Sensor Network Support for Real-time Indoor Localization of Four-rotor Flying Robots

Jürgen Eckert, Falko Dressler, Reinhard German

Computer Networks and Communication Systems
Department of Computer Science
University of Erlangen

August 13, 2008

Outline

1. Scenario
   - Description
   - Technical Implementation
2. Indoor Localization
   - Mathematical Approach
   - Communication
3. Measurements
   - Calculation Duration
   - Communication
   - Localisation Accuracy
4. Future Work

Scenario

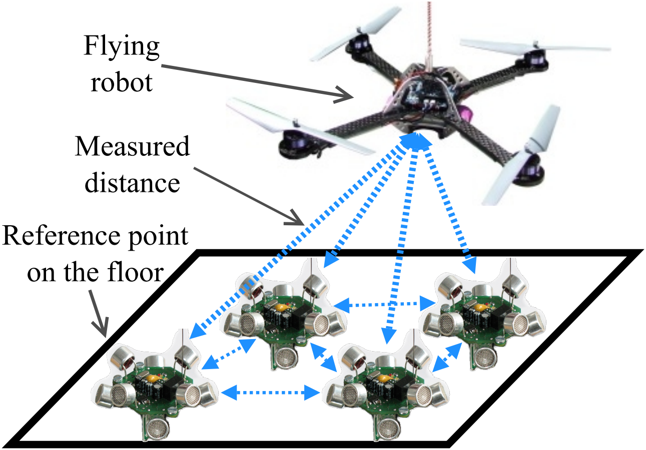
Requirements:

- Autonomous flying four-rotor robot
- Ubiquitous indoor environment
- Zero-configuration system

Approach:

1. Driving robots span a coordinate system by localizing themselves in relation to their neighbors (anchor-free, but global unique)
2. Flying robots can navigate through the environment using the created coordinate system.

Measurements (active-beacon model):



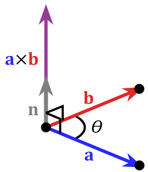
$$T_i = \langle d_i, \overrightarrow{x_i} \rangle; \ \overrightarrow{x_i} = (x_i, y_i, z_i)^T, \ i \in [1, n]$$

What are the necessary steps for position estimation?

1. Triggering and collecting data that represent the correlation between the involved nodes
2. Grouping and evaluating the gained measurements (find "good" pairs)
3. Calculating positions out of the best groups
4. Evaluating the most plausible position out of all positions

Step 2: Subset generation

- A group (subset) consists of three different measurements
- 11 subsets are required if a probability of 1% for selecting only bad subsets (includes one or more incorrect measurements) is accepted (incorrect measurement probability is about 30%)
- Subset evaluation is used to find jitter-insensible groups
- The evaluation is based on the mean distance and the cross product → Fast qualifier for "well-formed" tetrahedrons



Procedure:
Randomly picking pairwise dissimilar measurements; evaluate and insert them in a sorted list until a certain threshold is fulfilled.

Step 3: Position calculations

$$(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = d_i^2; \; i \in [1, n] \qquad (1)$$

Iterative solution approaches:

- Taylor series
- Extended Kalman filter
- Bancroft method
- ...

But a closed mathematical solution is faster and less memory consuming. Target: very low cost and power micro-controller using only basic mathematical operations.

Requirements:

$$\overrightarrow{x}_i \in \{S_j \mid z_i = c_j\};\ S_j \subseteq T;\ \|S_j\| = 3; c_j \in \Re \qquad (2)$$

$$z \geq c_j \text{ or } z \leq c_j \qquad (3)$$

Solution:

$$A\overrightarrow{x} = \overrightarrow{b}; \quad A \in \Re^{2\times 2};\ \overrightarrow{x} \in \Re^2;\ \overrightarrow{b} \in \Re^2 \qquad (4)$$

$$A = 2 \cdot \begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix} \qquad (5)$$

$$\overrightarrow{x} = (x, y)^T \qquad (6)$$

$$\overrightarrow{b} = \begin{pmatrix} (d_1^2 - d_3^2) + (x_3^2 - x_1^2) + (y_3^2 - y_1^2) + (z_3^2 - z_1^2) \\ (d_2^2 - d_3^2) + (x_3^2 - x_2^2) + (y_3^2 - y_2^2) + (z_3^2 - z_2^2) \end{pmatrix} \qquad (7)$$

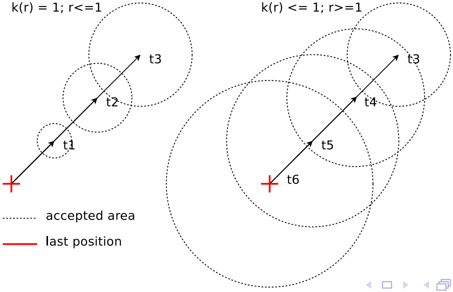
z-coordinate $\rightarrow$ (1)

Step 4: Position estimation

Kalman statement

1. **Prediction** using the state vector
2. **Correction** using the current (incorrect) measurements

Complex system dependent equations at design and run time.
Step 2 is exchanged by: "take the most plausible position".



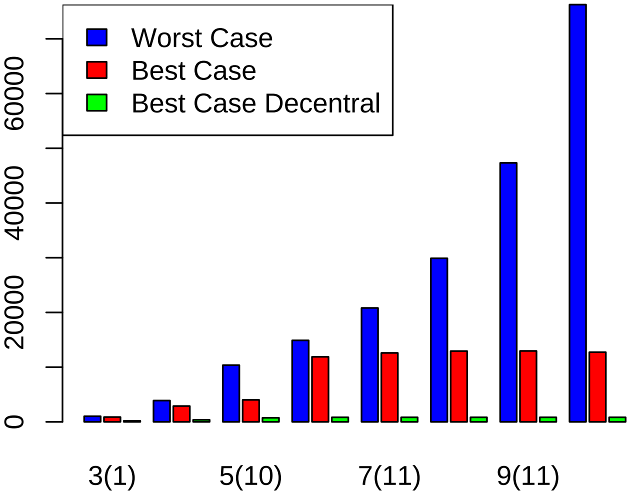
Communication

SunSpot from Sun Microsystems:
IEEE 802.15.4 radio; CSMA-CA;

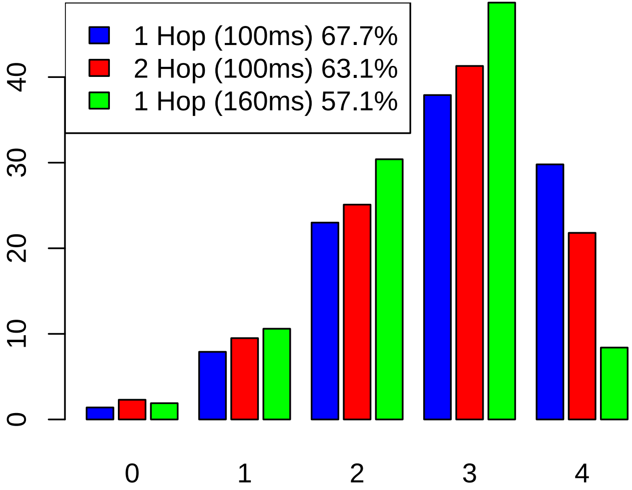
- Many collisions during the data collection phase due to the synchronization of the sensor nodes
- ALOHA based user level agent protocol
- The agent is hopping over the sensor network and collects the data while it is demerging and merging itself

Demerge: A frame is broadcast and more than one node received it.
Merge: Two different frames with the same ID arrive on a node. A union of all information will be build and may be send out.

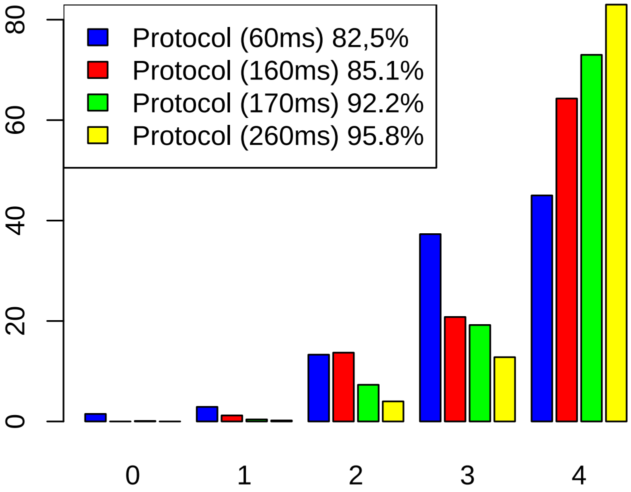
Calculation Duration (in μS)



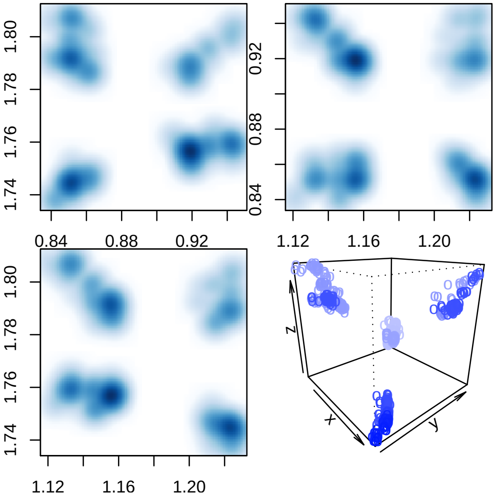
Availability measurements for the simple broadcast case



Availability measurements for the agent frame case



Localization accuracy with a fixed quadrocopter



Future work

Autonomous...

- **placement** and **anchor-free self-localization** of the ground nodes (near)
- **position holding** using the localization system (mid)
- **flight path** planing (far)

Remaining issues:
Globally valid heading hold mechanism for the flying robots are not (yet) available

Note:
Autonomous flying (especially none-line-of-sight) is actually not allowed in germany. A human controlled kill-switch must be available. (legitimate gray area)

Thanks for your attention!

Questions?