Towards Autonomous Landing for a Quadrotor, using Monocular SLAM Techniques

Jonatan Olofsson

Department of Computer and Information Science at Linköping University, Sweden

Master's Thesis Presentation, 2012-06-08

Presentation Outline

Background

- Increased interest in civilian applications.
- For many applications, a small scale vehicle is desired.
- MAV - Micro Air Vehicle; UAV weighing 5 kg or less[1].

[1] Definitions differ

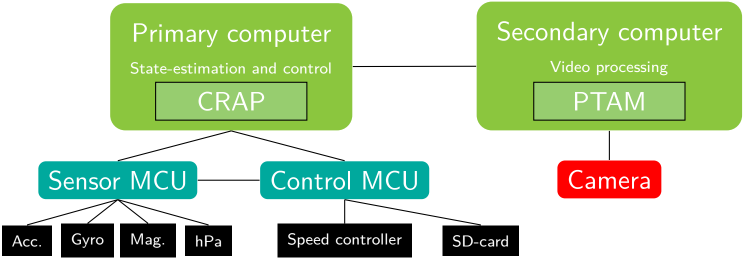
The LinkQuad Platform

Quadrotor research platform developed by AIICS at the Department of Computer and Information Science of Linköping University.

Used sensors: Accelerometers, gyroscopes, pressure sensor, camera

The LinkQuad Platform

- Dual gumstix micro-computers (Linux).
- Sensor-board with dual microcontrollers for sensor sampling, data logging and low-level control.



Problem Formulation

Primary goal: Develop a control system for the LinkQuad that use video-based positioning to provide stable landing.

Breakdown:

- Video-based SLAM.
- Sensor Fusion with available sensors.
- Use state-estimate for control.
- Generate control reference for landing procedure.

Method: Video-based Positioning

Extract 3D information from 2D video-stream.

- vSLAM - Visual SLAM.
- Simultaneous Localisation And Mapping to track features in the video-stream.
- Gives orientation and position relative to the features.
- High computational demands.
- Complex sensor measurements.

Method: Filtering

Current implementation is based on complementary filtering.

$$\text{angle} = (0.98) * (\text{angle} + \text{gyro} * \text{dt}) + (0.02) * (a_{\hat{x}})$$

Performance is adequate, but the c.f is difficult to extend.

Also, camera measurements

- cannot be used directly in the current filter,
- fits nicely into a standard state-space filter framework.

A high-level filtering framework with an advanced motion model was implemented and applied for state-estimation.

Method: Control

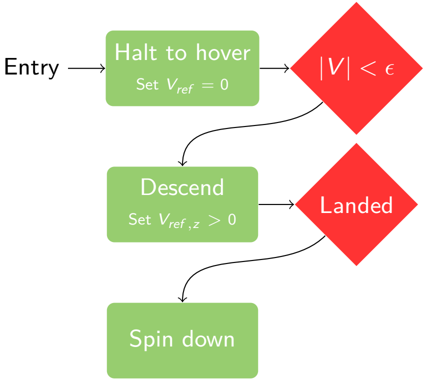
With a motion model available, control signals can be computed optimally. Linear Quadratic control offers

- simple implementation,
- simple tuning,
- optimal control.

Motion model needs to be linear, which is not the case for a quadrotor. This constraint can be circumvented by an extension to LQ control using the **State-dependent Riccati Equation**.

Method: Reference Generation

With properly implemented control, landing is a matter of descending steadily until landing is detected.



Method: Landing Detection

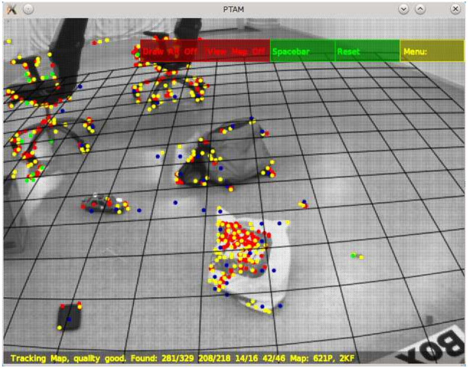
Landing is generally associated with a lack of descent. The observer will explain this with upward winds.

Two interesting states to monitor:

- Altitudinal velocity
- Altitudinal wind velocity

Monocular SLAM

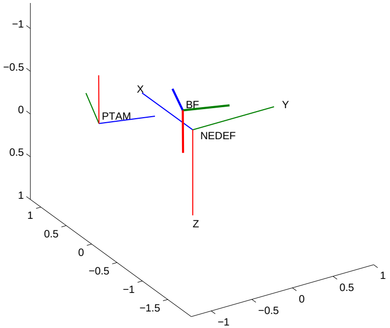
PTAM



- Measurements are not metric,

Camera Measurements

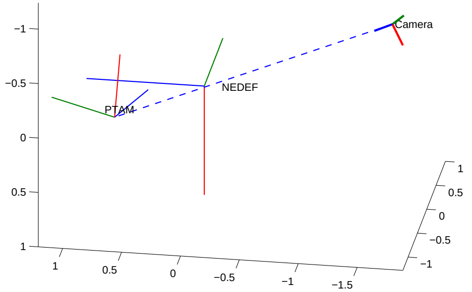
Measurement: Pose relative to PTAM coordinate system.



- Rotation and translation are measured in the local PTAM coordinate system.
- PTAM's coordinate system is quite arbitrarily initialised.
- Its relation to the world must be established

Initialization

PTAM tries to initialize its coordinate system on ground plane.



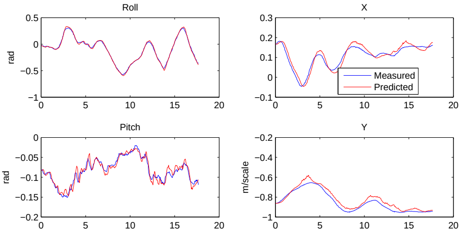
$$\begin{cases} \varnothing_{\text{PTAM}} & = \xi + R(q^{wb}) r_{\text{camera}/\mathcal{G}} + \lambda R(q^{wP}) \frac{X^{PTAM}}{|X^{PTAM}|} \\ \varnothing_{\text{PTAM}} \cdot \hat{z} & = 0 \end{cases}$$

Camera Measurements

To extract positioning measurements usable in the real world, the coordinate systems have to be related.

$$x^{\text{PTAM}} = S(s)R(q^{Pw})T(-\varnothing_{\text{PTAM}})x^{\text{NEDEF}}$$
$$q^{Pw} = q^{PTAM,c}q^{cb}q^{bw}$$

- Describes the measurements in terms of the estimated state and known transformations.



Library Modifications

In a deployment environment, the initialization and utilization of the camera sensor must be autonomous.

In the thesis, several changes are implemented to the PTAM library, e.g.

- Autonomous initialization procedure,
- Re-initialization,
- Origin positioning error detection,
- Remote interface for non-GUI use.

Sensor Fusion

Uses information from all sensors with models of expected behavior to estimate the movements of the vehicle.

The Kalman filter is the standard choice for high-level state estimation.

Non-linear extensions:

- UKF
- EKF

The UKF has issues with the applied motion model in high uncertainty simulations. The EKF was selected as the more reliable filter.

Filtering Framework

The standard formulation of a high-level (Bayesian) state estimation framework relies on two separate steps.

Time update: Use the motion model to predict the vehicle's motions.

Measurement update: Sample the sensors and weigh their likelihood against the current estimate.

Through discrete instances of time, the steps are independent.

Motion Model

For simulation purposes, a non-linear physical model of the quadrotor is studied in the thesis.

- Same model is used for state estimation, control and simulation.
- May easily be replaced by simpler model, e.g. for state estimation.

The model also includes sensor-models for all used sensors.

Linear Quadratic control

Linear Quadratic Control utilizes a linear motion model to optimally control the system.

Definition: Find $u = -Lx$ s.t.

$$\mathcal{J} = \int_0^\infty x^T Q x + u^T R u dt.$$

is minimized. A feedback-form closed solution exists, having solved the CARE[2];

$$A^T S + SA + M^T QM - SBR^{-1} B^T S = 0$$

$$L = R^{-1} B^T S$$

[2]Continous Algebraic Riccati Equation

Linear Quadratic control

- Appealingly simple.
- Requires a linear motion model

The motions of a quadrotor is non-linear, but can locally be described by a linear approximation.

- The motion model needs linearization.
- The linearization needs a linearization point.

State-Dependent Riccati Equation

Basic idea: Linearize the physical model and use LQ theory.

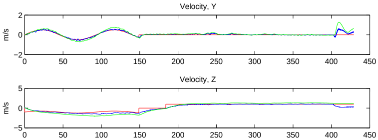
$$\dot{x} = f(x,u) \approx f(x_0,u_0) + \underbrace{\left.\frac{\partial f}{\partial x}\right|_{\substack{x=x_0\\u=u_0}}}_{A} \underbrace{(x-x_0)}_{\Delta x} + \underbrace{\left.\frac{\partial f}{\partial u}\right|_{\substack{x=x_0\\u=u_0}}}_{B} \underbrace{(u-u_0)}_{\Delta u}$$

By adding a homogeneous state, the linear property of the equation is regained. The result is locally valid in every differentiable point in the state space.

$$\dot{X} = \begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} A & f(x_0,u_0) - Ax_0 \\ 0 & 0 \end{bmatrix}}_{\bar{A}} \underbrace{\begin{bmatrix} x \\ 1 \end{bmatrix}}_{\bar{X}} + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{\bar{B}} \Delta u.$$

Conclusions

- General algorithms and control structure were fully implemented.
- PTAM modifications enables full autonomousity.
- Simulated advanced control and landing performed in simulation.



- Implementation covers advanced control.
- Tuning of filtering and control remain.
- Results suggest the system is viable to perform landing.