From autonomous lawn mowers to bat drones

Dynamical distance geometry in the wild

Frederike Dümbgen
Joined work with Michalina Pacholska, Adam Scholefield, Adrien Hoffet, Martin Vetterli

Mini-symposium on Sensor Network Localization and Dynamical Distance Geometry

Monday May 24, 2021

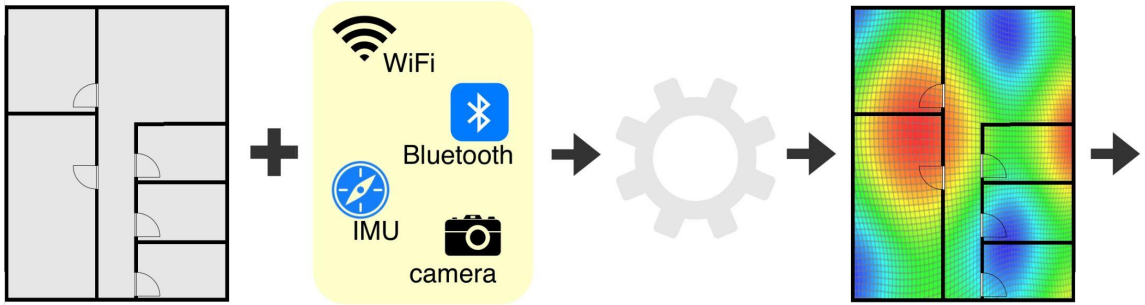
LCAV EPFL

How we got here...

©Vidinoti

How we got here...

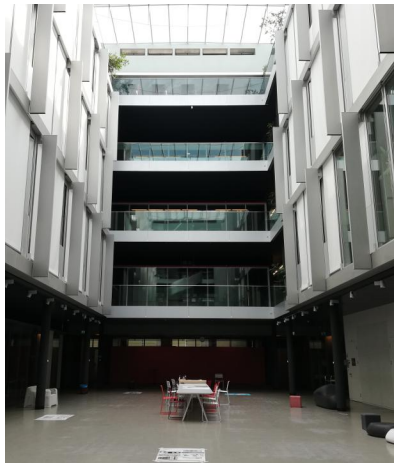
Question: can we create a **user-friendly** indoor localization system which is **smartphone-compatible** and has **meter-level accuracy**?

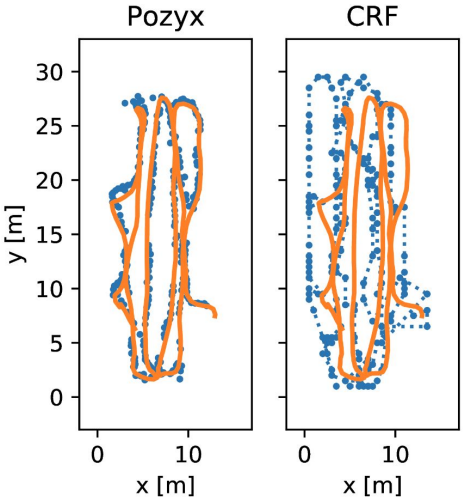


localized AR
contact tracing
pedestrian flow
...

How we got here...

Question: can we create a **user-friendly** indoor localization system which is **smartphone-compatible** and has **meter-level accuracy**?





Answer: yes, **but**

probabilistic, iterative algorithm without guarantees
→ **what are the limits?**

common modalities are very noisy
→ **what about sound?**

Dümbgen, F., Oeschger, C., Kolundzija, M., Scholefield, A., Girardin, E., Leuenberger, J., & Ayer, S. (2019). Multi-Modal Probabilistic Indoor Localization on a Smartphone. International Conference on Indoor Positioning and Indoor Navigation.

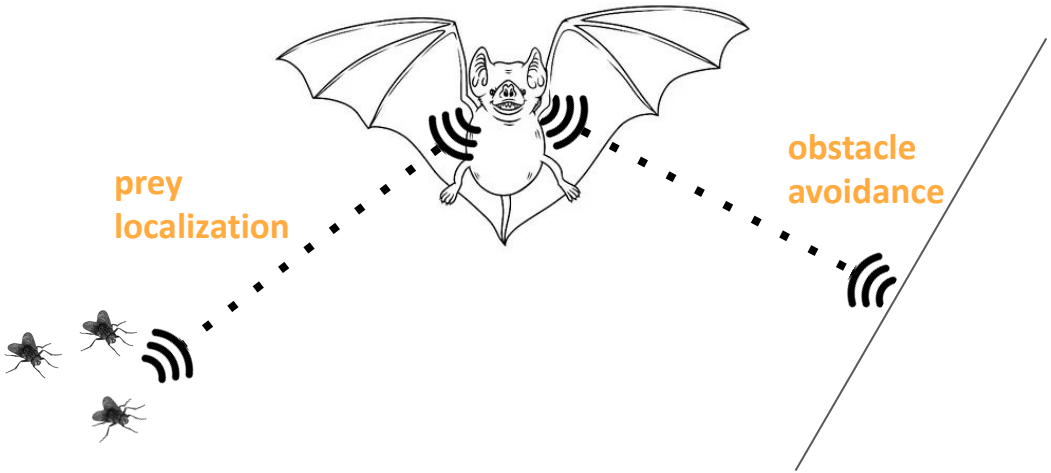
From autonomous lawn mowers...

[1]

Question: when and how can we recover the trajectory of lawnmower?

[1] J. Djugash, B. Hamner, and S. Roth, “Navigating with Ranging Radios: Five Data Sets with Ground Truth”, Journal of Field Robotics, vol. 26,no. 9, pp. 689–695, 2009.

… to bat drones



Question: can we build a drone that can localize walls using sound?

Outline

Autonomous lawnmowers

Continuous localization

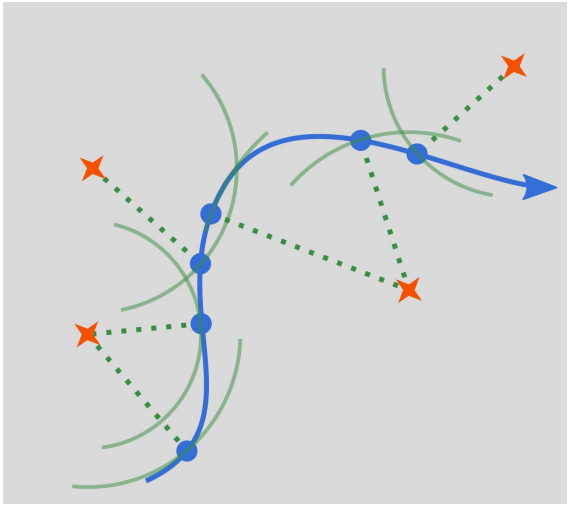
Problem setup
Related work
Algorithm
Recovery guarantees
Results

Bat drones

Wall detection “on the fly”

Problem setup
Traditional methods
Interference detection
Experimental platform
Results

Continuous localization



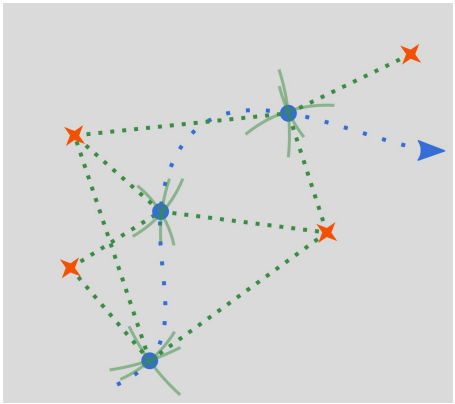
Problem statement: given one distance measurement at a time, can we recover the continuous trajectory of a moving device?

Related work

lateration

e.g. [Beck 2008]

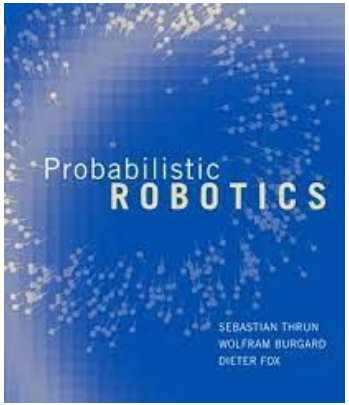
static setups



robotics

e.g. [Thrun 2005]

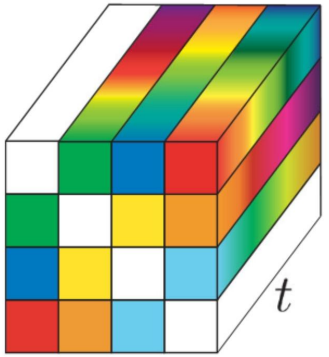
no guarantees



kinetic EDMs

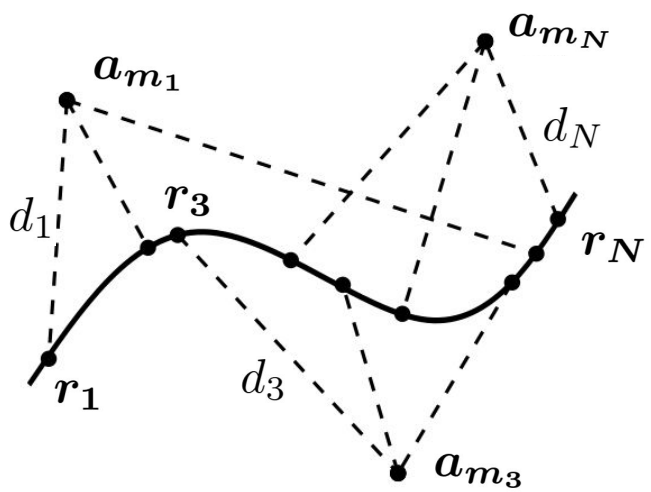
e.g. [Tabaghi 2020]

how about fixed anchors?



Beck, A., Stoica, P., & Li, J. (2008). Exact and approximate solutions of source localization problems. IEEE TSP
Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. MIT press.
Tabaghi, P., Dokmanić, I., & Vetterli, M. (2020). Kinetic Euclidean Distance Matrices. IEEE TSP

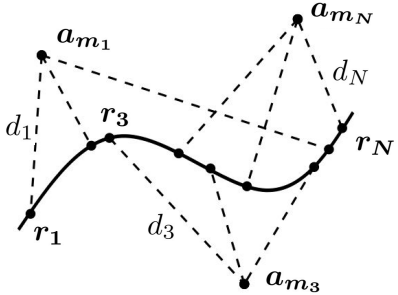
Algorithm — derivation



1) parametrize
2) square
3) vectorize
4) relax

M. Pacholska*, F. Dümbgen*, and A. Scholefield. Relax and Recover: Guaranteed Range-Only Continuous Localization, RAL 2020. (*equal contributions)

Algorithm — derivation



trajectory model:

$$\boldsymbol{r}(t) = \sum_{k=0}^{K-1} \boldsymbol{c_k} f_k(t) = \boldsymbol{c_0} + \boldsymbol{c_1} \cos\left(\frac{2\pi}{\tau} t\right) + \boldsymbol{c_2} \sin\left(\frac{4\pi}{\tau} t\right) \quad \text{(bandlimited)}$$

$$= \boldsymbol{c_0} + \boldsymbol{c_1} t + \boldsymbol{c_2} t^2 \quad \text{(polynomial)}$$

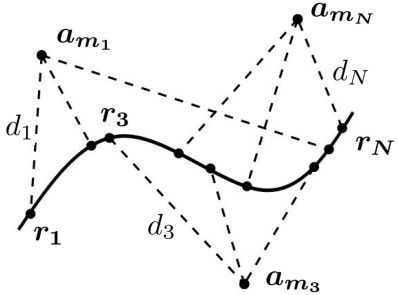
1) **parameterize**
2) square
3) vectorize
4) relax

sampled version:

$$\boldsymbol{r_n} := \boldsymbol{r}(t_n) = \begin{bmatrix} \boldsymbol{c_0} & \dots & \boldsymbol{c_{K-1}} \end{bmatrix} \begin{bmatrix} f_0(t_n) \\ \vdots \\ f_{K-1}(t_n) \end{bmatrix} = \boldsymbol{C f_n}$$

M. Pacholska*, F. Dümbgen*, and A. Scholefield. Relax and Recover: Guaranteed Range-Only Continuous Localization, RAL 2020. (*equal contributions)

Algorithm — derivation



1) parameterize
2) **square**
3) vectorize
4) relax

ideal but hard to solve:

$$\boldsymbol{C}^{\boldsymbol{MLE}} = \arg\min_{C} \sum_{n=0}^{N-1} \left(d_n - \|\boldsymbol{a}_{\boldsymbol{m_n}} - \boldsymbol{C}\boldsymbol{f_n}\|\right)^2$$

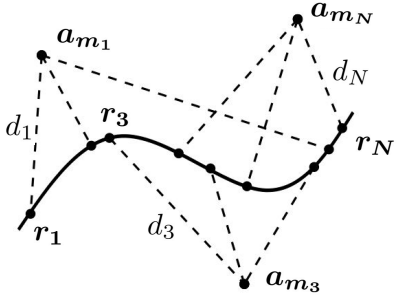
"squaring trick":

$$\hat{\boldsymbol{C}} = \arg\min_{C} \sum_{n=0}^{N-1} \left(d_n^2 - \|\boldsymbol{a}_{\boldsymbol{m_n}} - \boldsymbol{C}\boldsymbol{f_n}\|^2\right)^2$$

$$= \arg\min_{C} \sum_{n=0}^{N-1} \left(d_n^2 - \|\boldsymbol{a}_{\boldsymbol{m_n}}\|^2 + 2\boldsymbol{a}_{\boldsymbol{m_n}}^{\top}\boldsymbol{C}\boldsymbol{f_n}^2 - \boldsymbol{f_n}^{\top}\boldsymbol{C}^{\top}\boldsymbol{C}\boldsymbol{f_n}\right)^2$$

M. Pacholska*, F. Dümbgen*, and A. Scholefield. Relax and Recover: Guaranteed Range-Only Continuous Localization, RAL 2020. (*equal contributions)

Algorithm — derivation



"trace trick"

$$a_{m_n}^\top C f_n = \mathrm{vect}(a_{m_n} f_n^\top)^\top \mathrm{vect}(C)$$

$$f_n^\top C^\top C f_n = \mathrm{vect}(f_n f_n^\top)^\top \mathrm{vect}(C^\top C)$$

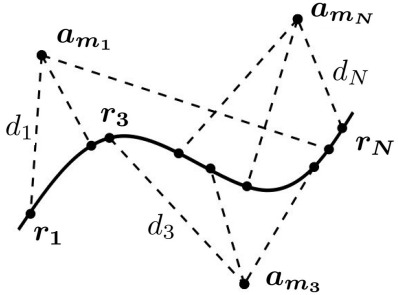
1) parameterize
2) square
3) **vectorize**
4) relax

$$\hat{C} = \arg\min_C \sum_{n=0}^{N-1} \left(d_n^2 - \|a_{m_n}\|^2 + \begin{bmatrix} 2\mathrm{vect}(a_{m_n} f_n^\top)^\top & -\mathrm{vect}(f_n f_n^\top)^\top \end{bmatrix} \begin{bmatrix} \mathrm{vect}(C) \\ \mathrm{vect}(L) \end{bmatrix} \right)^2$$

$$\text{s.t. } L = C^\top C$$

M. Pacholska*, F. Dümbgen*, and A. Scholefield. Relax and Recover: Guaranteed Range-Only Continuous Localization, RAL 2020. (*equal contributions)

Algorithm — derivation



drop quadratic constraint → linear problem

$$\underbrace{\frac{1}{2}\begin{bmatrix} \|a_{m_1}\|^2 - d_1^2 \\ \|a_{m_2}\|^2 - d_2^2 \\ \vdots \\ \|a_{m_N}\|^2 - d_N^2 \end{bmatrix}}_{b} = \underbrace{\begin{bmatrix} \mathrm{vect}(a_{m_1} f_1^\top)^\top \\ \mathrm{vect}(a_{m_2} f_2^\top)^\top \\ \vdots \\ \mathrm{vect}(a_{m_N} f_N^\top)^\top \end{bmatrix}}_{T_A} \underbrace{\begin{bmatrix} \mathrm{vect}(f_1 f_1^\top)^\top \\ \mathrm{vect}(f_2 f_2^\top)^\top \\ \vdots \\ \mathrm{vect}(f_N f_N^\top)^\top \end{bmatrix}}_{T_F} \begin{bmatrix} \mathrm{vect}(C) \\ -\frac{1}{2}\mathrm{vect}(L) \end{bmatrix}$$

1) parameterize
2) square
3) vectorize
4) **relax**

M. Pacholska*, F. Dümbgen*, and A. Scholefield. Relax and Recover: Guaranteed Range-Only Continuous Localization, RAL 2020. (*equal contributions)

Algorithm — final considerations

want to solve:

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{T}_A & \boldsymbol{T}_F \end{bmatrix} \left[\begin{array}{c} \mathbf{vect}(\boldsymbol{C}) \\ -\frac{1}{2}\mathbf{vect}(\boldsymbol{L}) \end{array} \right] \Big\} DK+K^2$$

But measurement matrix is never full (column) rank!

- **bad news** if we wanted to recover full vector
- **good news** since it reduced "unnecessary" dimensions:

rank-reduced problem:

$$\boldsymbol{T}_F = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V} \qquad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{T}_A & \boldsymbol{U}\boldsymbol{\Sigma} \end{bmatrix} \left[\begin{array}{c} \mathbf{vect}(\boldsymbol{C}) \\ -\frac{1}{2}\boldsymbol{V}^\top \mathbf{vect}(\boldsymbol{L}) \end{array} \right] \Big\} DK+r$$

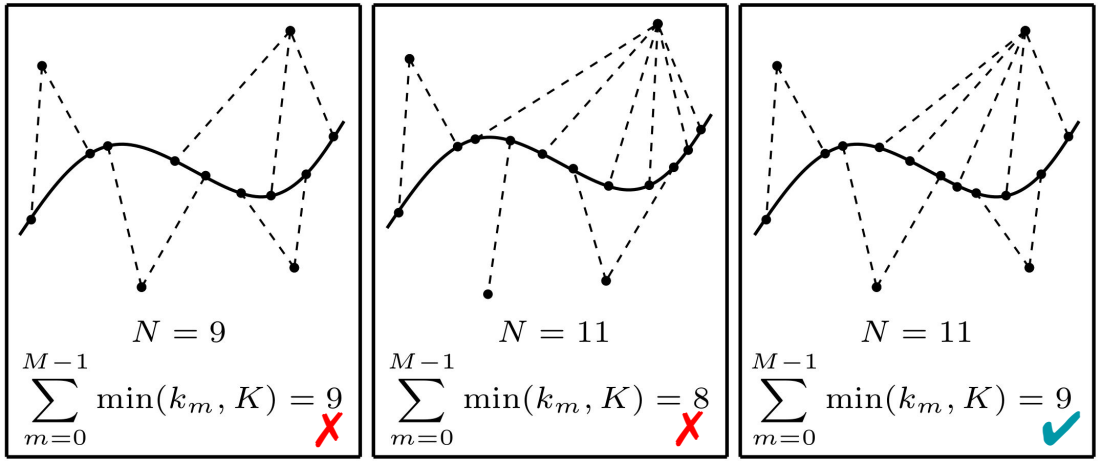
M. Pacholska*, F. Dümbgen*, and A. Scholefield. Relax and Recover: Guaranteed Range-Only Continuous Localization, RAL 2020. (*equal contributions)

Recovery guarantees — theorem

Theorem (simplified): Necessary and sufficient for recovery of relaxed problem:

number of measurements: $N \geq K(D+2) - 1$

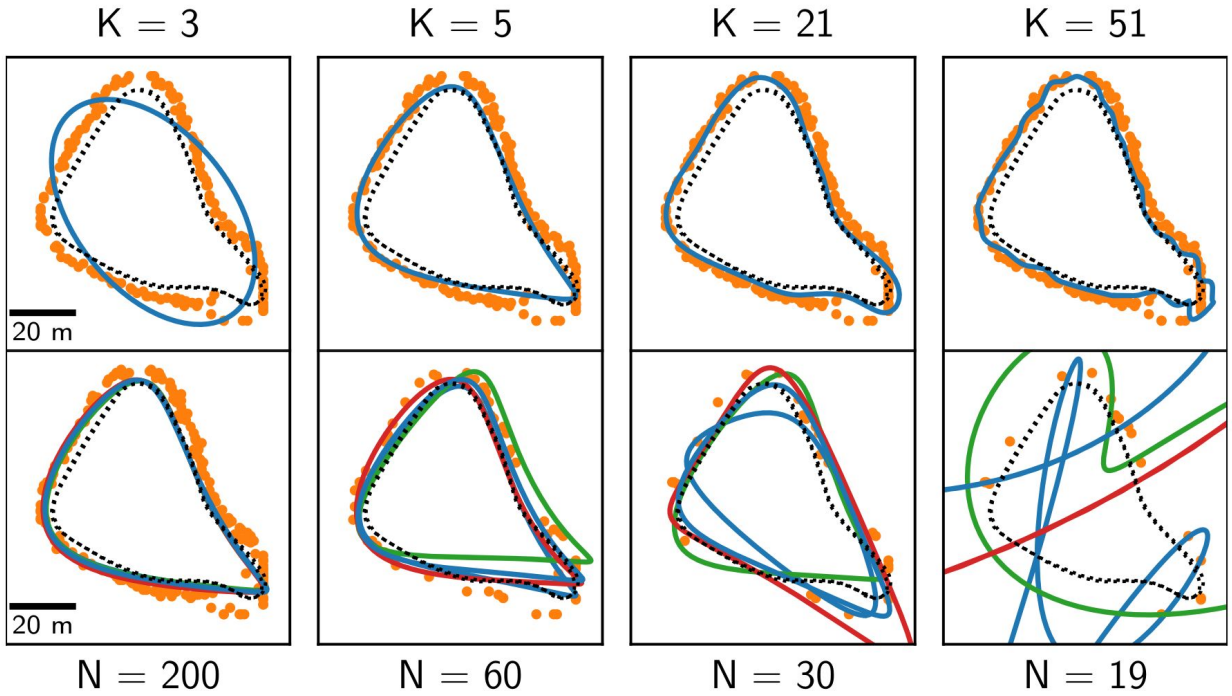
distribution of measurements: $\sum_{m=0}^{M-1} \min(k_m, K) \geq K(D+1),$



$K = 3, D = 2$

Results — qualitative

[1]



[1] J. Djugash, B. Hamner, and S. Roth, “Navigating with Ranging Radios: Five Data Sets with Ground Truth”, Journal of Field Robotics, vol. 26,no. 9, pp. 689–695, 2009.

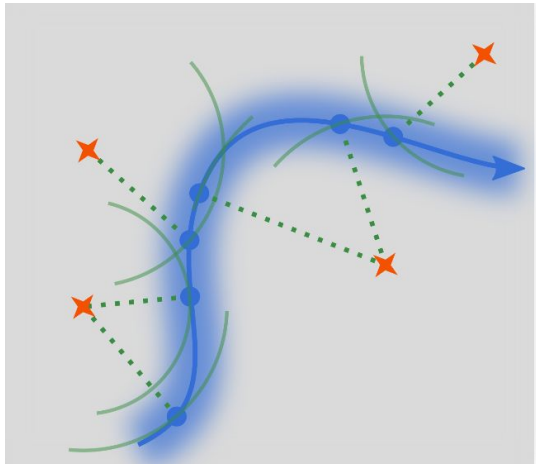
Results — quantitative

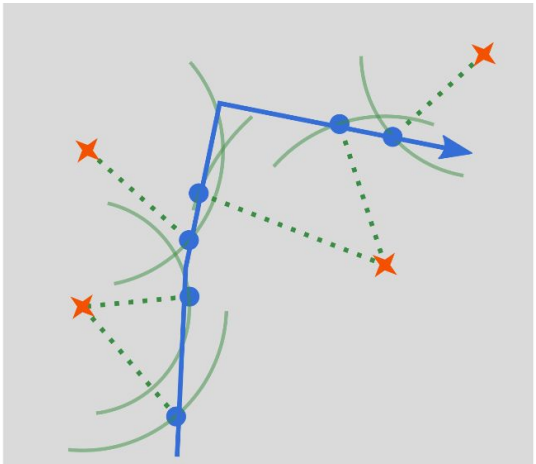
MSE OF RECOVERED TRAJECTORIES FOR TWO REAL-WORLD UWB-BASED LOCALIZATION DATASETS [27].

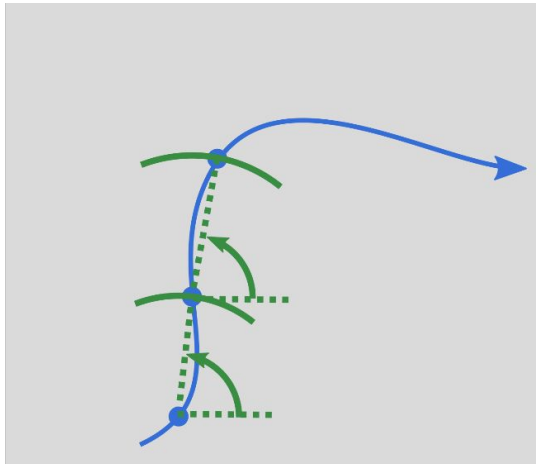
# measurements **N**	**100**			**300**			**499**		
model complexity **K**	**5**	**11**	**19**	**5**	**11**	**19**	**5**	**11**	**19**
model mismatch	3.7	1.8	1.5	3.8	2.0	1.7	3.8	2.1	1.8
SRLS	137.5	141.0	114.5	22.2	22.2	21.8	13.2	13.2	13.2
SRLS fitted	77.9	100.5	4378.0	18.1	17.6	17.8	15.6	14.1	13.9
RLS	113.6	134.3	106.1	16.5	17.2	16.2	9.7	9.7	9.7
RLS fitted	62.1	91.5	5971.6	13.2	12.8	12.9	10.5	9.2	9.4
LM ellipse/line	12.7	147.6	339.8	11.3	25.9	13.2	11.4	11.5	12.0
LM ours weighted	12.7	13.0	20.5	11.3	11.9	12.2	11.4	11.5	12.0
ours	13.9	14.1	69.5	13.2	11.5	11.6	13.2	11.4	11.3
ours weighted	10.3	10.6	68.2	8.8	7.3	7.3	8.7	7.2	6.9
	bandlimited trajectory (*Plaza2*)								

Future work

- probabilistic treatment?
- piecewise trajectories?
- integration of IMU measurements?







Outline

Autonomous lawnmowers

Continuous localization

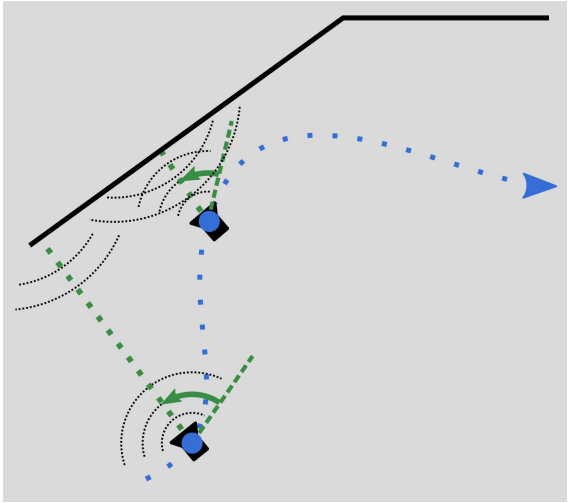
Problem setup
Related work
Algorithm
Recovery guarantees
Results

Bat drones

Wall detection “on the fly”

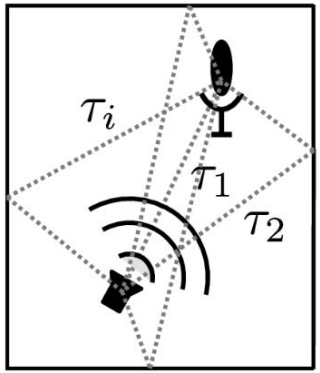
Problem setup
Traditional methods
Interference detection
Experimental platform
Results

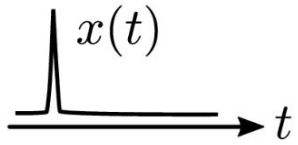
Wall detection “on the fly”

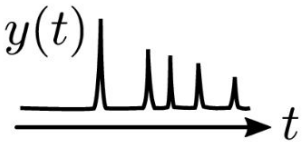


Problem statement: can we detect the distance and angle of the closest wall, given only audio measurements?

Traditional method — RIR reconstruction

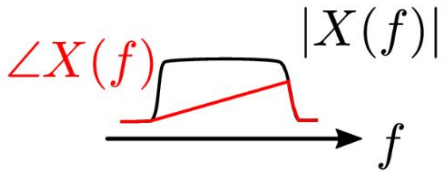


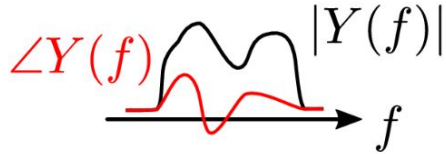




$$y(t) = x(t) \circledast h(t)$$

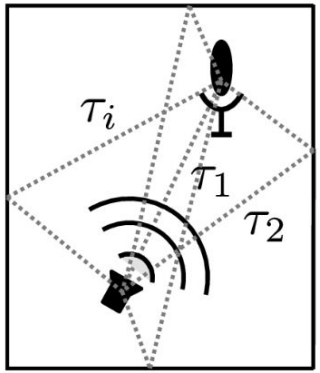
$$= \sum_i \alpha_i x(t - \tau_i)$$



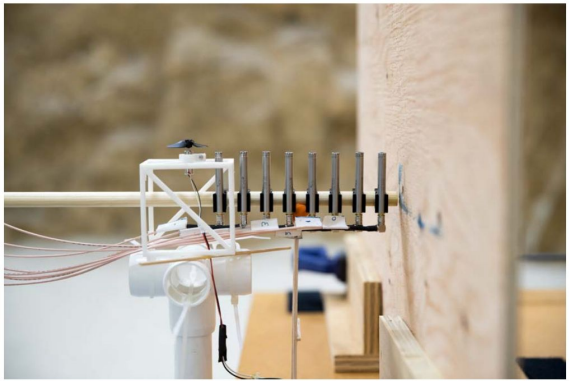


$$Y(f) = X(f)H(f)$$

Traditional method — RIR reconstruction



[1]

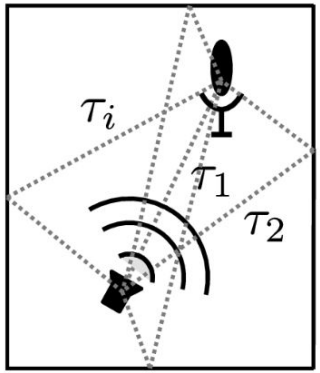


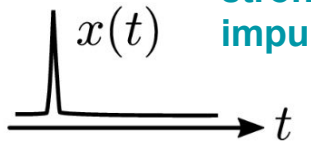
[2]

How about a drone-compatible, portable setup?

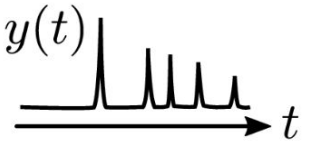
[1] Saqib, U., & Jensen, J. R. (2020). A Model-based Approach to Acoustic Reflector Localization with a Robotic Platform. IROS
[2] Calkins, L. et.al. (2020). Bio-Inspired Distance Estimation using the Self-Induced Acoustic Signature of a Motor-Propeller System. IEEE ICRA

Traditional method — RIR reconstruction





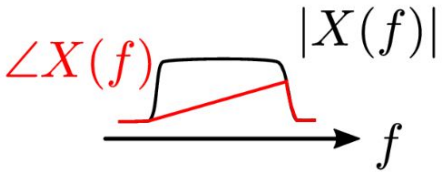
short and strong impulse?



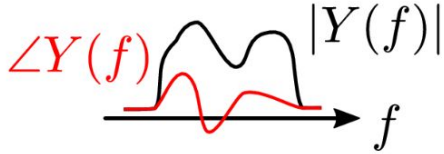
$$y(t) = x(t) \circledast h(t)$$

$$= \sum_i \alpha_i x(t - \tau_i)$$

peak detection?



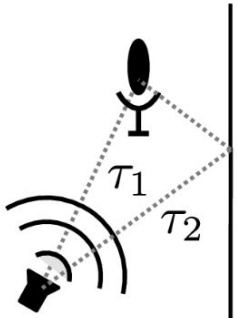
wideband source?

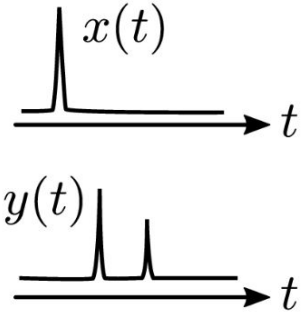


phase measurements?

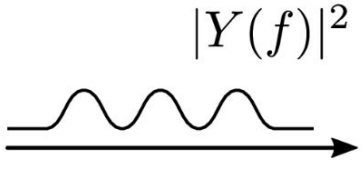
$$Y(f) = X(f)H(f)$$

Our method — interference detection



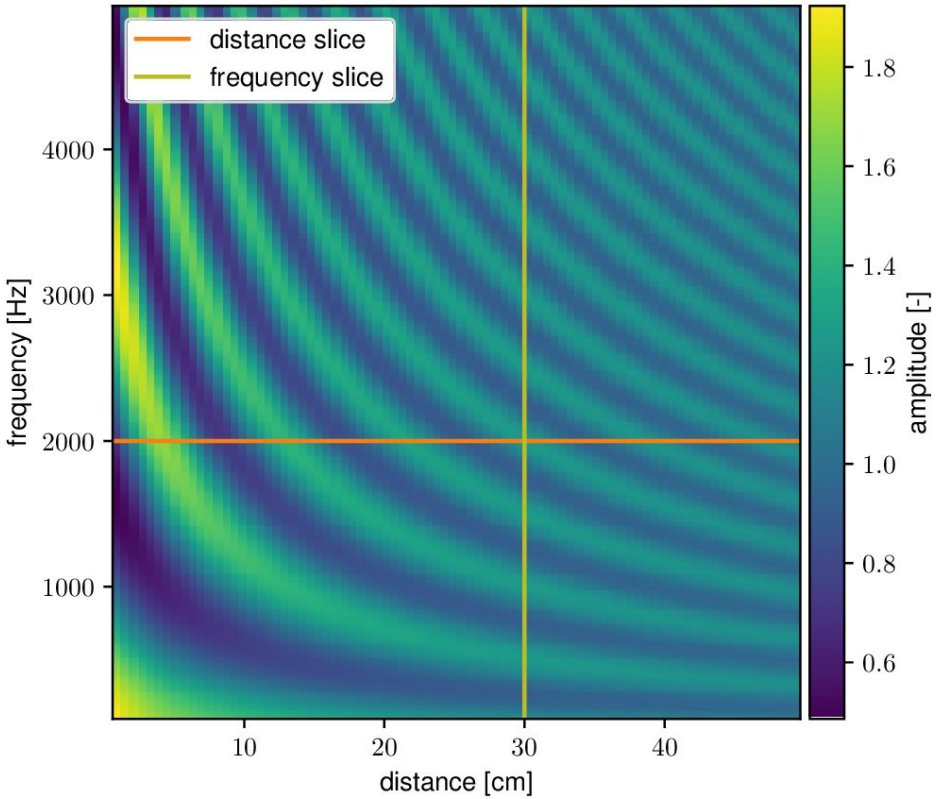


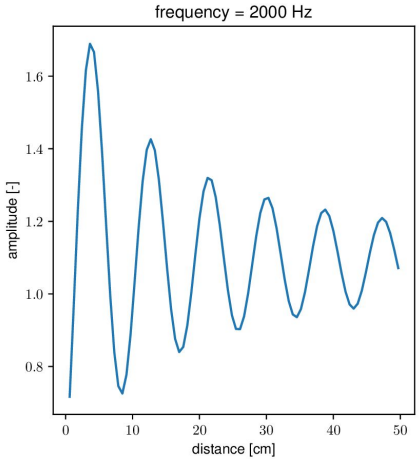
$$y(t) = \alpha_1 x(t - \tau_1) + \alpha_2 x(t - \tau_2)$$

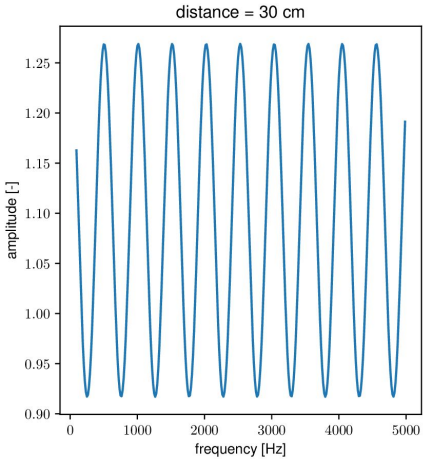


$$|Y(f)|^2 = |X(f)|^2 |\alpha_1 e^{2\pi f \tau_1} + \alpha_2 e^{2\pi f \tau_2}|^2$$
$$\propto \cos\left(2\pi f(\tau_1 - \tau_2)\right)$$

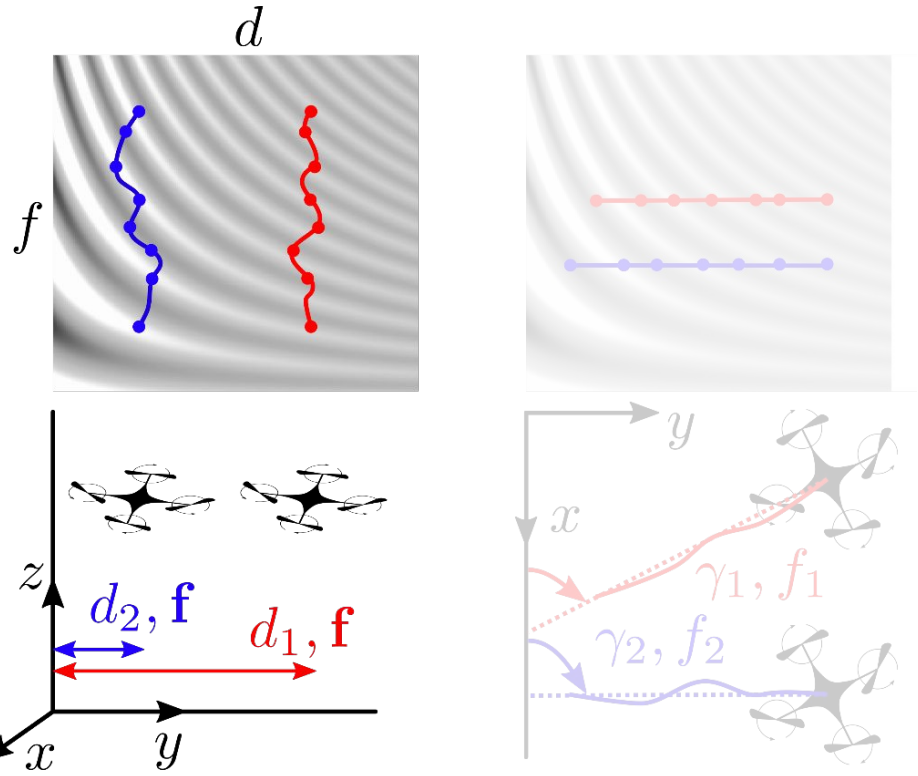
Interference detection — examples



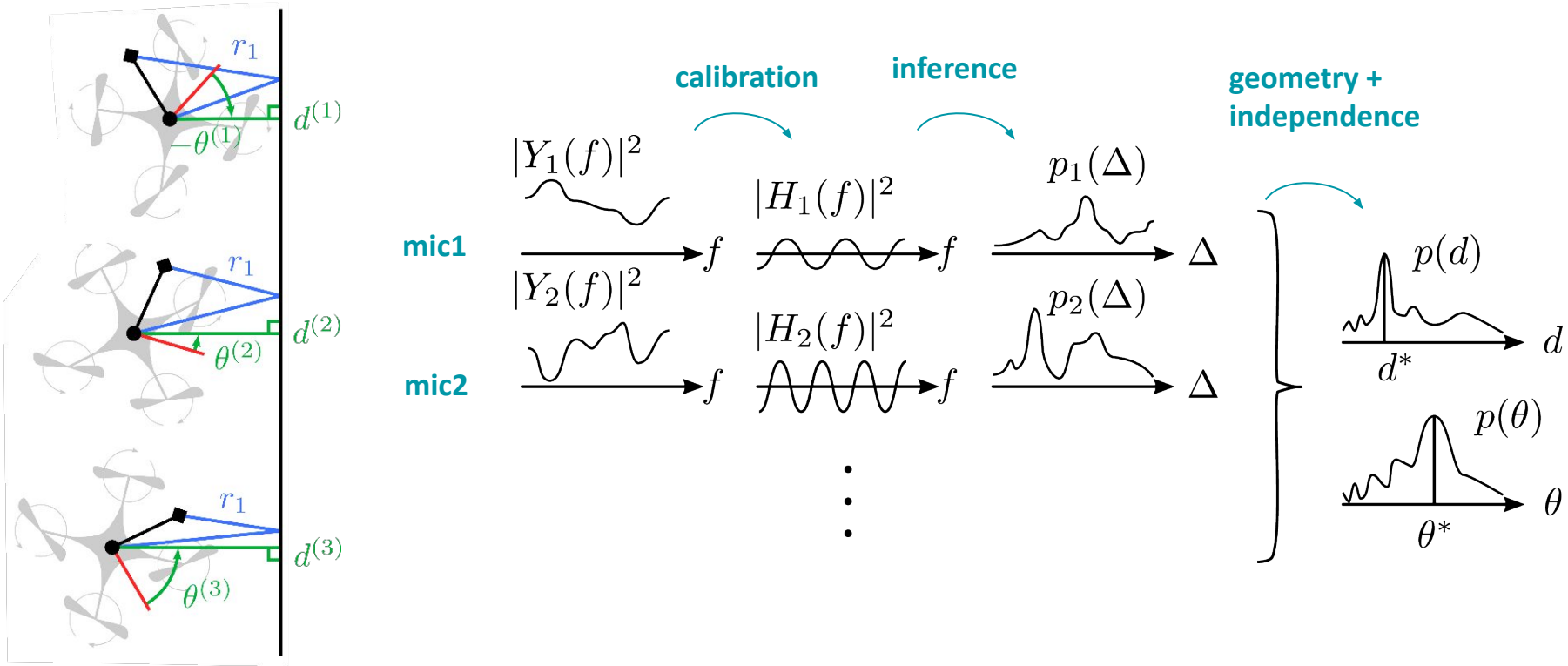




Interference detection — practice



Interference detection — pipeline



Experimental platform — choice

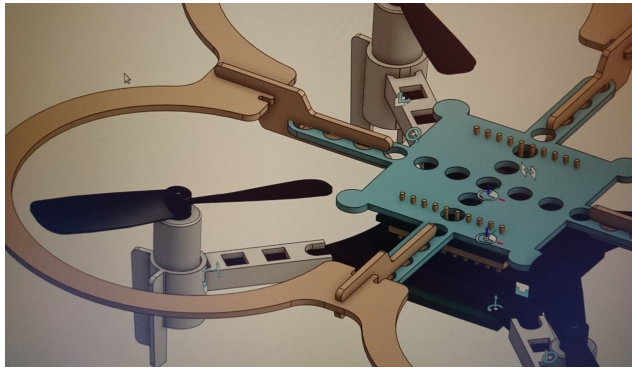
https://www.bitcraze.io/2019/05/sneak-peak-of-whats-coming-at-icra-2019/

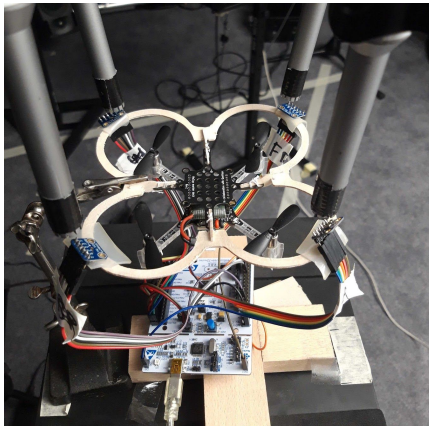
Crazyflie drone by bitcraze:

- modular design (expansion decks)
- open-source firmware and clients
- widely used in research community

“only” need to create audio deck

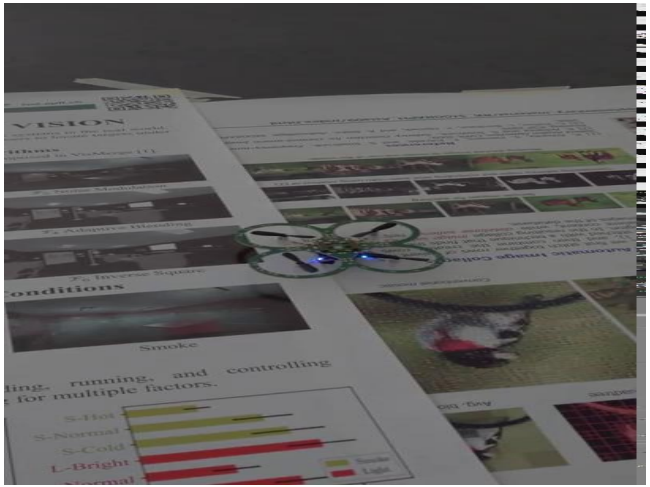
Experimental platform — development process



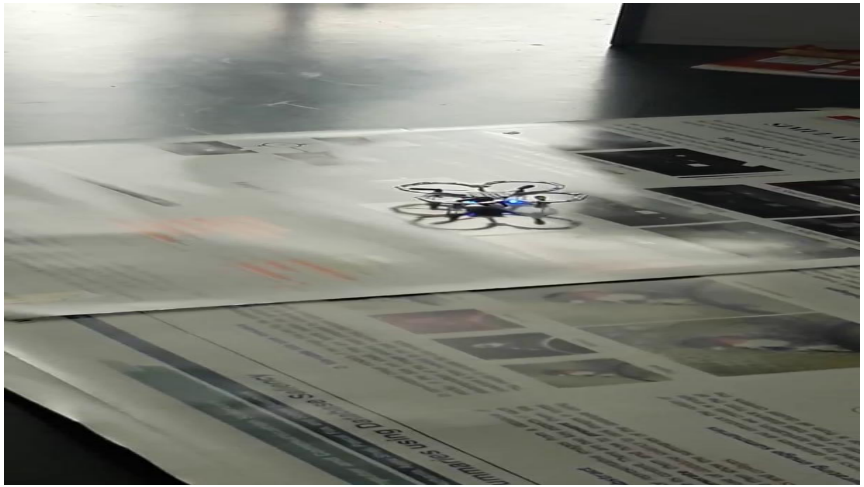


Experimental platform — learning to fly

alpha version



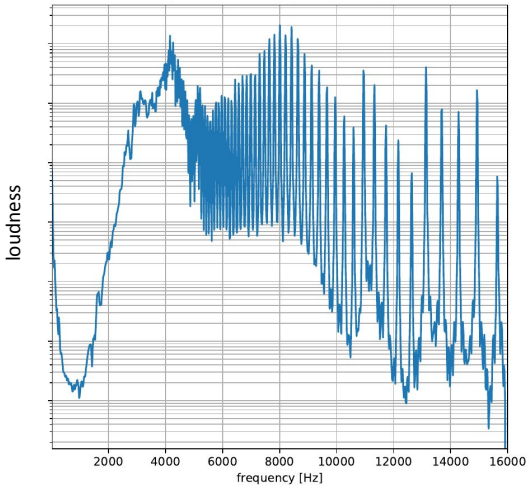
beta version



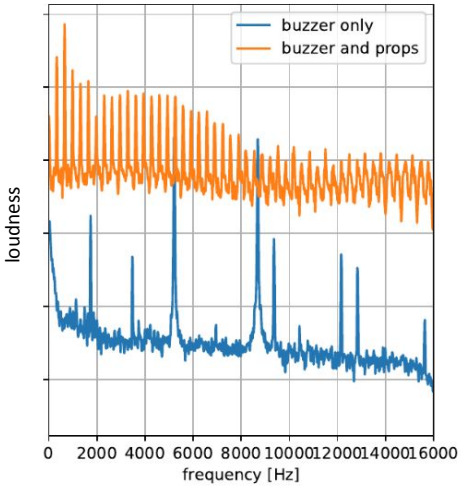
Experimental platform — final design

Experimental platform — challenges

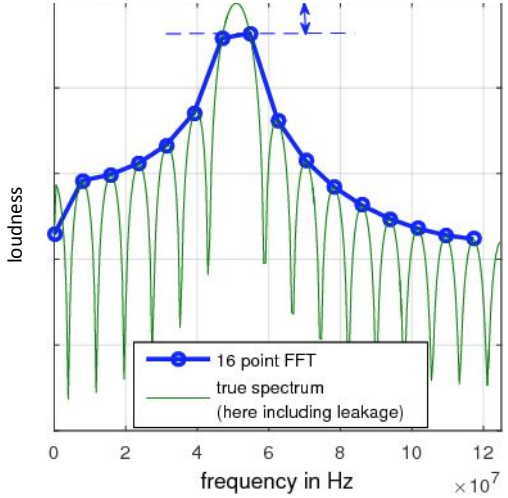
buzzer frequencies



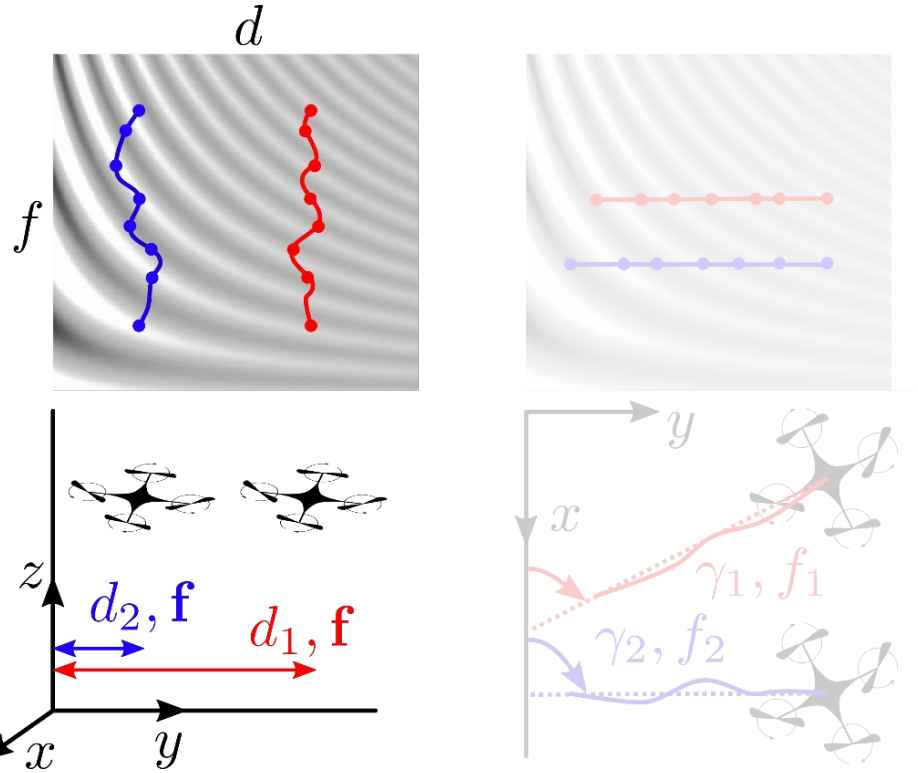
propeller noise



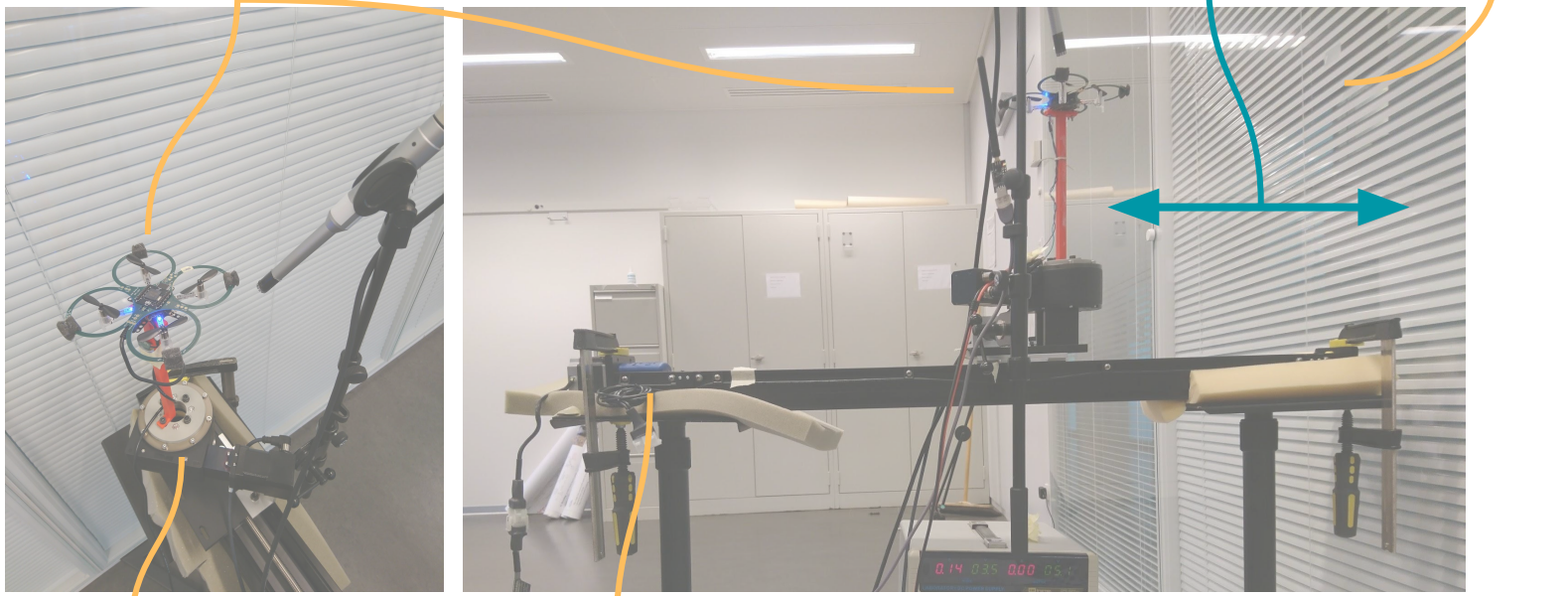
amplitude estimation



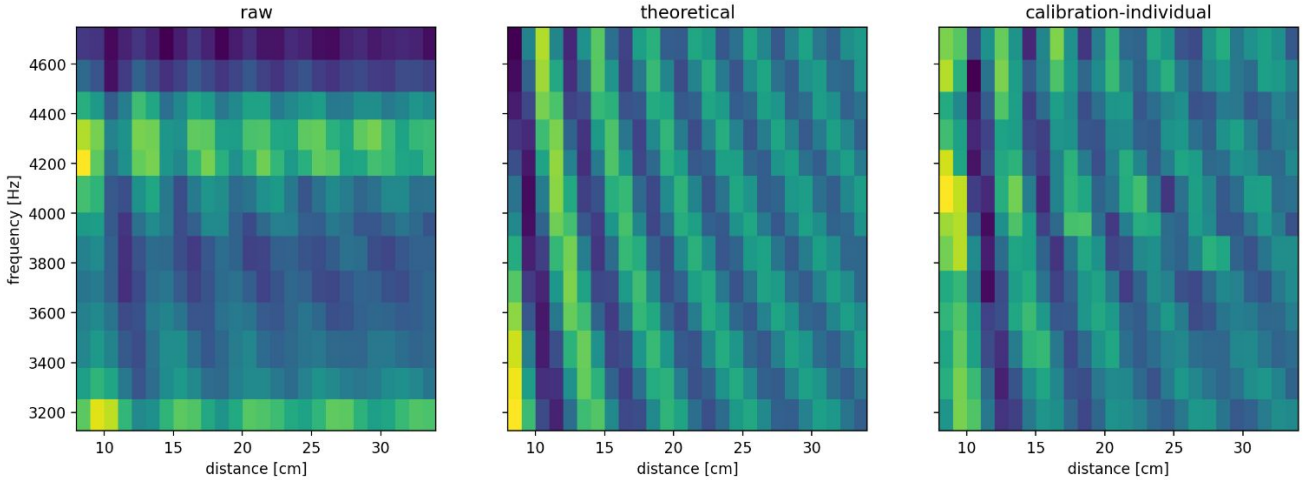
Results — frequency slice



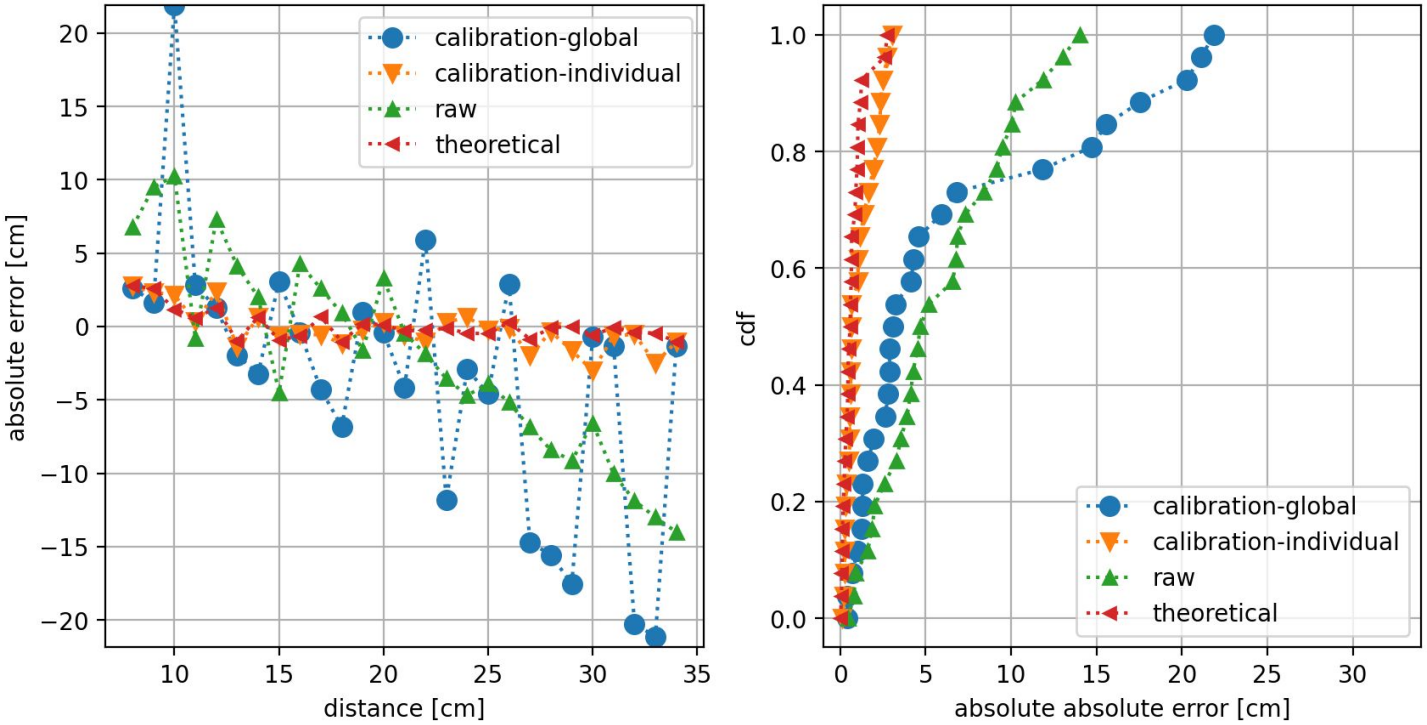
Frequency slice — stepper motor results



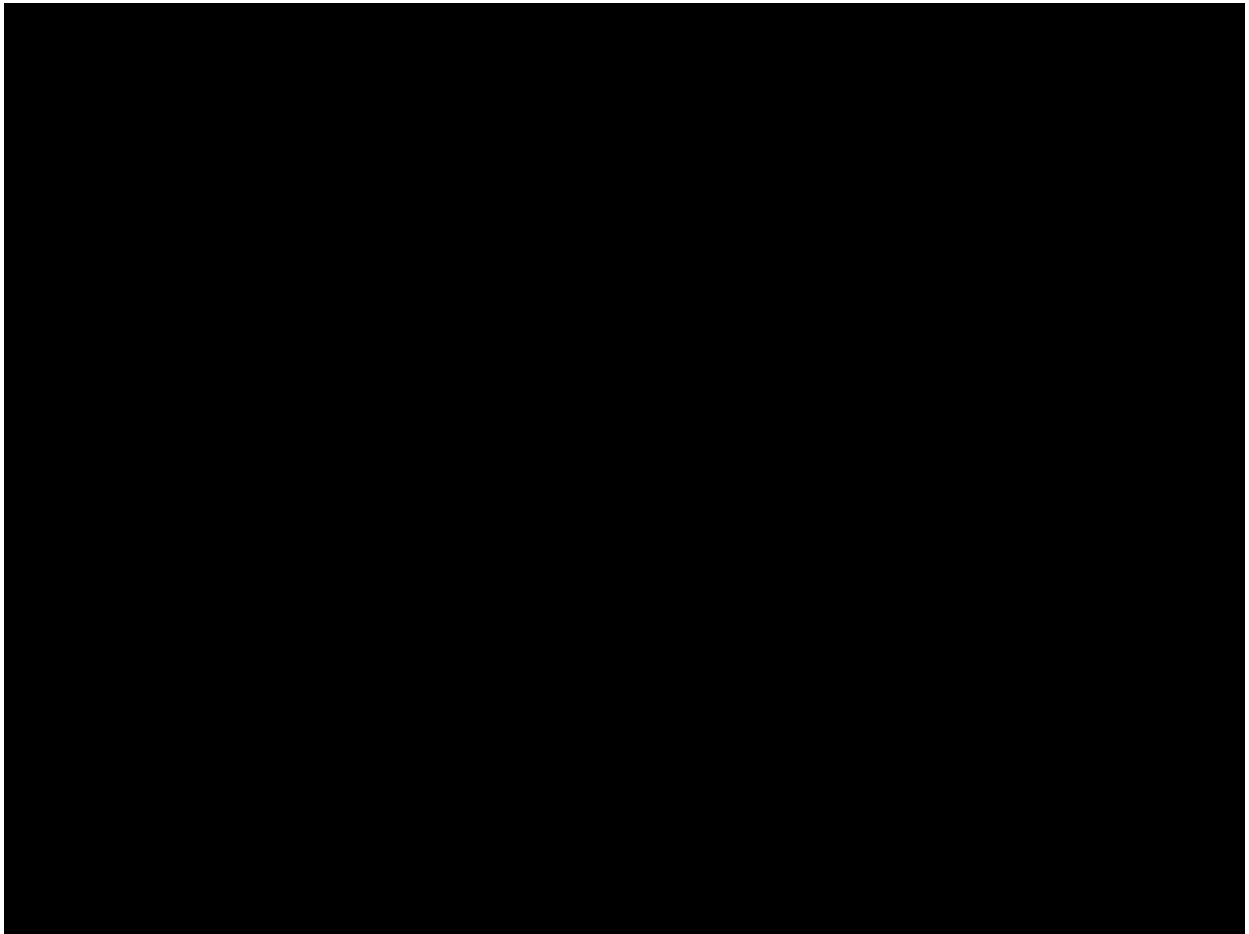
Frequency slice — stepper motor results



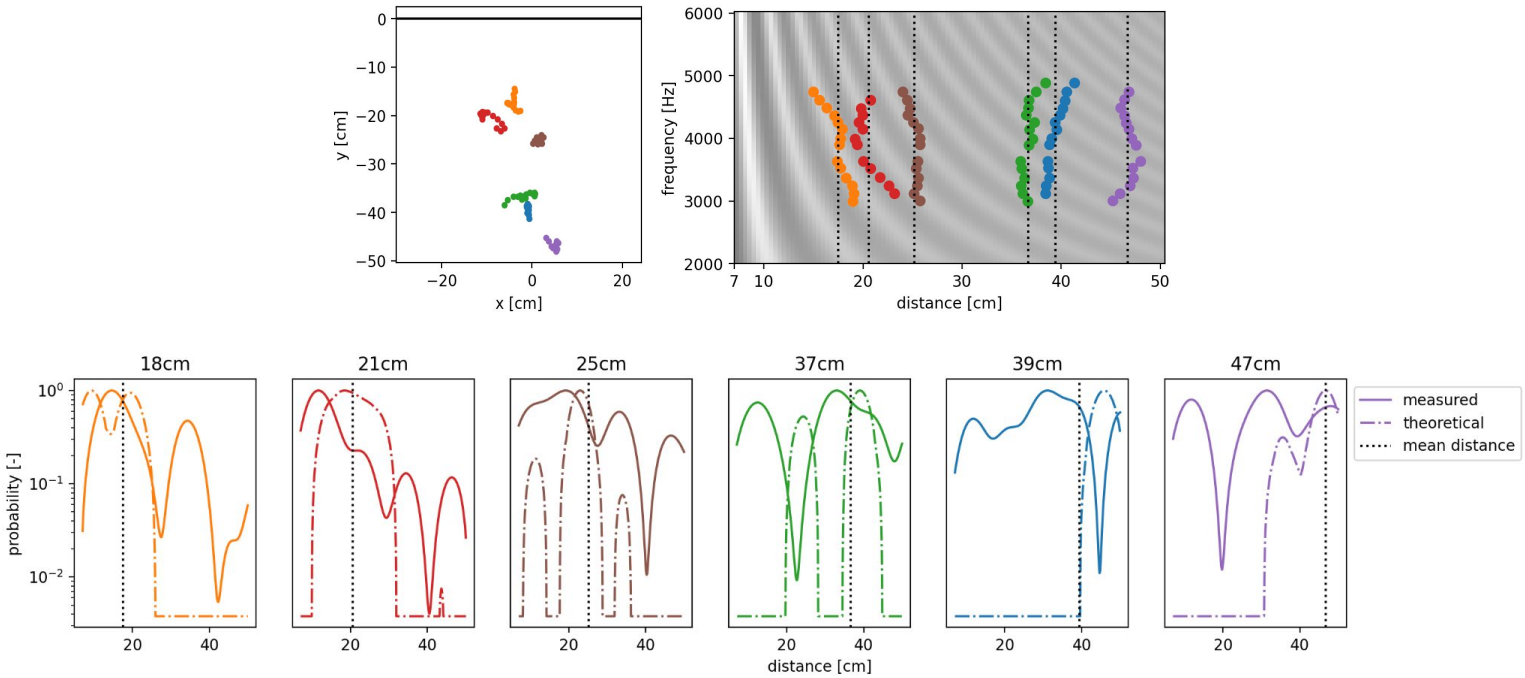
Frequency slice — stepper motor results



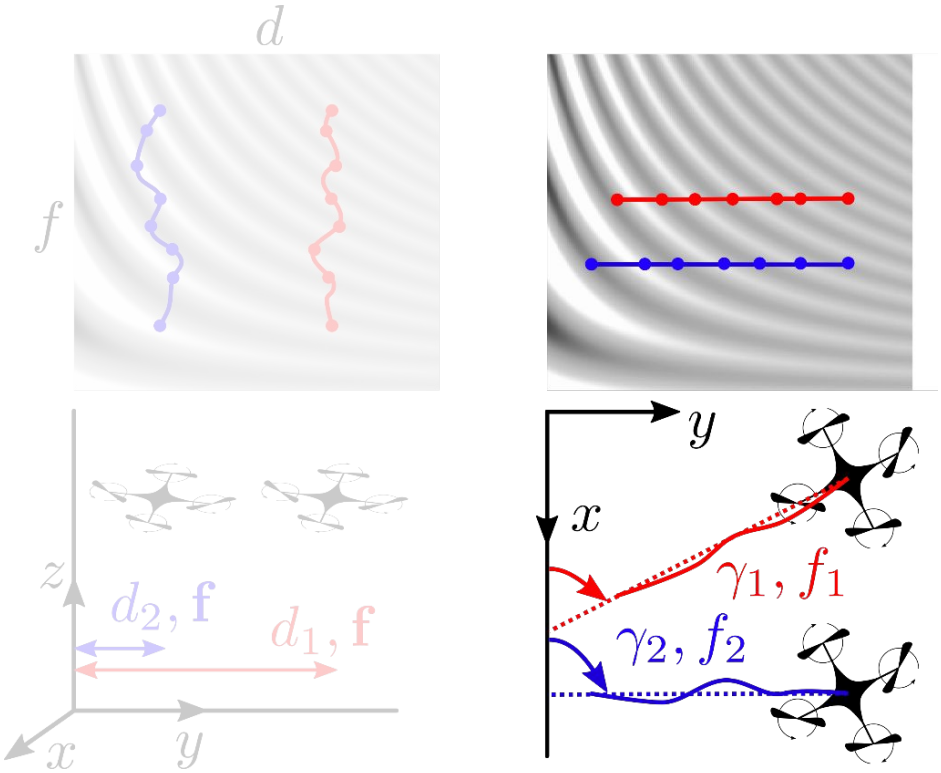
Frequency slice — flying results



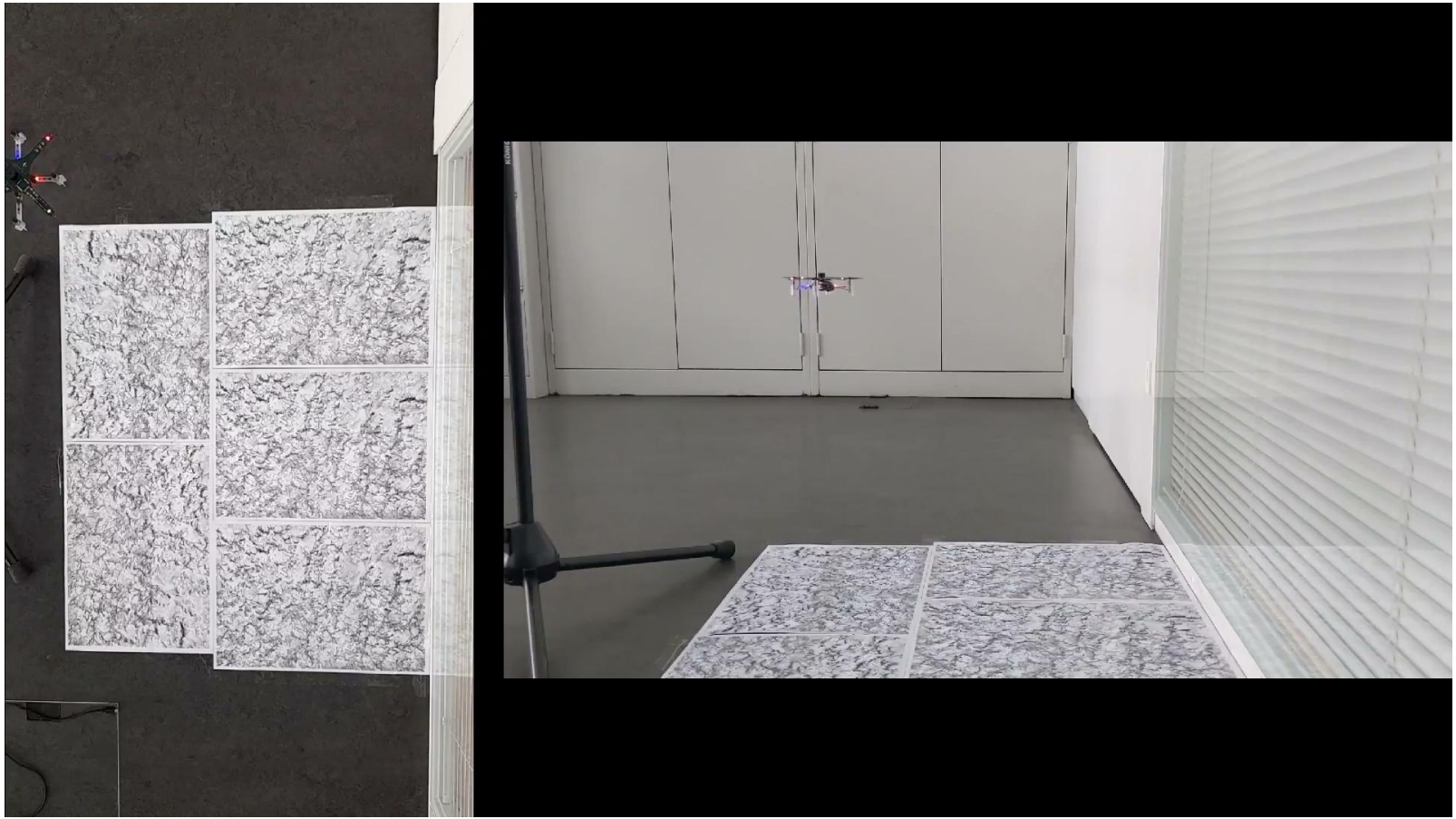
Frequency slice — flying results



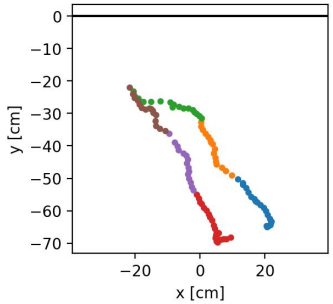
Results — distance slice

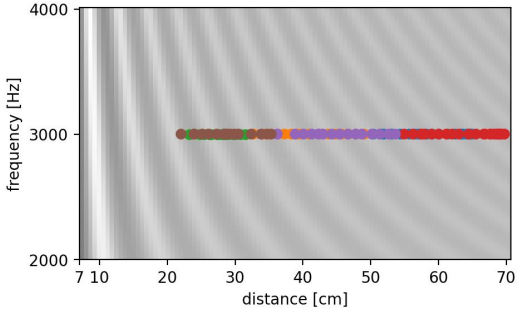


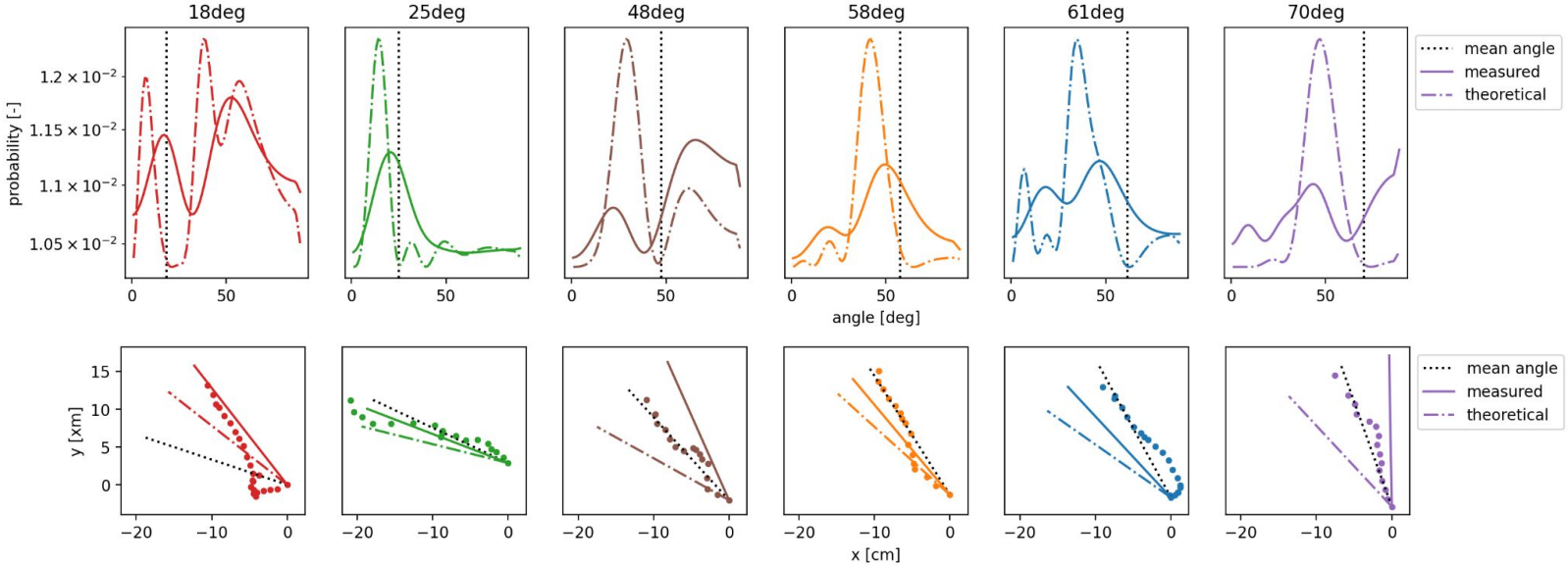
Distance slice — flying results



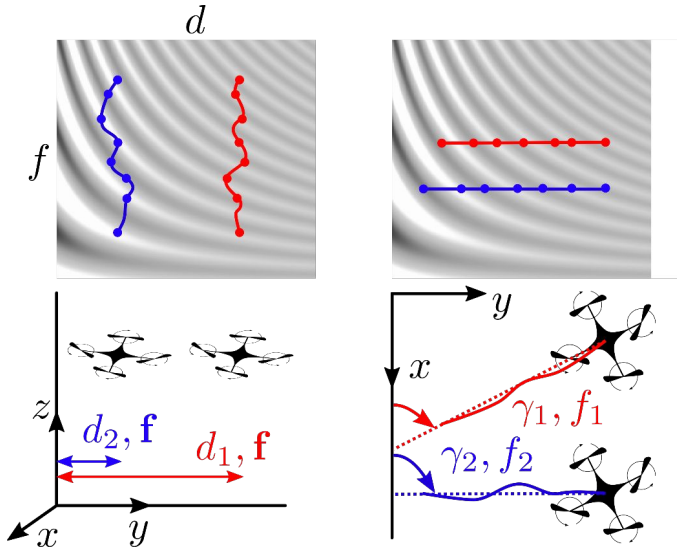
Distance slice — flying results





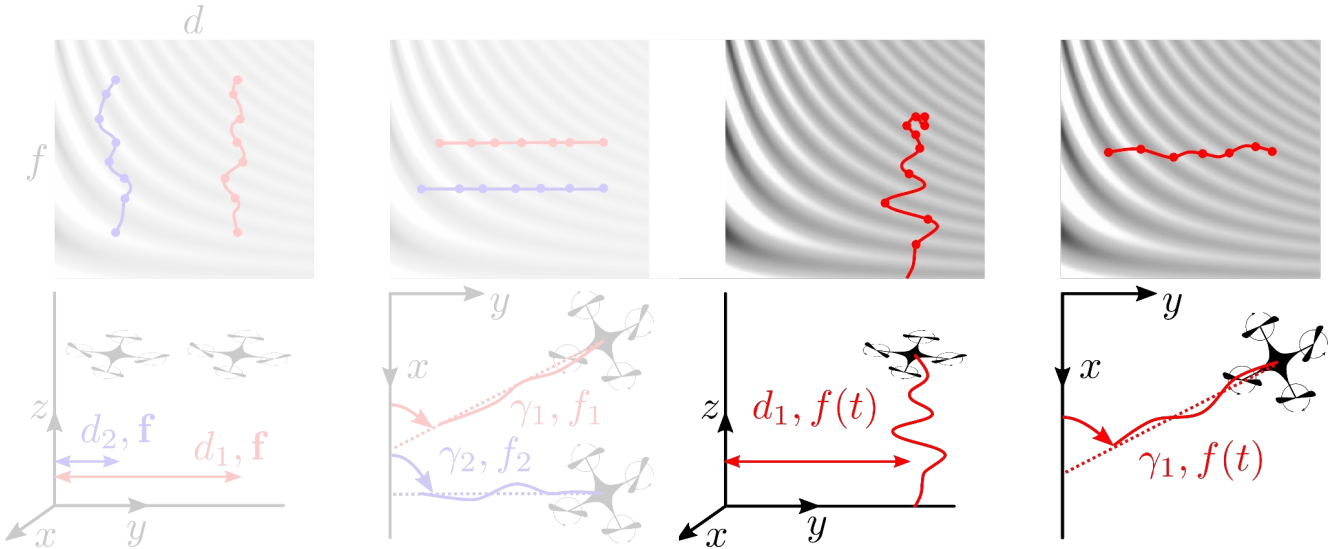


Open questions



theoretical limits?

Open questions



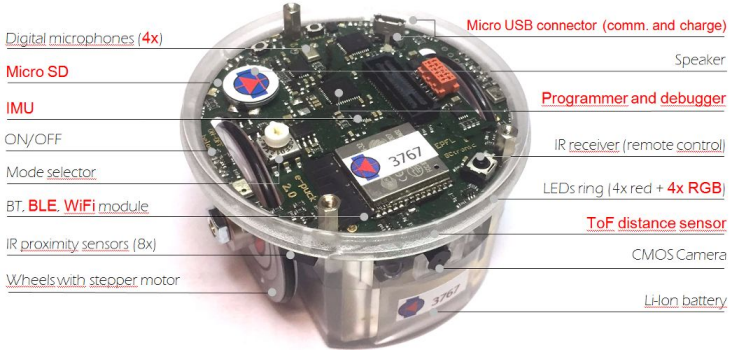
can hinderance be a friend?

Conclusion and future work

Continuous localization on drone?

https://www.decawave.com/product/dwm1000-module/

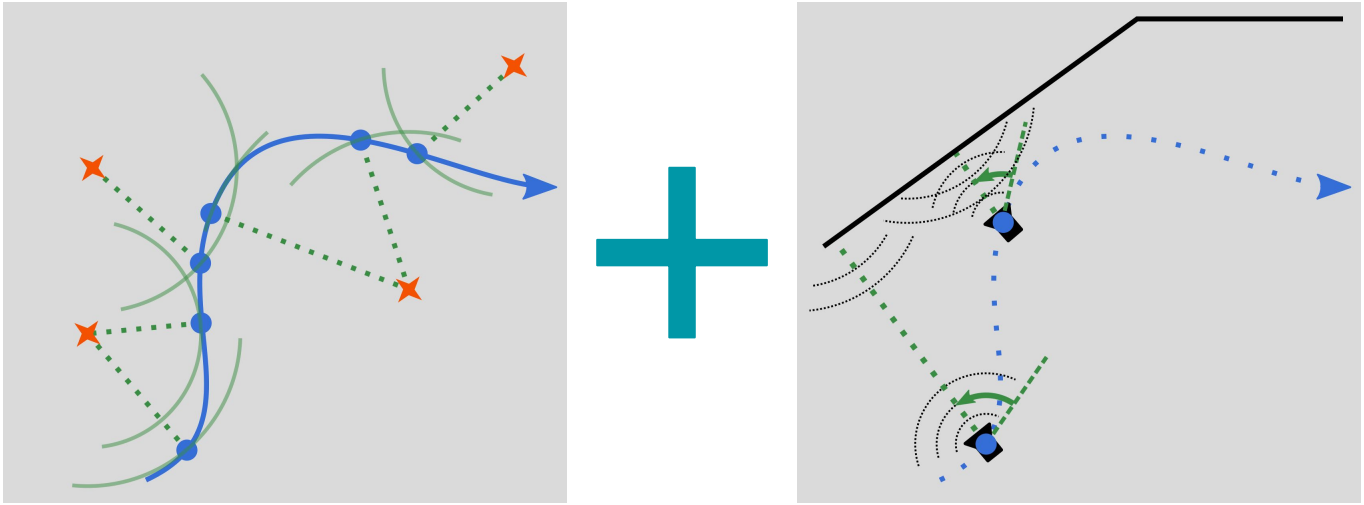
Wall detection on ground-based robot?



https://www.gctronic.com/doc/index.php/e-puck2

Conclusion and future work

Can we do continuous localization on a drone using audio signals?



Thank you!