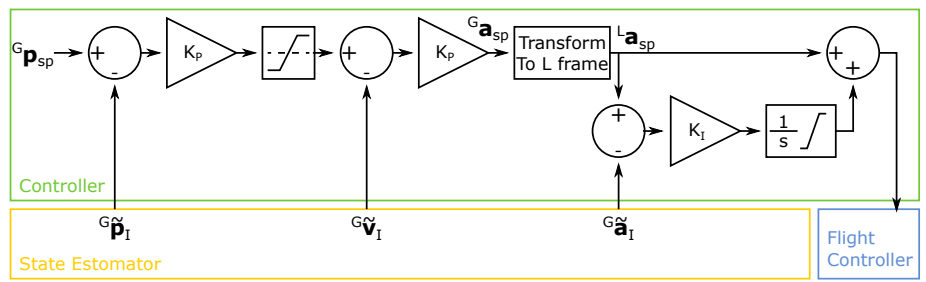
Attitude and Altitude

Attitude Controller



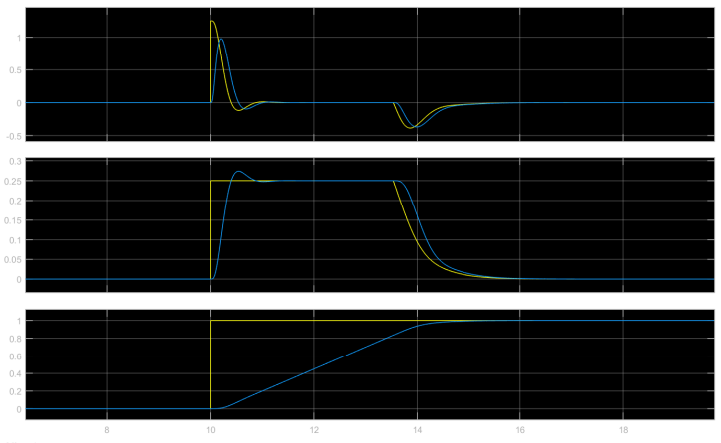
Position I

Position Controller



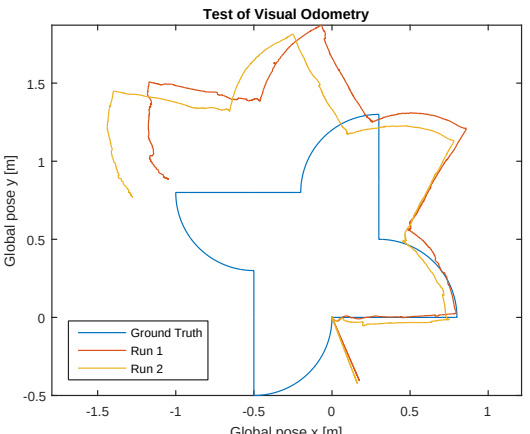
Position II

State Limitation



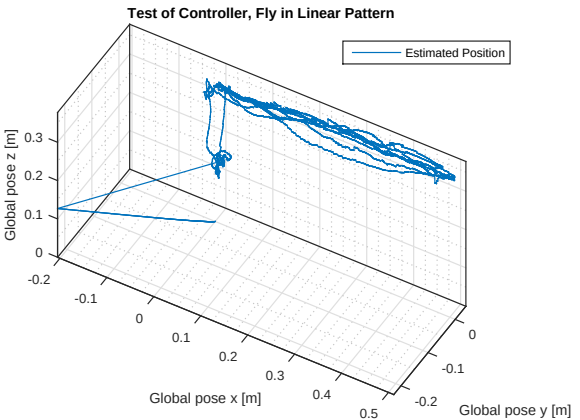
Results I

Visual Odometry



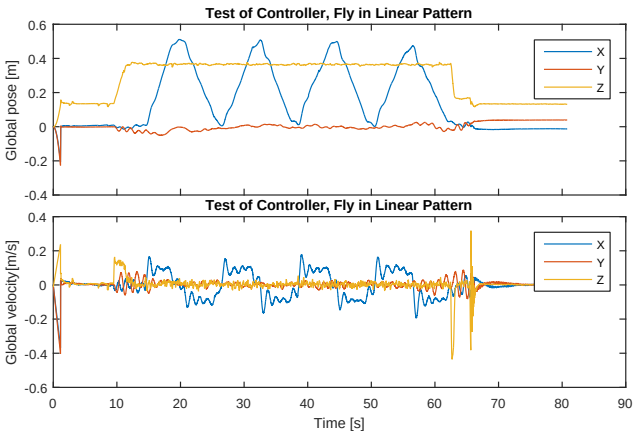
Results II

State Limitation



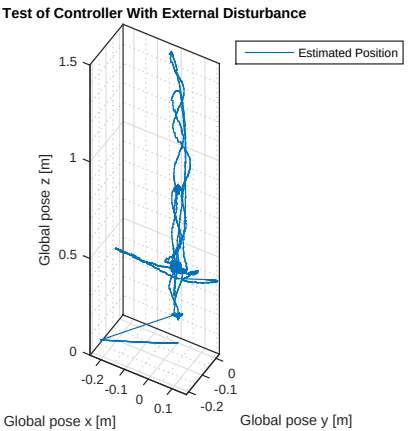
Results III

State Limitation



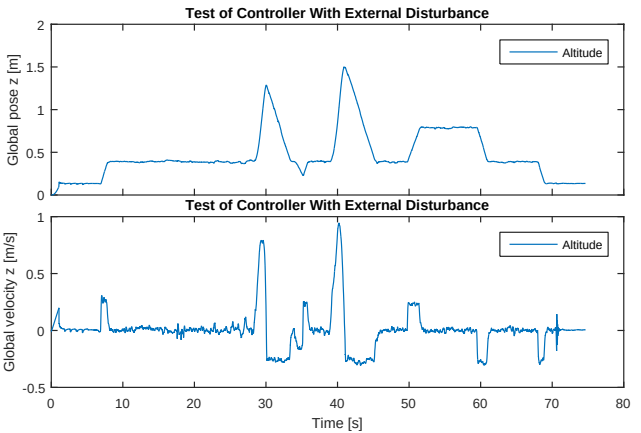
Results IV

State Limitation



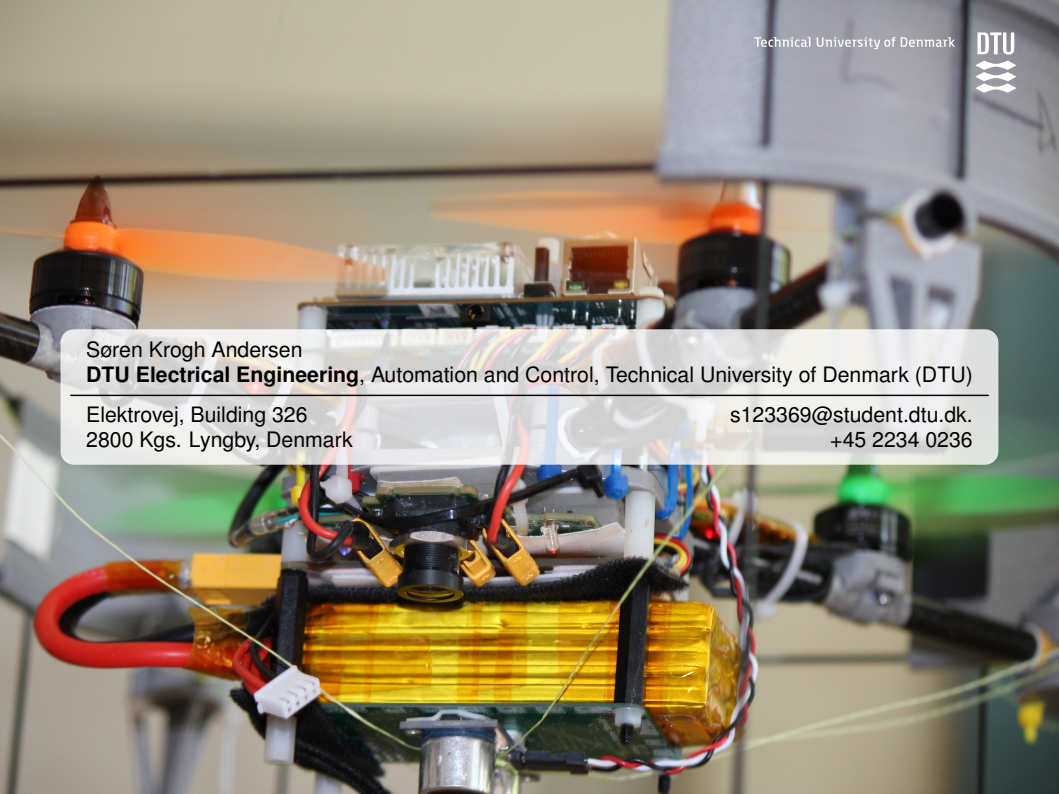
Results IV

State Limitation



Source

Source code available at `https://github.com/skrogh/oDroidDrone`



Søren Krogh Andersen
DTU Electrical Engineering, Automation and Control, Technical University of Denmark (DTU)

Elektrovej, Building 326
2800 Kgs. Lyngby, Denmark

s123369@student.dtu.dk.
+45 2234 0236