ETH zürich

Digital Circuits and Systems Group

Enabling UAV-based Localization Leveraging the UWB Technology: From Theory to Practice

Vlad Niculescu

October 10, 2024

The Rise of UAVs: Pervasive Applications Across Industries

- High interest in unmanned aerial vehicles (UAVs)
- Civil and military applications

Rescue missions [2]

Agriculture [1]

Entertainment [3]

20 Billion $ market

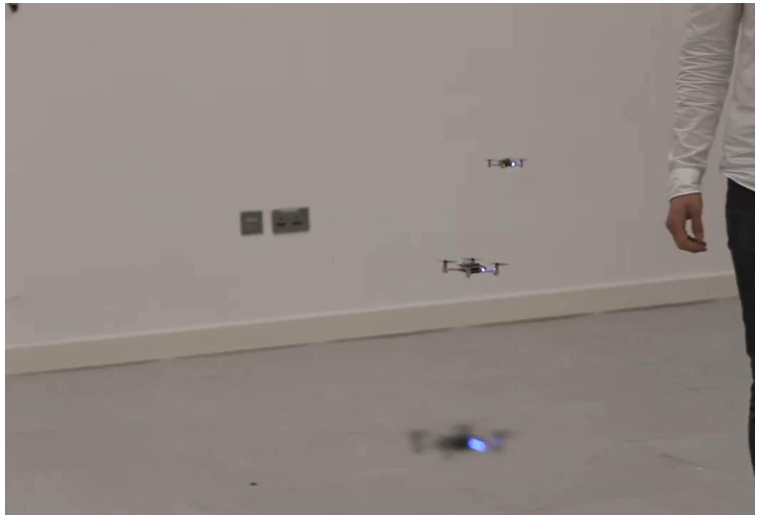
[1] https://iotechworld.com/application-of-drones-in-indian-agriculture
[2] https://www.swissinfo.ch/
[3] Firefly Drone Shows Promo Video

UAV Classification: from Standard to Nano

- High interest in unmanned aerial vehicles (UAVs)
- Civil and military applications
- My work focuses on **nano-size UAVs**
- **Localization** is necessary

Category	Weight [g:cm]	Power [W]	Computation & Sensing
Standard-size	$\geq 1000 : \sim 50$	≥ 100	> 10 W (Intel NUC + LiDAR)
Micro-size	$\sim 500 : \sim 25$	~ 50	~ 5 W (Nvidia TX2 + Camera)
Nano-size	$\leq 50 : \sim 10$	~ 10	< 1 W (ARM Cortex M4 + Low-power camera)

Safe near-human navigation



Enhance scalability

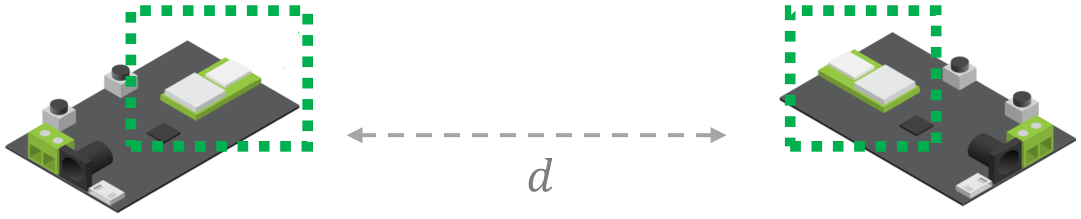
Can reach confined spaces

[1]: https://www.bitcraze.io/
[2]: https://digital.floribusiness.com/
[3]: https://www.upi.com/
[4]: https://arxiv.org/abs/1909.11236

UWB: Introduction

- UWB gained momentum in the last 10 years
- Novel technology for:
  - Centimeter-precision ranging
  - Data transmission @ 6.8 Mbps
- Requires at least two transceivers
- Qorvo DW1000 often used in research
  - Ranging @ 400 Hz
  - Power consumption 300 mW

UWB Ranging
(node-to-node distance)

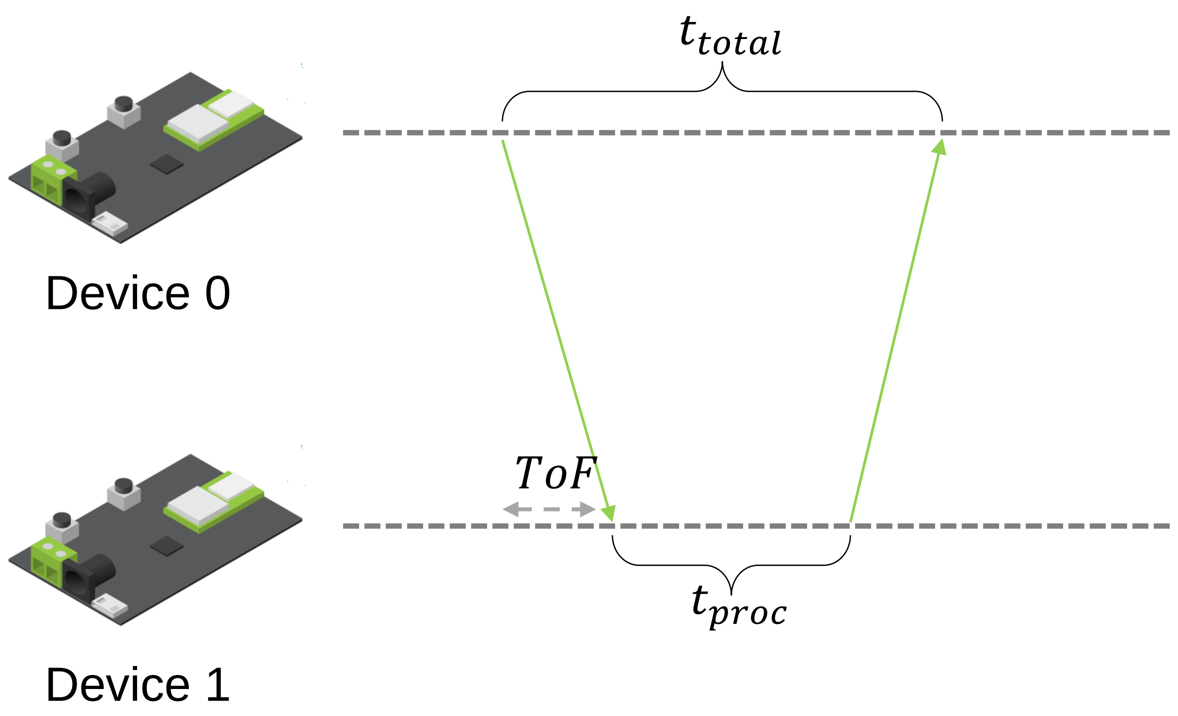


UWB: Ranging schemes

- Single-sided Two-Way Ranging (TWR)
  - Device 0 sends a UWB message
  - Device 1 receives the message
  - Device 1 acknowledges
  - Device 0 receives the response
  - Device 0 computes the Time of Flight (ToF)

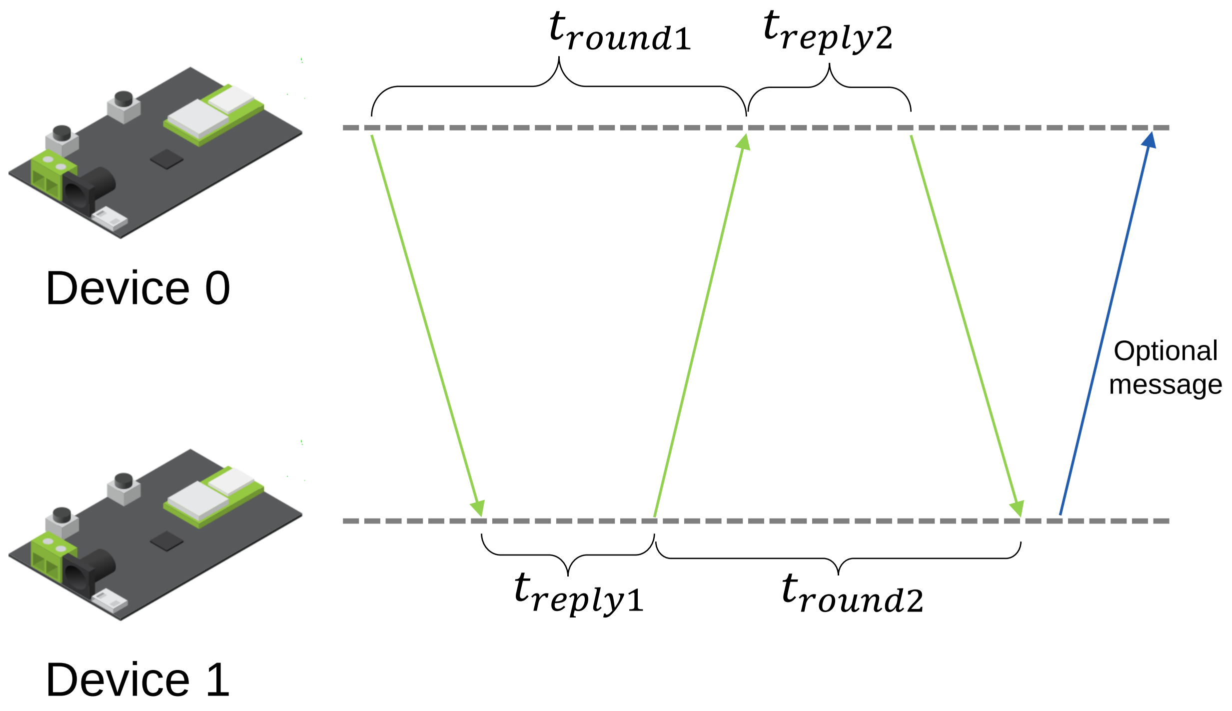
$$ToF = \frac{t_{total} - t_{proc}}{2}$$

$$d = c \cdot ToF$$



UWB: Ranging schemes

- Double-sided Two-Way Ranging (TWR)
  - Involves an additional message
  - Less sensitive to clock drift
  - Requires a fourt message to communicate the distance to Device 0

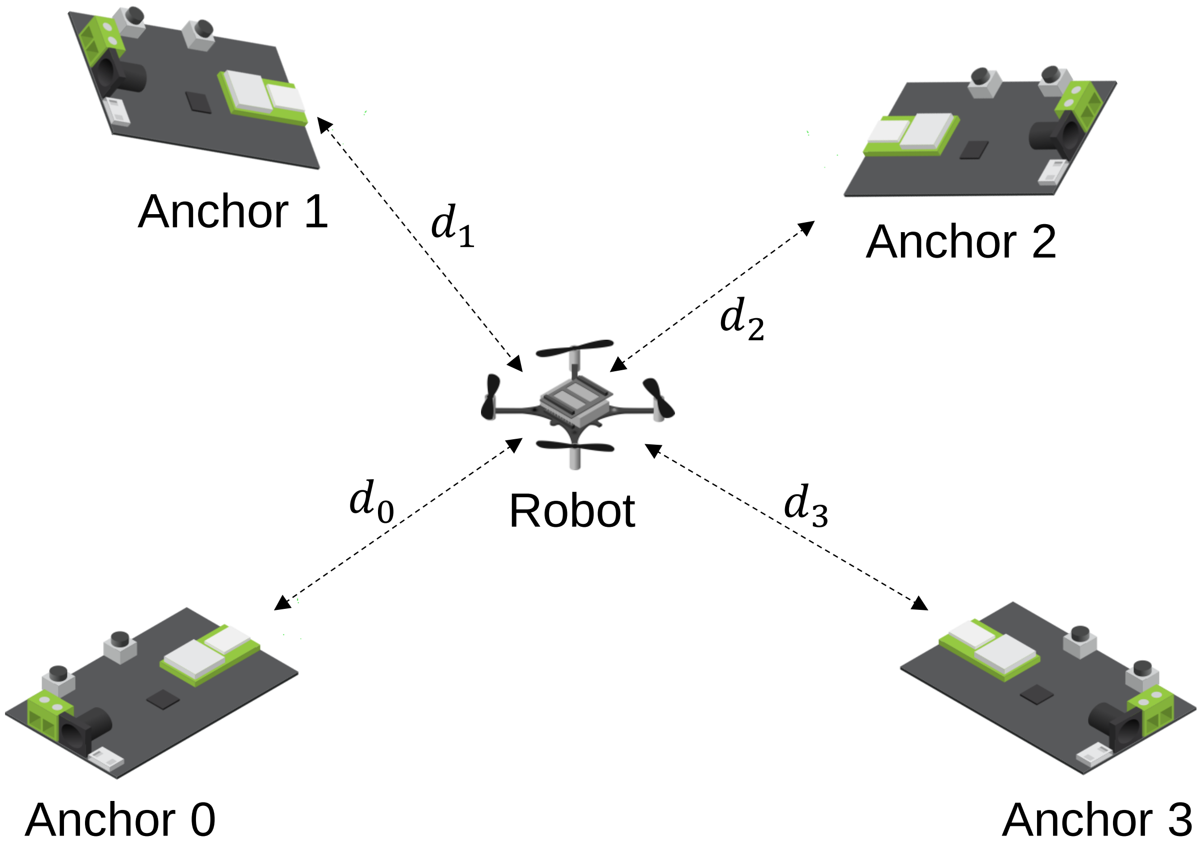


$$ToF = \frac{t_{round1} \cdot t_{round2} - t_{reply1} \cdot t_{reply2}}{t_{round1} + t_{round2} + t_{reply1} + t_{reply2}}$$

$$d = c \cdot ToF$$

From Ranging to Localization

- Fixed points with known positions – Anchors
- Robot to be localized
- Continous ranging between Robot and Anchors → Robot's position



Range Measurements → Robot Localization

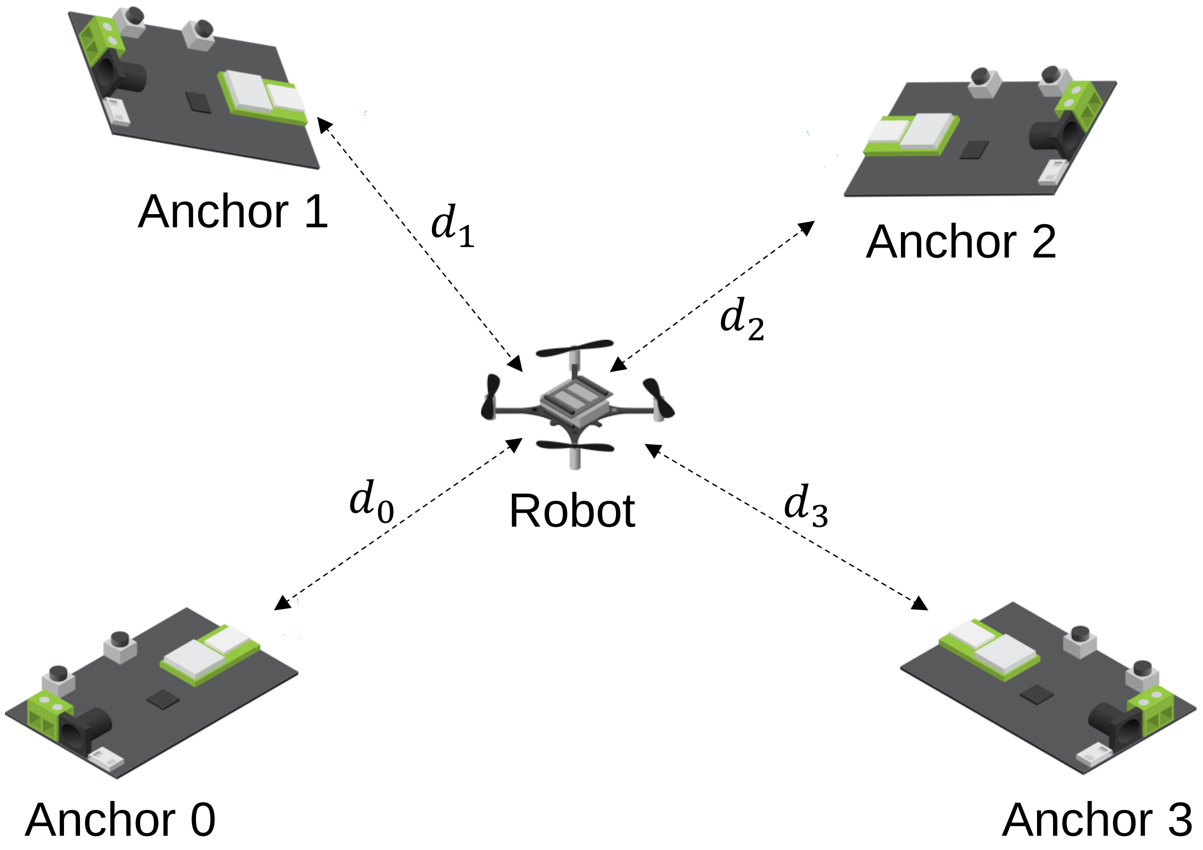
Two Main Methods

Filtering-based Localization:

- Typically employing Kalman Filters

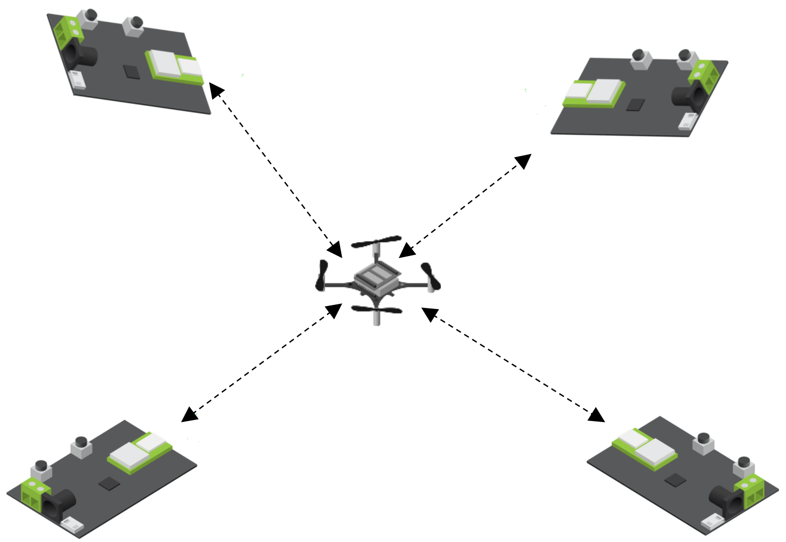
Optimization-based Localization:

- Typically employing Trilateration
- Minimizes a cost function



Kalman Filter for Localization

- Kalman Filter offers a convenient way to incorporate distance measurements
- Fully recursive
- Measurements incorporated one by one



Distances $d_0,\, d_1,\, d_2,\, d_3$ → Kalman Filter → Position

$$\boldsymbol{x} = \begin{pmatrix} x \\ y \end{pmatrix}$$

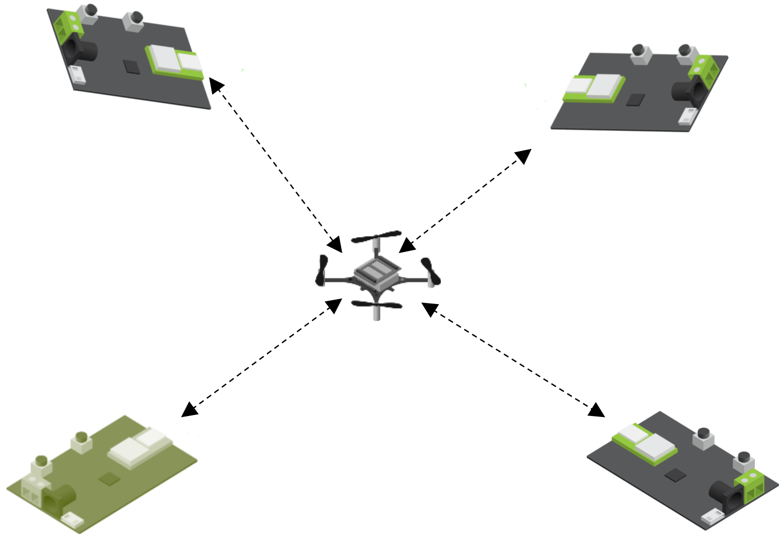
$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$$

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + K(d_i - H\, x_p)$$

Kalman Filter for Localization

- Kalman Filter offers a convenient way to incorporate distance measurements
- Fully recursive
- Measurements incorporated one by one

Distances $\cancel{d_0}$ → **Kalman Filter** d_0 → Position $(0.1, 0.2)$



Anchor 0

Kalman Filter for Localization

- Kalman Filter offers a convenient way to incorporate distance measurements
- Fully recursive
- Measurements incorporated one by one

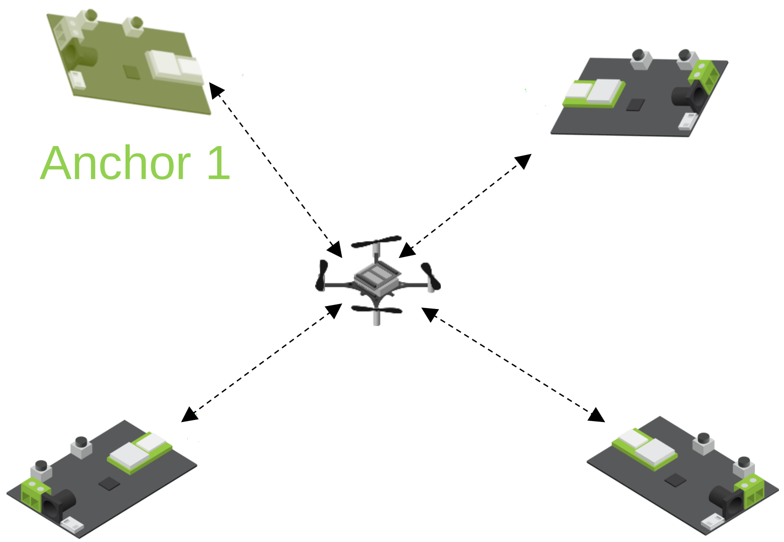
Distances

$\cancel{d_1}$

→ **Kalman Filter** d_1 →

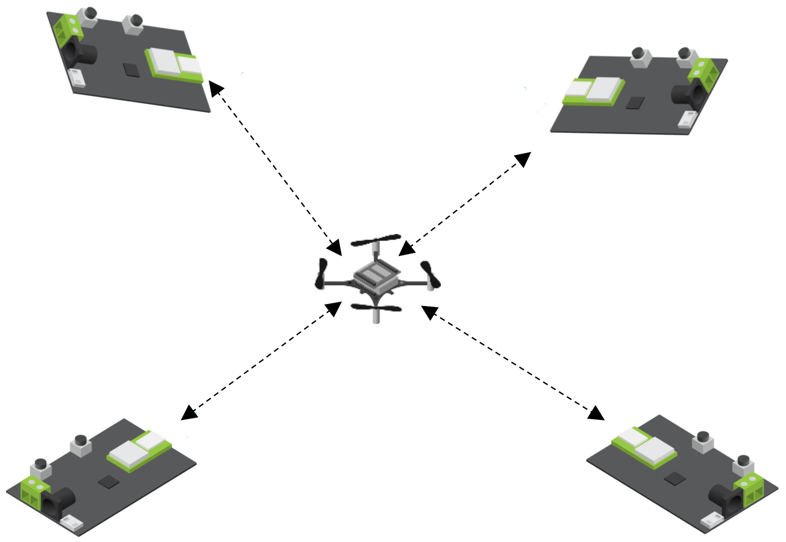
Position

$(0.15, 0.25)$



Kalman Filter for Localization

- Kalman Filter can incorporate multiple various sensor information



State

Accelerometer

Distances

Gyroscope

Kalman Filter

Attitude

Position

Velocity

$$acceleration = f_0(state)$$

$$angular\ velocity = f_1(state)$$

$$distance = f_2(state)$$

Localization Methods Comparison

Kalman Filters

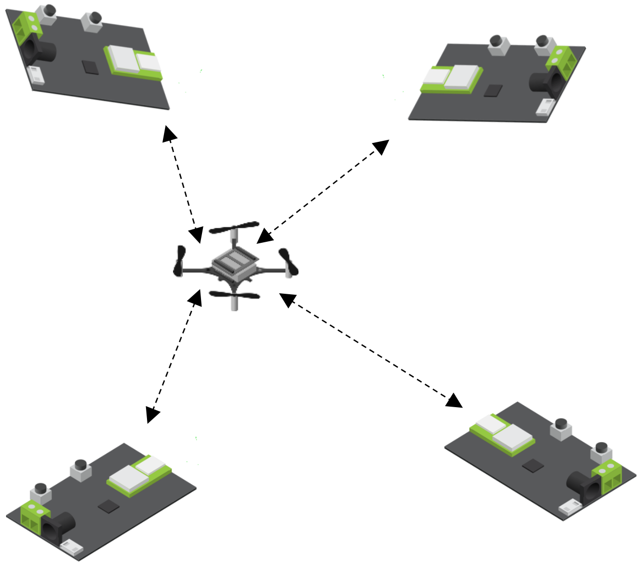
$d_0 \rightarrow$ Kalman Filter $\rightarrow$ State

- Discard measurements after update
- Recursive: computation efficient
- Memory efficient
- Can fuse multiple sensor information
- Work poorly for large non-linearities

Optimization-based methods

$\left.\begin{matrix} d_0 \\ d_1 \\ d_2 \\ \vdots \end{matrix}\right\} \rightarrow$ Cost function $\rightarrow$ State

- More effectve for small number of measurements
- Unlike KFs, it does not have to converge
- Store the whole measurement history




Indoor Flying

Hardware Used for UWB

DWM1000

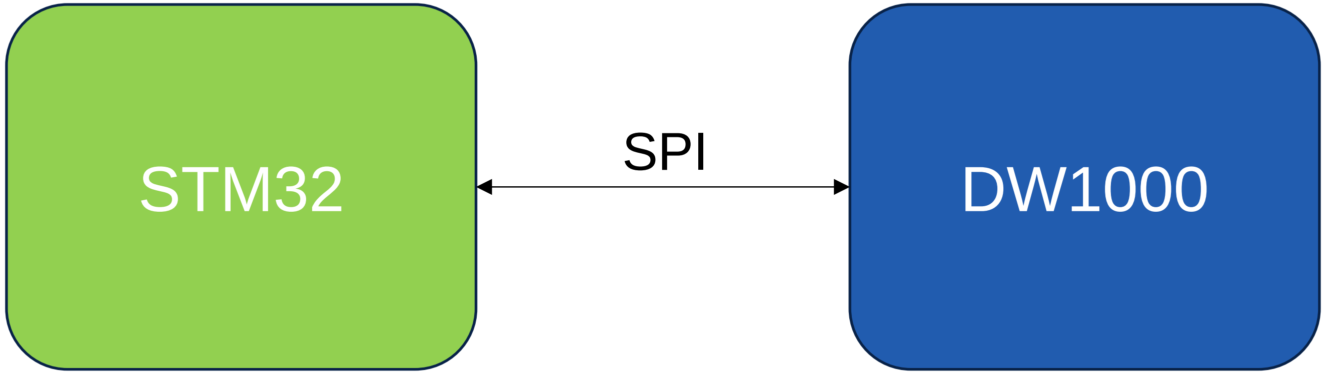
Loco Positioning Deck

Crazyflie 2.1

<10 cm

~30 g

Open source



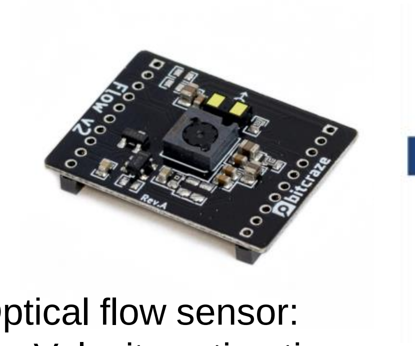
Library - including code examples

https://github.com/vladniculescu/uwb-software-library

Hardware Platform

- Commercial drone platform Crazyflie v2.1 from Bitcraze
- Additional hardware: Flow Deck v2 and Loco Positioning Deck
- Payload: 1.6 g + 3.3 g → 4.9 g
- Flight Time: ~ 7 min

Flow Deck v2



Optical flow sensor:
- Velocity estimation
- Position estimation (by integration)

Crazyflie v2.1

Loco Positioning Deck

UWB module

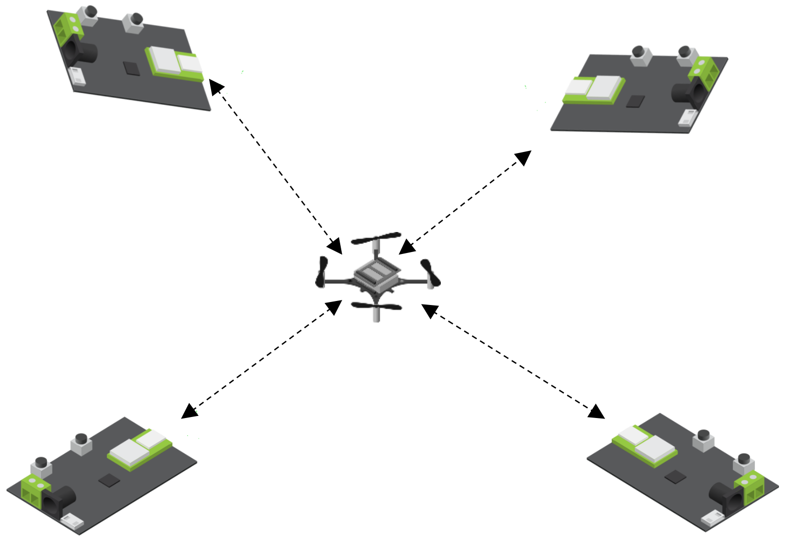
Introducing the Dual Problem

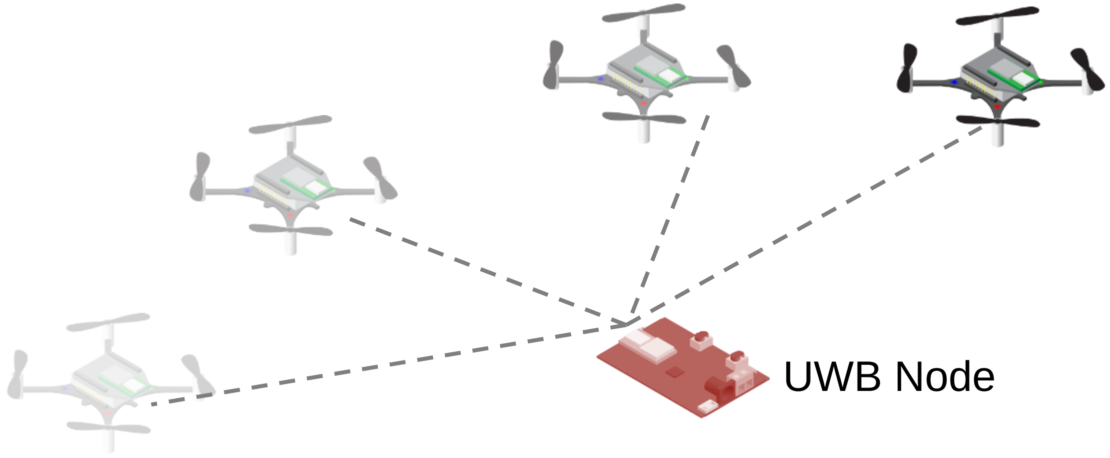
Classical localization problem:

- Fixed anchors with know positions
- The anchors enable the robot to localize

The dual problem:

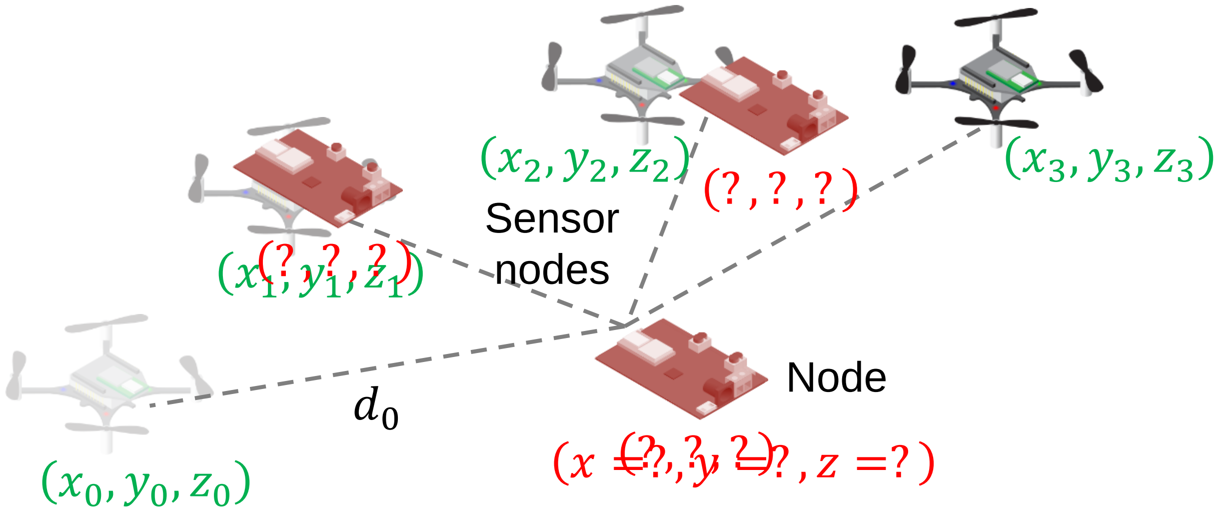
- Drone can determine its own position
- Drone acts as moving anchor to localize fixed UWB nodes of unknown positions





Range-Based Localization for Navigating Sensor Networks

- In sensor networks, nodes’ positions are often not known
- Precise localization is necessary
- Use case: UAV reaching sensor nodes
- Precise localization is necessary
- Onboard odometry → drone’s position



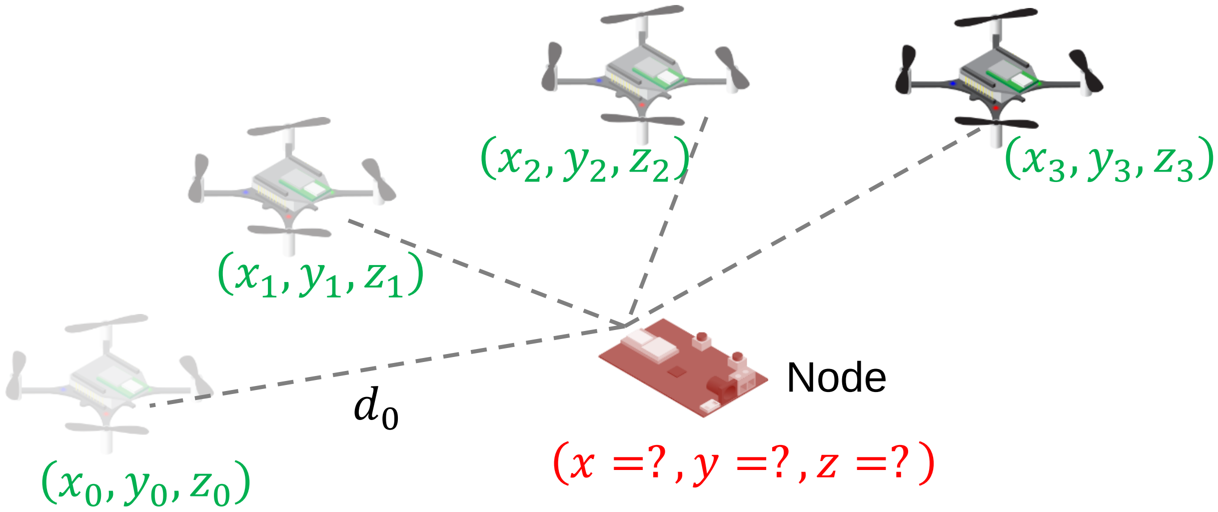
Trilateration algorithm

$$\left.\begin{array}{l}(x_0, y_0, z_0)\\(x_1, y_1, z_1)\\(x_2, y_2, z_2)\\(x_3, y_3, z_3)\\d_0, d_1, d_2, d_3\end{array}\right\}\xrightarrow{\text{Trilateration}}(x, y, z)$$

≥ 4 measurements

Range-Based Localization for Navigating Sensor Networks

- In sensor networks, nodes' positions are often not known
- Precise localization is necessary
- Use case: UAV reaching sensor nodes
- Precise localization is necessary
- Onboard odometry → drone's position
- At least 4 measurements for 3D localization



Trilateration algorithm

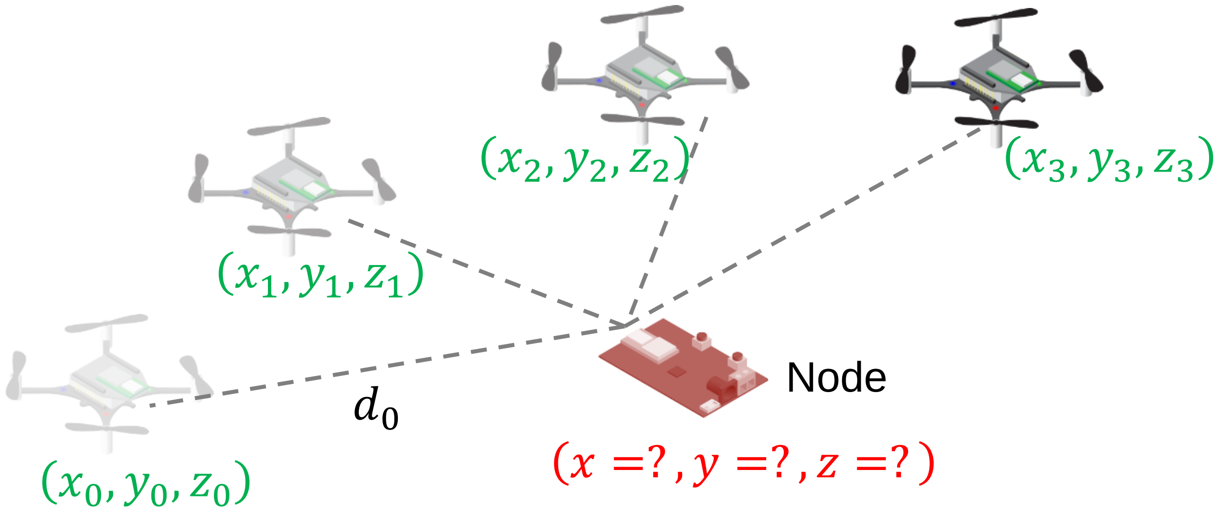
$$e_i = \underbrace{\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}}_{\text{calculated error}} - d_i$$

$$L(x, y, z) = \sum_i e_i^2$$

$$(\tilde{x}, \tilde{y}, \tilde{z}) = \underset{x,y,z}{\operatorname{argmin}}\, L(x, y, z)$$

Range-Based Localization for Navigating Sensor Networks

- In sensor networks, nodes' positions are often not known
- Precise localization is necessary
- Use case: UAV reaching sensor nodes
- Precise localization is necessary
- Onboard odometry → drone's position
- At least 4 measurements for 3D localization



M. Larsson et al., "Optimal Trilateration Is an Eigenvalue Problem," ICASSP 2019

Optimal Trilateration

Trilateration algorithm
(modified cost function)

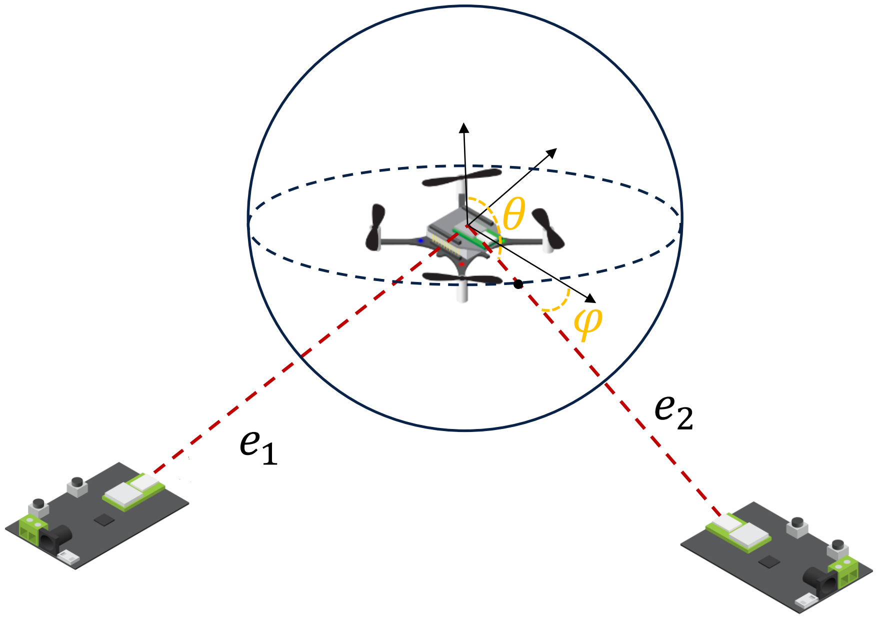
$$e_i = \underbrace{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}_{\text{calculated error}} - d_i^2$$

$$L(x, y, z) = \sum_i e_i^2$$

$$(\tilde{x}, \tilde{y}, \tilde{z}) = \underset{x,y,z}{\operatorname{argmin}} \, L(x, y, z)$$

Uneven UWB Antenna Gain Influences the Localization Error

- Errors depend on direction: azimuth (φ) and polar (θ) angles
- UWB error model $e_{UWB} = f(\theta, \varphi)$
- SoA [1] uses a neural network for the error model



[1]: W. Zhao et al., "Learning-based bias correction for ultra-wideband localization of resource-constrained mobile robots," RAL 2021

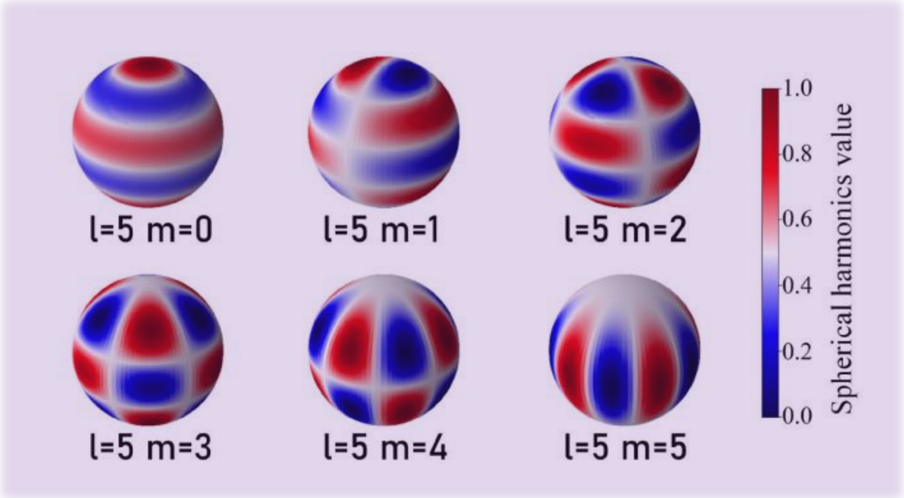
Uneven UWB Antenna Gain Influences the Localization Error

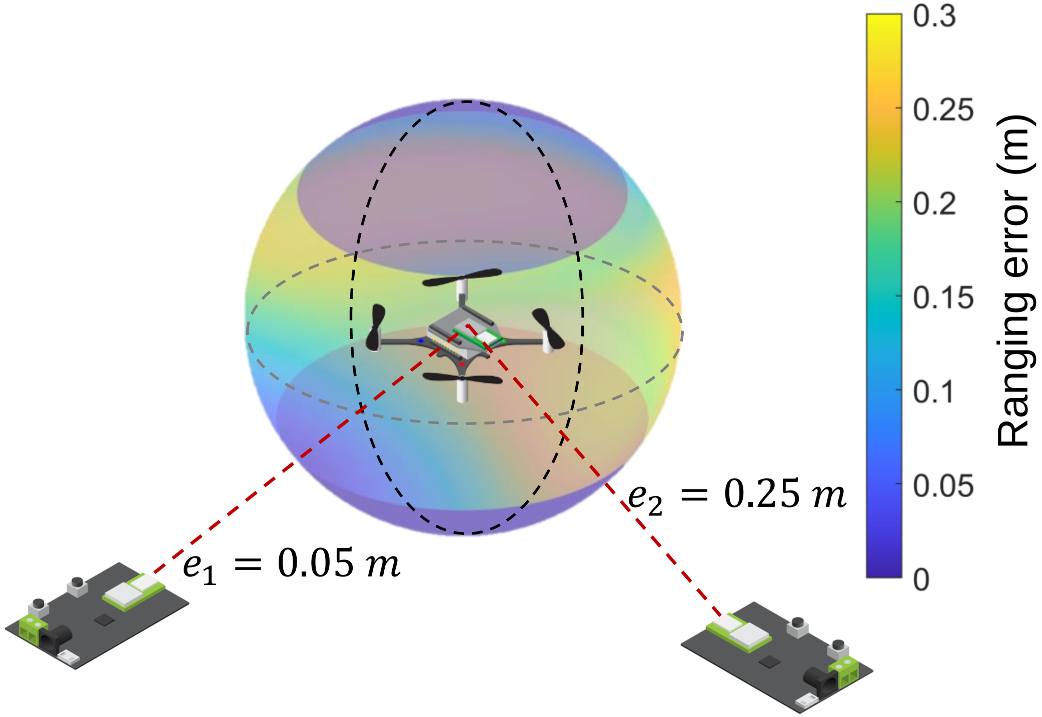
- Errors depend on direction: azimuth (φ) and polar (θ) angles
- UWB error model $e_{UWB} = f(\theta, \varphi)$
- SoA [1] uses a neural network for the error model

Our solution:

- Spherical harmonics $Y_i(\theta, \varphi)$ used as feature mapping

$$f(\theta, \varphi) = \sum_{i=0}^{N} c_i Y_i(\theta, \varphi)$$





[1]: W. Zhao et al., "Learning-based bias correction for ultra-wideband localization of resource-constrained mobile robots," RAL 2021

Uneven UWB Antenna Gain Influences the Localization Error

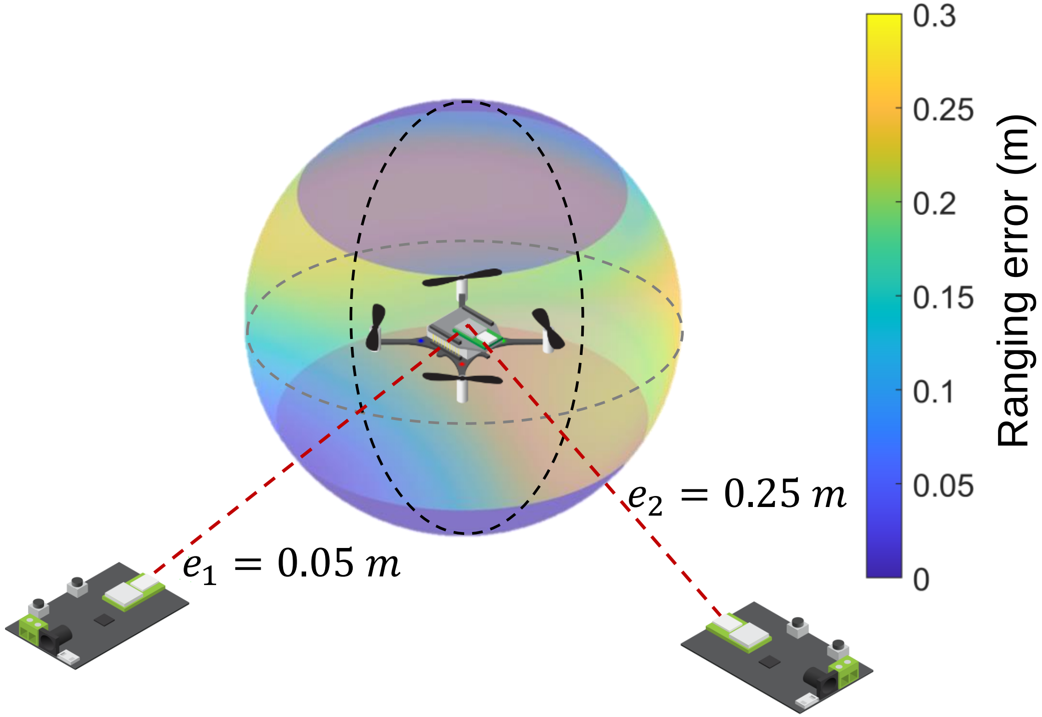
- Errors depend on direction: azimuth (φ) and polar (θ) angles
- UWB error model $e_{UWB} = f(\theta, \varphi)$
- SoA [1] uses a neural network for the error model

Our solution:

- Spherical harmonics $Y_i(\theta, \varphi)$ used as feature mapping

$$f(\theta, \varphi) = \sum_{i=0}^{N} c_i Y_i(\theta, \varphi)$$ trained as *least squares*

- 3X error reduction



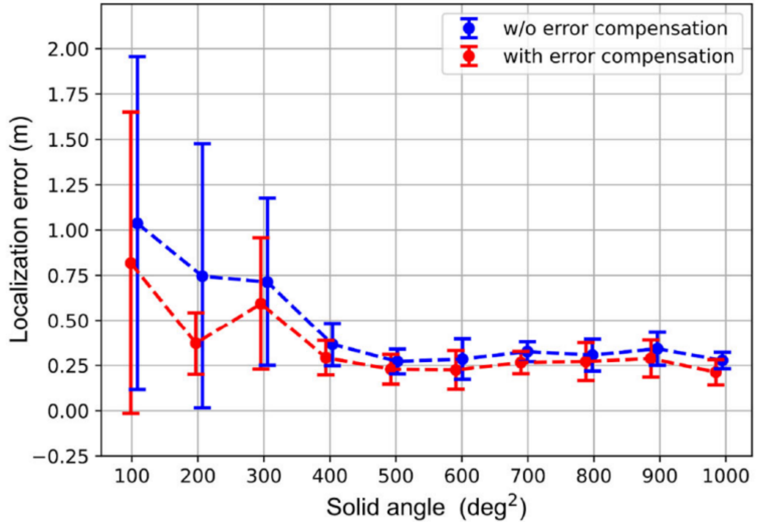
Number of harmonics	0	1	4	9	16	25	36
Parameters	0	2	20	90	272	650	1332
Ranging error (m)	0.22	0.12	0.102	0.08	0.071	0.067	0.064

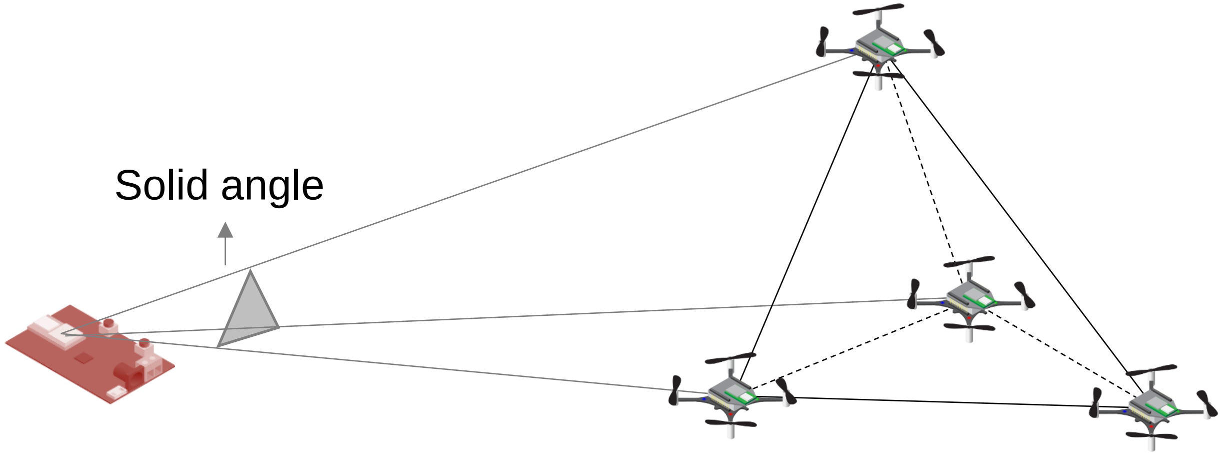
[1]: W. Zhao et al., “Learning-based bias correction for ultra-wideband localization of resource-constrained mobile robots,” RAL 2021

Range-Based Localization: In-field Evaluation

- Localization error vs solid angle
- Best localization error:
  - without error modelling 28 cm
  - with error modelling 21 cm
- Localization runs in < 2 ms onboard the STM32 @ 168 MHz

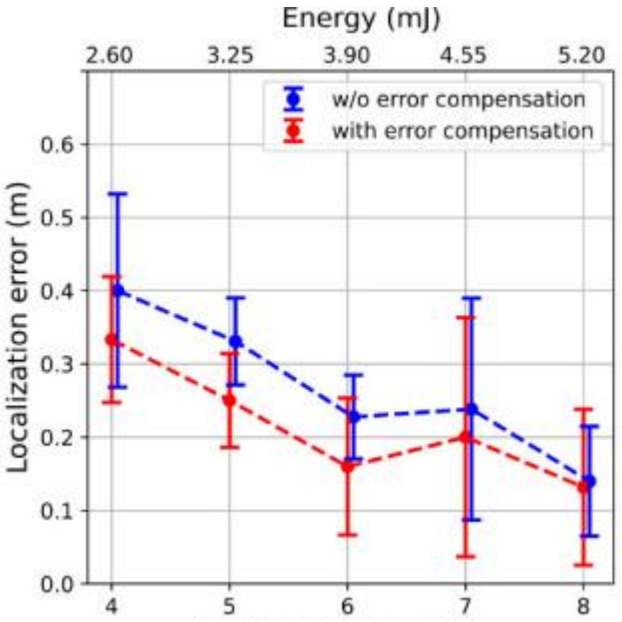
Closed-loop operation





Range-Based Localization: In-field Evaluation

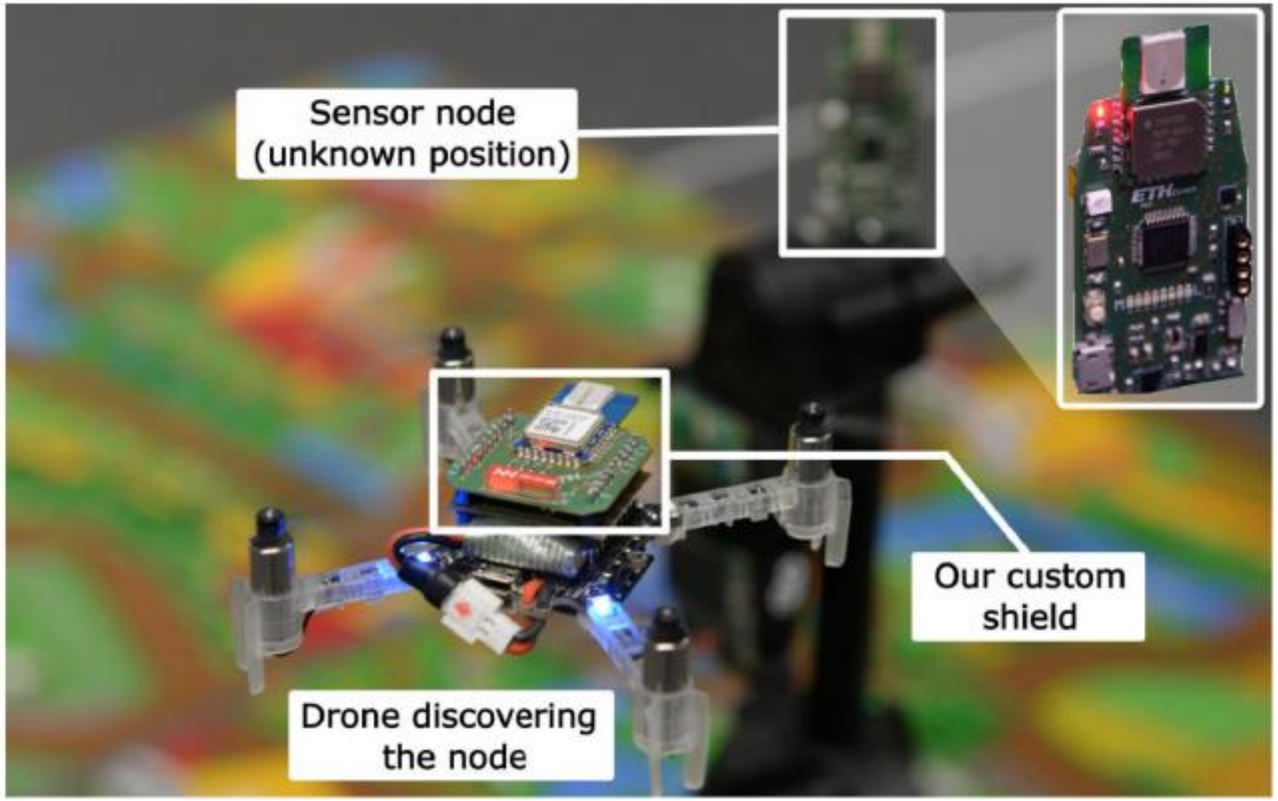
- Localization error vs solid angle
- Best localization error:
  - without error modelling 28 cm
  - with error modelling 21 cm
- Localization runs in < 2 ms onboard the STM32 @ 168 MHz



Number of UWB measurements

Range-Based Localization: In-field Evaluation

- Localization error vs solid angle
- Best localization error:
  - without error modelling 28 cm
  - with error modelling 21 cm
- Localization runs in < 2 ms onboard the STM32 @ 168 MHz



V. Niculescu, D. Palossi, M. Magno, and L. Benini, "Energy-efficient, Precise UWB-based 3-D Localization of Sensor Nodes with a Nano-UAV," **IEEE Internet of Things Journal**, 2023.

Energy-efficient, Precise UWB-based 3-D Localization of Sensor Nodes with a Nano-UAV

Vlad Niculescu[*], Daniele Palossi[*‡], Michele Magno[*], Luca Benini[*†]

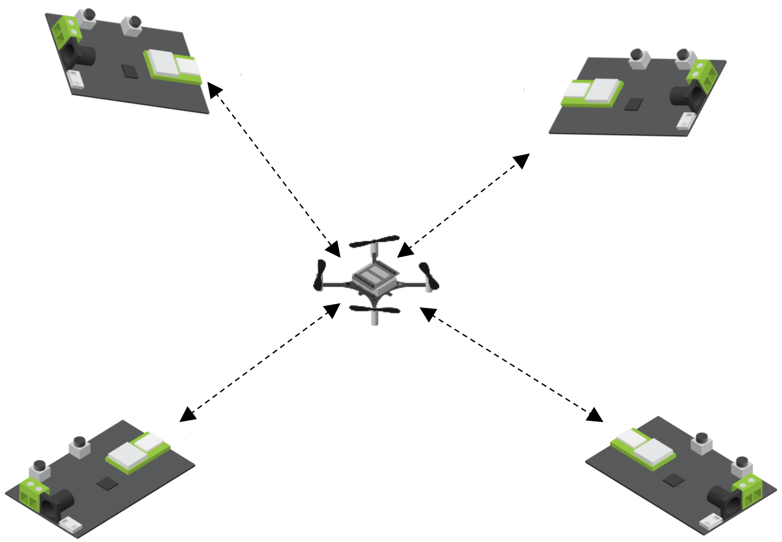
[*] *Integrated Systems Laboratory - ETH Zurich, Switzerland*

[†] *Department of Electrical, Electronic and Information Engineering - University of Bologna, Italy*

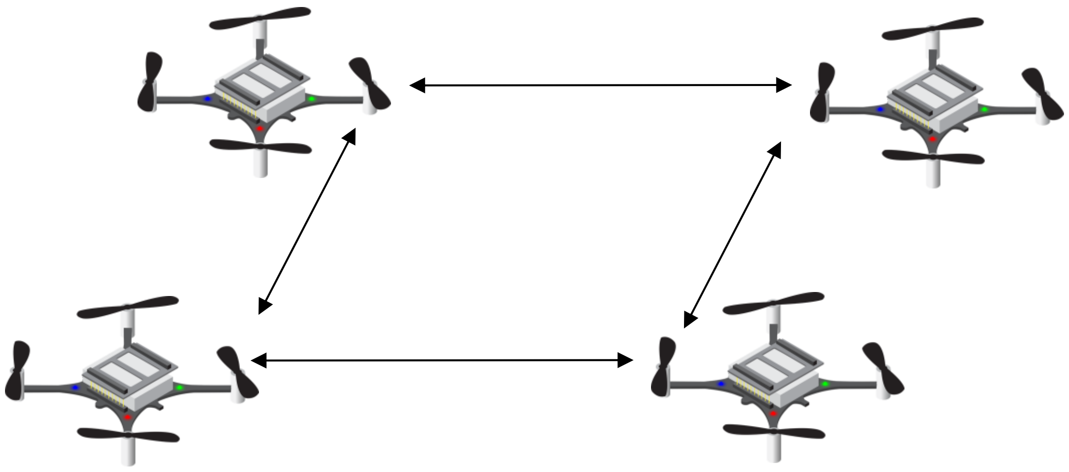
[‡] *Dalle Molle Institute for Artificial Intelligence - University of Lugano and SUPSI, Switzerland*

Supplementary video

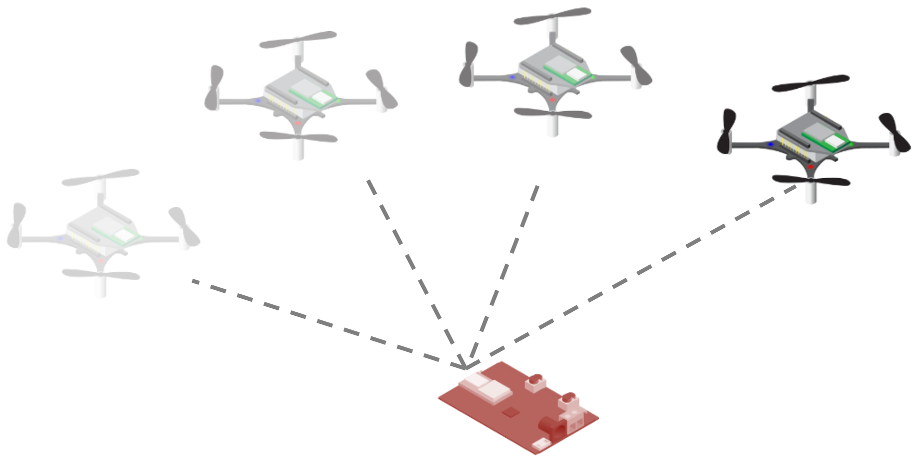
Going Beyond Anchors



Anchor-based localization



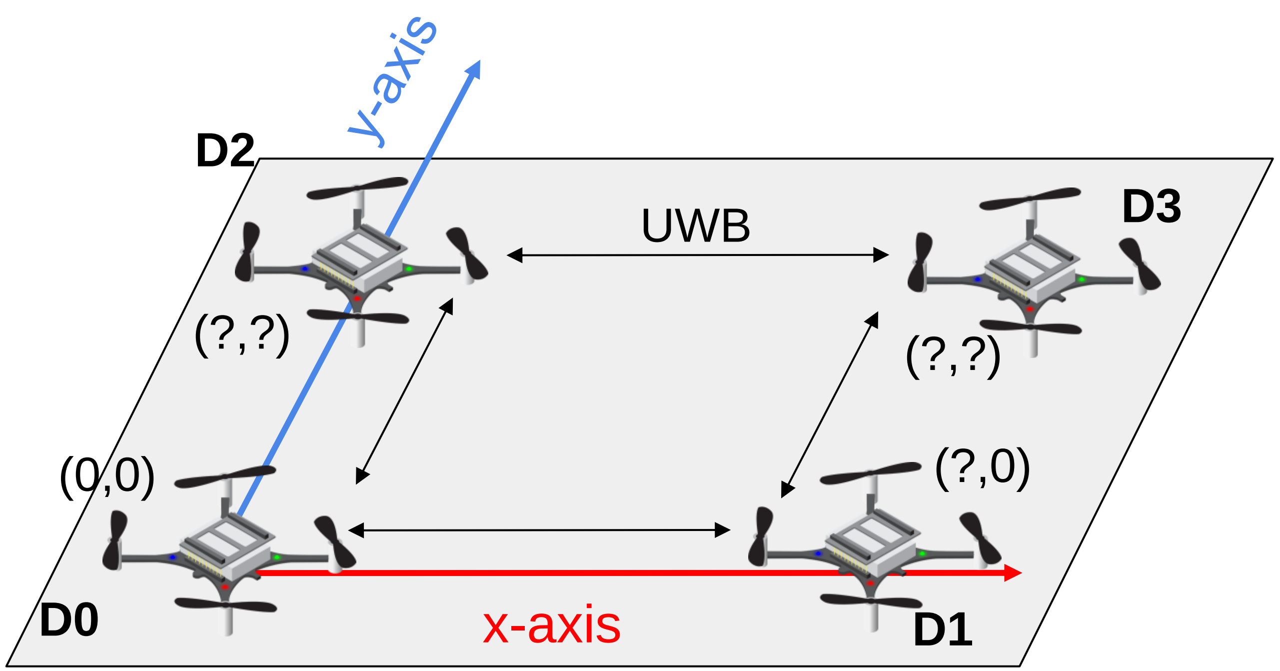
Relative localization



Dropping the fixed anchors

Concept of Relative Localization

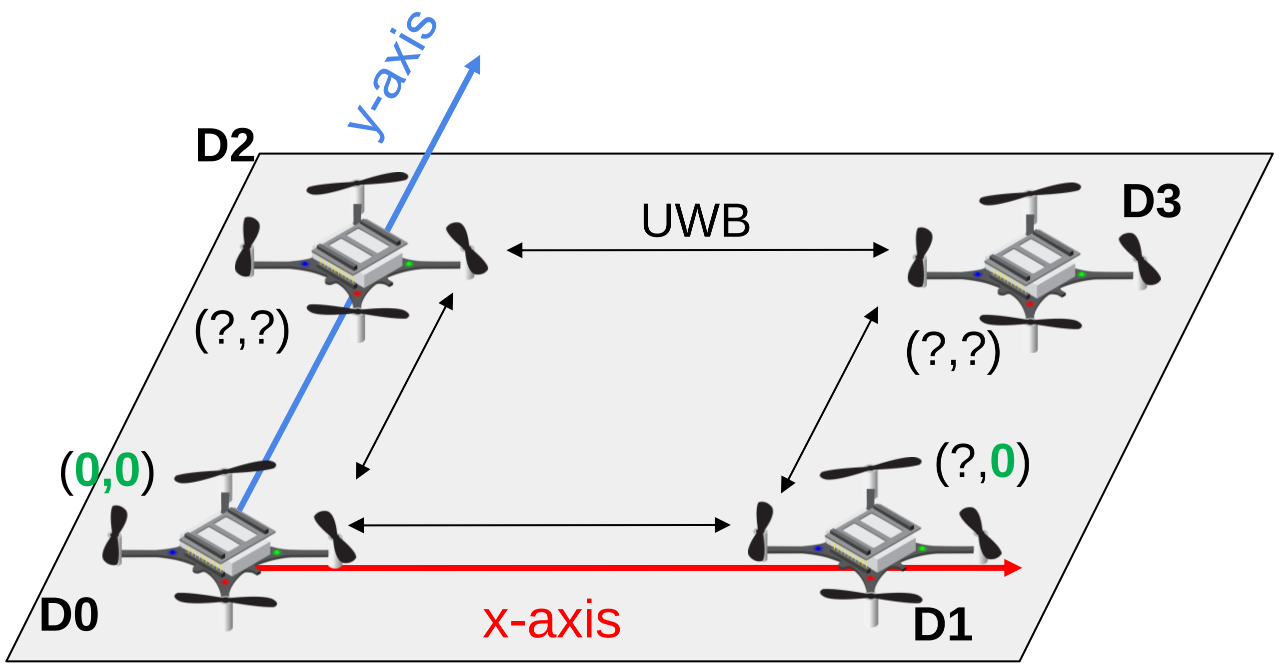
- Positions of the anchor-drones (ADs) are a priori unknown.
- The drones exchange UWB range measurements between each other.
- Measurements used to compute the relative coordinates.
- Convention:
  - D0 is the origin of the coordinate system
  - D0 - D1 the OX axis: x-coordinate of D1 is always positive
  - y-coordinate of D2 is always positive
- Algorithm:
  - Multidimensional Scaling (MDS) [1]



[1] Nhat, Vo Dinh Minh, et al. "Nonmetric MDS for sensor localization." 2008 3rd International Symposium on Wireless Pervasive Computing. IEEE, 2008.

Concept of Relative Localization

- Positions of the anchor-drones (ADs) are a priori unknown.
- The drones exchange UWB range measurements between each other.
- Measurements used to compute the relative coordinates.
- Convention:
  - D0 is the origin of the coordinate system
  - D0 - D1 the OX axis: x-coordinate of D1 is always positive
  - y-coordinate of D2 is always positive
- Algorithm:
  - Multidimensional Scaling (MDS) [1]



[1] Nhat, Vo Dinh Minh, et al. "Nonmetric MDS for sensor localization." 2008 3rd International Symposium on Wireless Pervasive Computing. IEEE, 2008.

Multidimensional Scaling

Multidimensional scaling (MDS):

- We note the coordinate vector of one drone as

$$X_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

- The squared distance between two anchors is

$$d_{i,j}^2 = \|X_i\|^2 + \|X_j\|^2 - 2 \underbrace{< X_i, X_j >}_{\text{Dot product}} \xrightarrow{\textit{Matrix form}}$$

- The distance matrix:

$$D := \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,M} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,M} \\ \vdots & \vdots & & \vdots \\ d_{M,1} & d_{M,2} & \cdots & d_{M,M} \end{pmatrix} \qquad D^{(2)} = [d_{i,j}^2]_{i,j=1}^N$$

- We use the following notations:

$$P = [X_1^T X_1, X_2^T X_2, \cdots X_M^T X_M]$$

$$X = [X_1, X_2, \cdots, X_M]$$

$$D^{(2)} = P\mathbf{1}^T - 2\underbrace{\boxed{X^T X}}_{\text{We aim to calculate this}} + \mathbf{1}P^T$$

↓

Symmetrical, pos. def.

$$\boxed{X^T X} = -\frac{1}{2} H D^{(2)} H$$

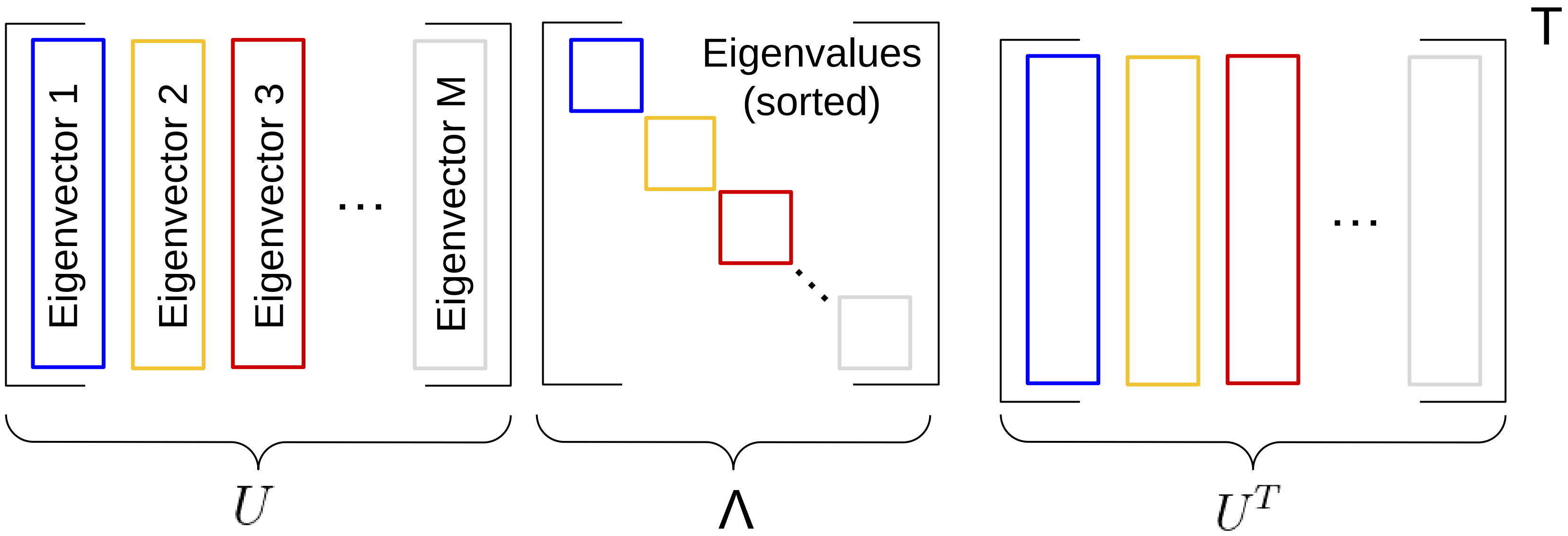
$$H = I - \frac{1}{M}\mathbf{1}\mathbf{1}^T$$

↓

$$X^T X = U \Lambda U^T$$

Multidimensional Scaling

$$X^T X = U \Lambda U^T \longrightarrow X = U \Lambda^{\frac{1}{2}}$$

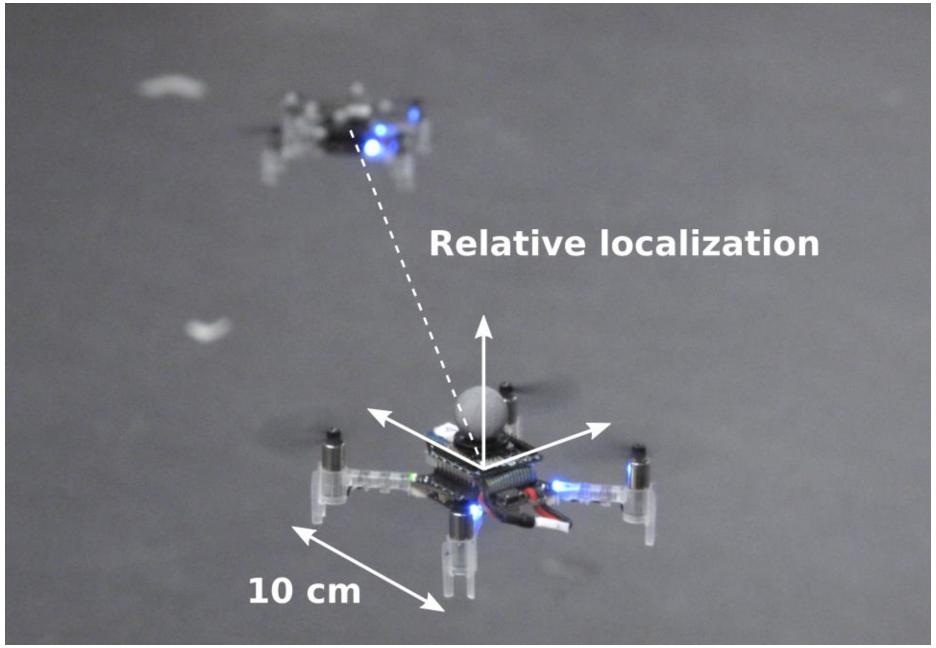


Multidimensional scaling works for static formations

What about dynamic relative localization?

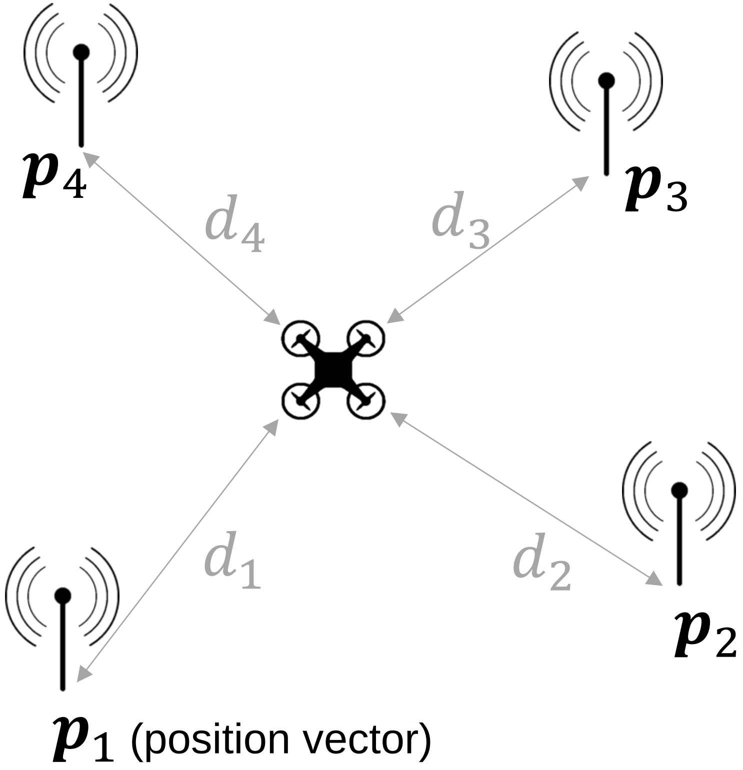
Pair-wise Relative Localization

- Relative localization for swarm formation
- Uses inter-drone UWB distance measurements
- Pairwise: two drones at a time
- No initial values required
- Requires velocity measurements



Classical Trilateration Revisited

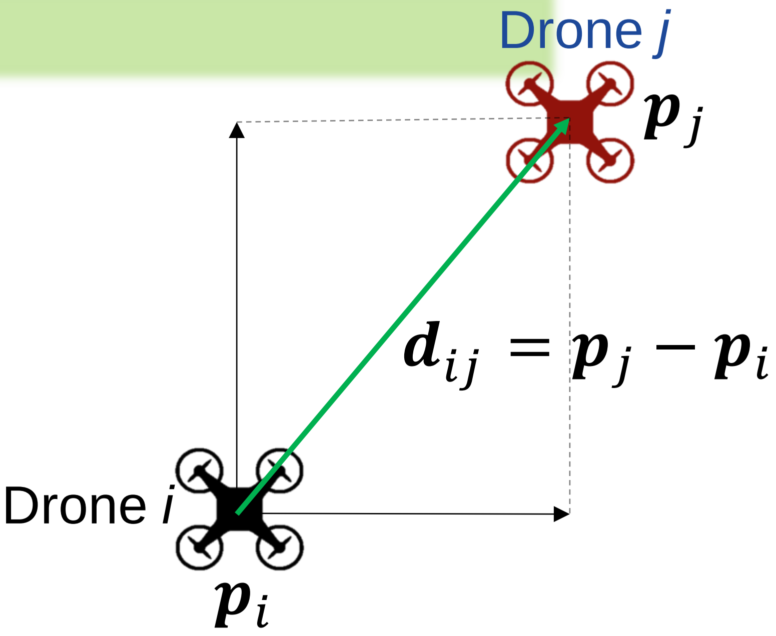
1. Fixed anchors with knows positions
2. Goal: localize the drone



$$\boldsymbol{p}_{drone}^{ML} = \arg\min_{\boldsymbol{p}_{UAV}} \sum_{i=1}^{n} w_i^2$$

$$s.t. \;\; w_i = \|\boldsymbol{p}_{drone} - \boldsymbol{p}_i\| - d_i$$

Pair-wise Relative Localization



1. Involves two agents at a time
2. The goal is to estimate the vector $\boldsymbol{d}_{ij}$
3. $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are position vectors

Drone i 

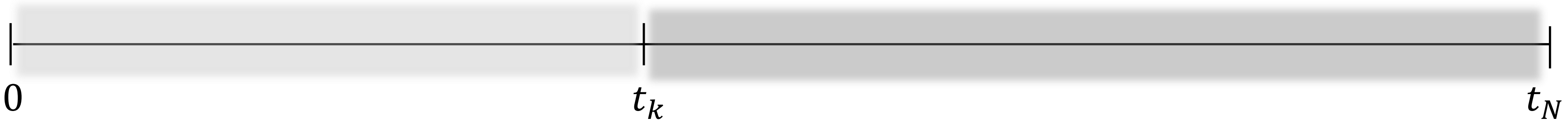
$$\boldsymbol{p}_i(t_k) = \boldsymbol{p}_i(t_N) - \int_{t_k}^{t_N} \boldsymbol{v}_i \, dt$$

velocity

Drone j 

$$\boldsymbol{p}_j(t_k) = \boldsymbol{p}_j(t_N) - \int_{t_k}^{t_N} \boldsymbol{v}_j \, dt$$

$$\longrightarrow \quad \boldsymbol{d}_{ij}(t_k) = \boldsymbol{d}_{ij}(t_N) - \int_{t_k}^{t_N} (\boldsymbol{v}_j - \boldsymbol{v}_i) dt$$

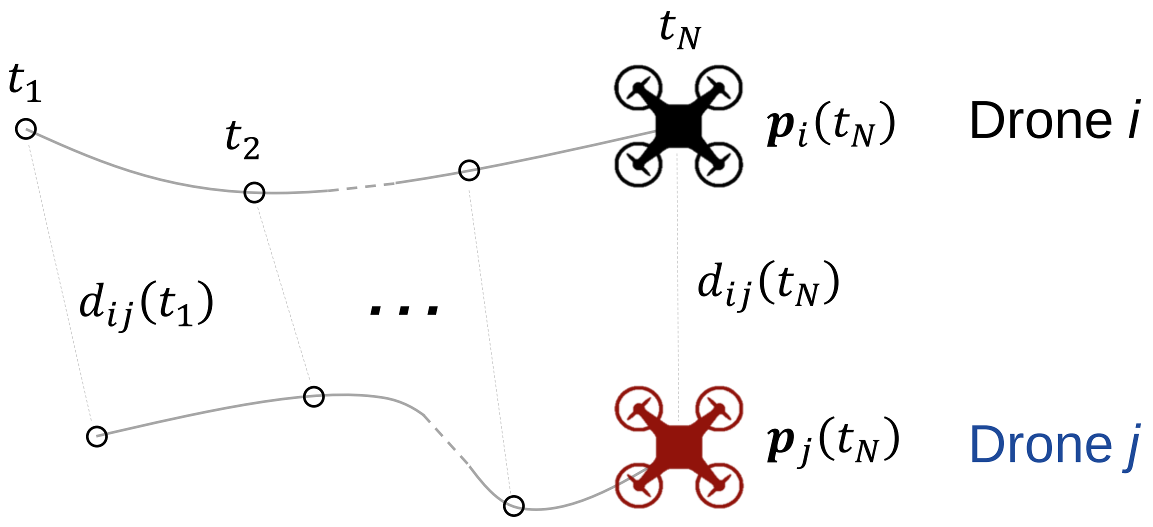


Relative Localization: the Optimization Problem

1. Requires a history of measurements
2. d_{ij} are measured via UWB
3. $\boldsymbol{v}_i$, $\boldsymbol{v}_j$ are estimated onboard

$$\boldsymbol{d}_{ij}^{ML}(t_N) = arg \min_{\boldsymbol{d}_{ij}(t_N)} \sum_{k=1}^{N} w_k^2$$

$$\text{s.t. } w_k = \left\| \boldsymbol{d}_{ij}(t_N) - \int_{t_k}^{t_N} (\boldsymbol{v}_j - \boldsymbol{v}_i) dt \right\| - d_{ij}(t_k)$$



Distinction:

- d_{ij} distance measurement, scalar
- $\boldsymbol{d}_{ij}$ relative localization vector

Comparison with Classical Trilateration

The relative localization problem

$$\boldsymbol{d}_{ij}^{ML}(t_N) = arg \min_{\boldsymbol{d}_{ij}(t_N)} \sum_{k=1}^{N} w_k^2$$

$$\text{s.t.} \ \ w_k = \left\| \boldsymbol{d}_{ij}(t_N) - \int_{t_k}^{t_N} (\boldsymbol{v}_j - \boldsymbol{v}_i) dt \right\| - d_{ij}(t_k)$$

The trilateration problem

$$\boldsymbol{p}_{UAV}^{ML} = \arg\min_{\boldsymbol{p}_{UAV}} \sum_{i=1}^{n} w_i^2$$

$$\text{s.t.} \ \ w_i = \|\boldsymbol{p}_{UAV} - \boldsymbol{p}_i\| - d_i$$

Comparison with Classical Trilateration

The relative localization problem

$$\boldsymbol{d}_{ij}^{ML}(t_N) = arg \min_{\boldsymbol{d}_{ij}(t_N)} \sum_{k=1}^{N} w_k^2$$

$$s.t.\ \ w_k = \left\| \boldsymbol{d}_{ij}(t_N) - \int_{t_k}^{t_N} (\boldsymbol{v}_j - \boldsymbol{v}_i) dt \right\| - d_{ij}(t_k)$$

The trilateration problem

$$\boldsymbol{p}_{UAV}^{ML} = \arg\min_{\boldsymbol{p}_{UAV}} \sum_{i=1}^{n} w_i^2$$

$$s.t.\ \ w_i = \|\boldsymbol{p}_{UAV} - \boldsymbol{p}_i\| - d_i$$

Comparison with Classical Trilateration

The relative localization problem

$$\boldsymbol{d}_{ij}^{ML}(t_N) = arg \min_{\boldsymbol{d}_{ij}(t_N)} \sum_{k=1}^{N} w_k^2$$

$$s.t.\ \ w_k = \left\| \boldsymbol{d}_{ij}(t_N) - \int_{t_k}^{t_N} (\boldsymbol{v}_j - \boldsymbol{v}_i) dt \right\| - d_{ij}(t_k)$$

The trilateration problem

$$\boldsymbol{p}_{UAV}^{ML} = \arg \min_{\boldsymbol{p}_{UAV}} \sum_{i=1}^{n} w_i^2$$

$$s.t.\ \ w_i = \|\boldsymbol{p}_{UAV} - \boldsymbol{p}_i\| - d_i$$

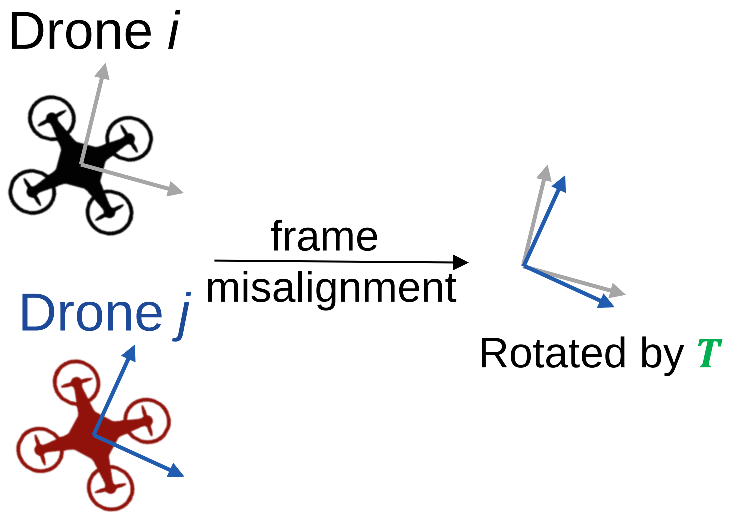
Relative Localization: including the heading

1. Agents exhibit different yaw drift
2. $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are expressed in different frames
3. Solution: account for the rotation

Rotation matrix

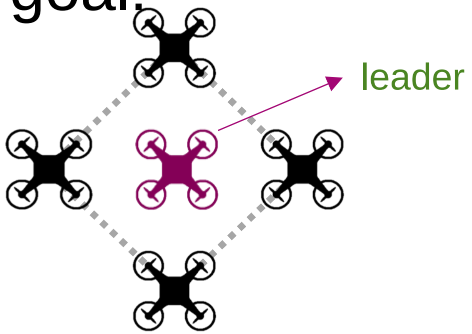
$$\boldsymbol{d}_{ij}^{ML}(t_N), \boldsymbol{T} = arg \min_{\boldsymbol{d}_{ij}(t_N)} \sum_{k=1}^{N} w_k^2$$

$$\text{s.t.} \quad w_k = \left\| \boldsymbol{d}_{ij}(t_N) - \int_{t_k}^{t_N} (\boldsymbol{T}\boldsymbol{v}_j - \boldsymbol{v}_i) dt \right\| - d_{ij}(t_k)$$

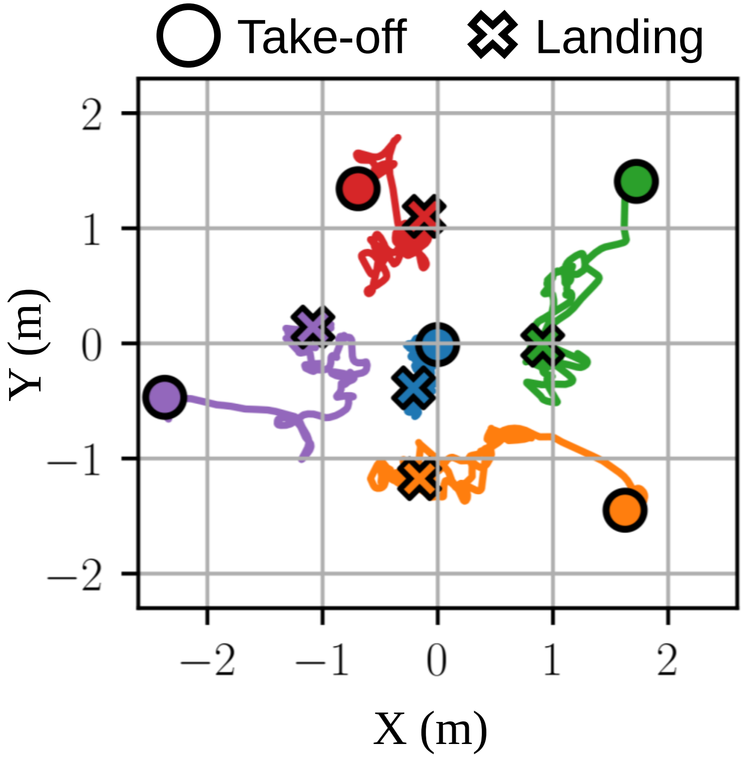


Pair-wise Relative Localization: Experimental Results

1. Measurements acquired in-field
2. Performed with five drones
3. Formation goal:

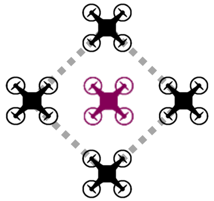


4. Relative localization error: 55.5 cm
5. Heading estimation error: 24°

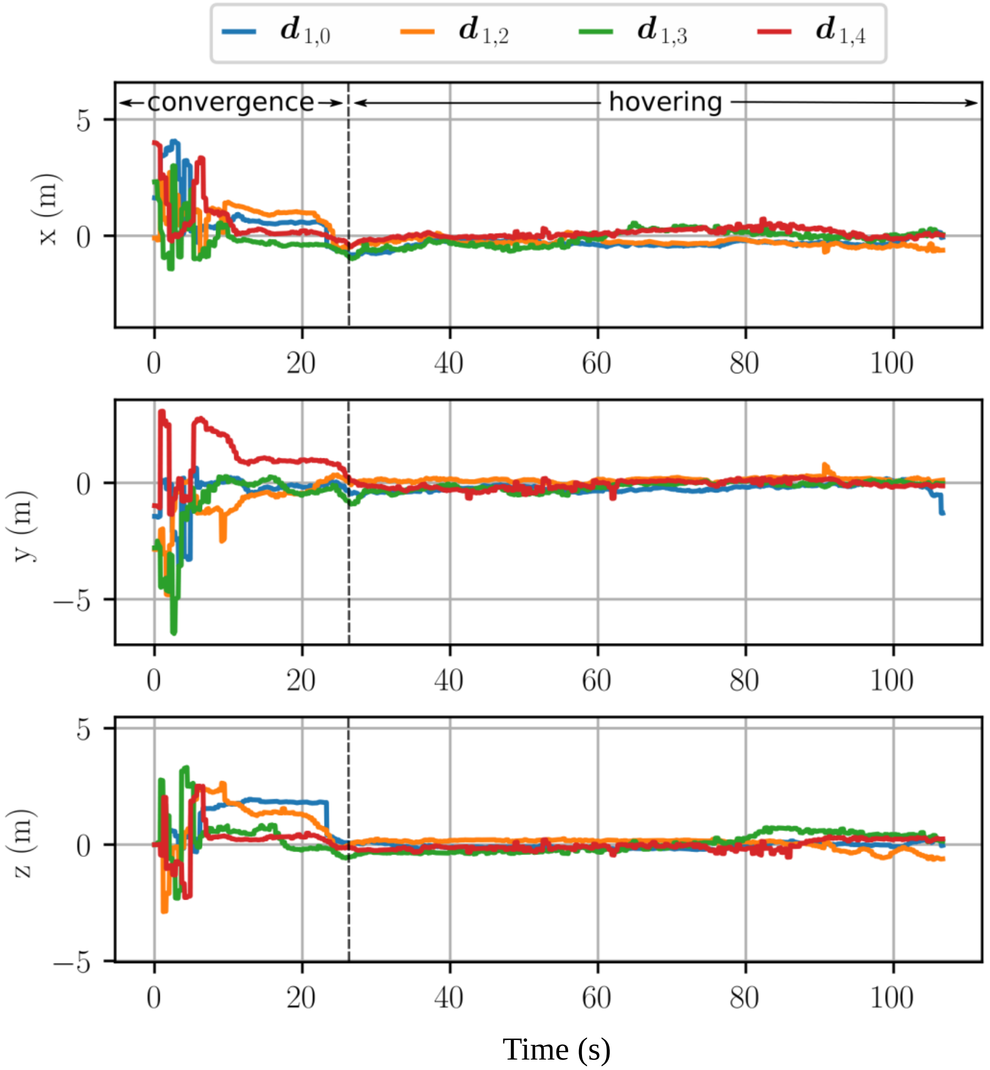


Relative Localization: Experimental Results

1. Acquired in-field
2. Performed with five drones
3. Formation goal:

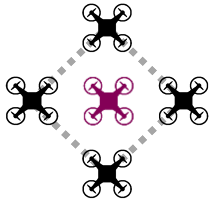


4. Relative localization error: 55.5 cm
5. Heading estimation error: 24°
6. Convergence time: 25 s



Relative Localization: Experimental Results

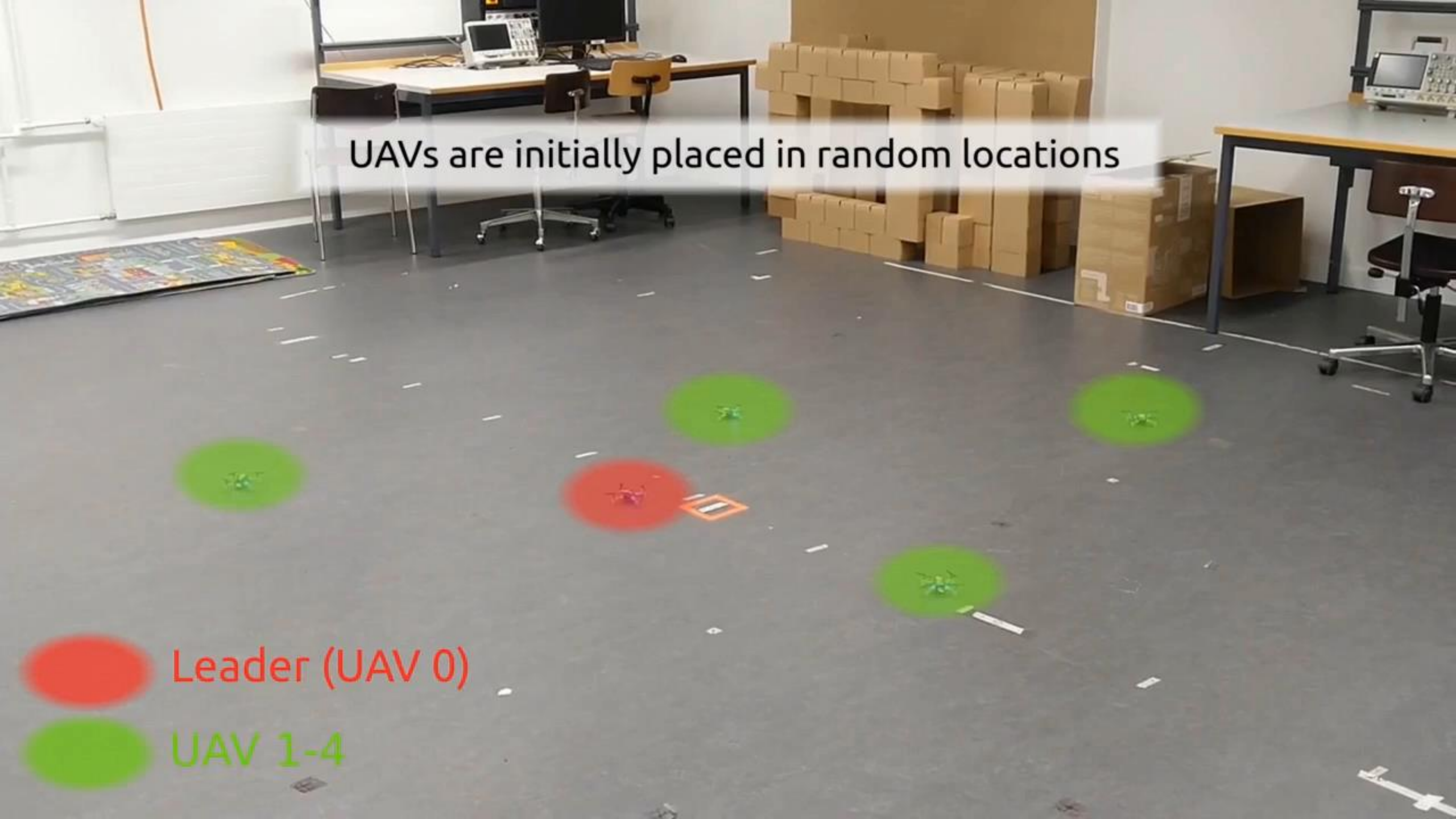
1. Acquired in-field
2. Performed with five drones
3. Formation goal:



4. Relative localization error: 55.5 cm
5. Heading estimation error: 24°
6. Convergence time: 25 s

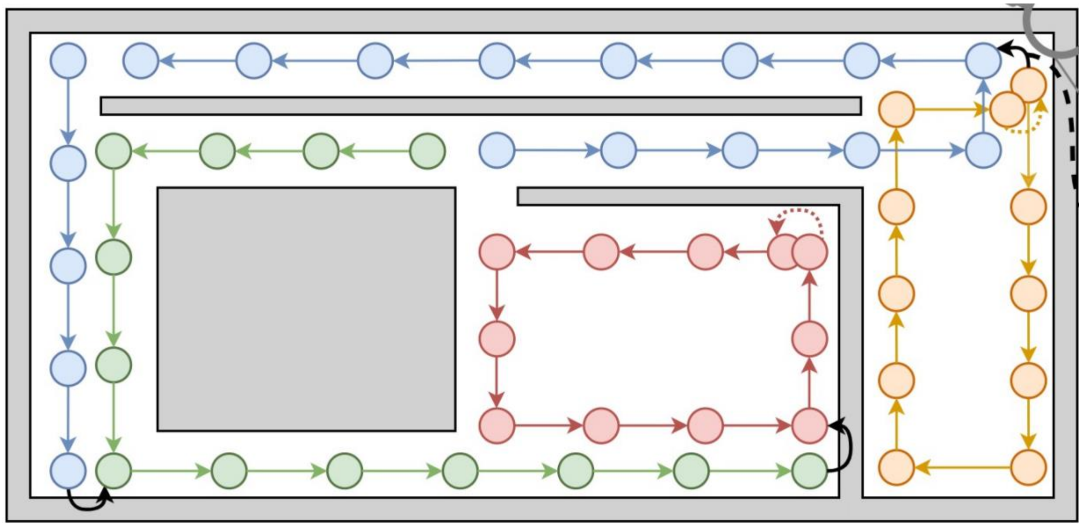
https://github.com/ETH-PBL/swarm-relative-localization

A Relative Infrastructure-less Localization Algorithm for Decentralized and Autonomous Swarm Formation


UAVs are initially placed in random locations
Leader (UAV 0)
UAV 1-4

UWB for Collaborative SLAM

- Autonomous collaborative exploration
- Inter-drone scan-matching
- Sharing only the common, not the poses
- Coordination and scan exchange via UWB




TX
RX
SFD
RXOK
UWB05

Hands-On Exercise

https://github.com/vladniculescu/localization-workshop

ETH zürich