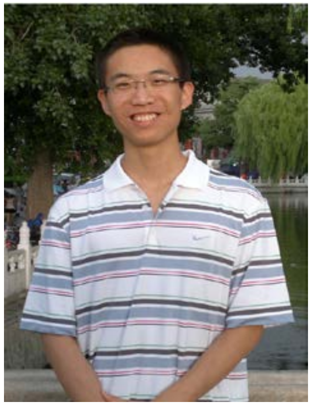
Localization of airborne vehicles in a GPS-denied environment

Brian D.O. Anderson
Hangzhou Dianzi University
The Australian National University
Data61-CSIRO

Thanks

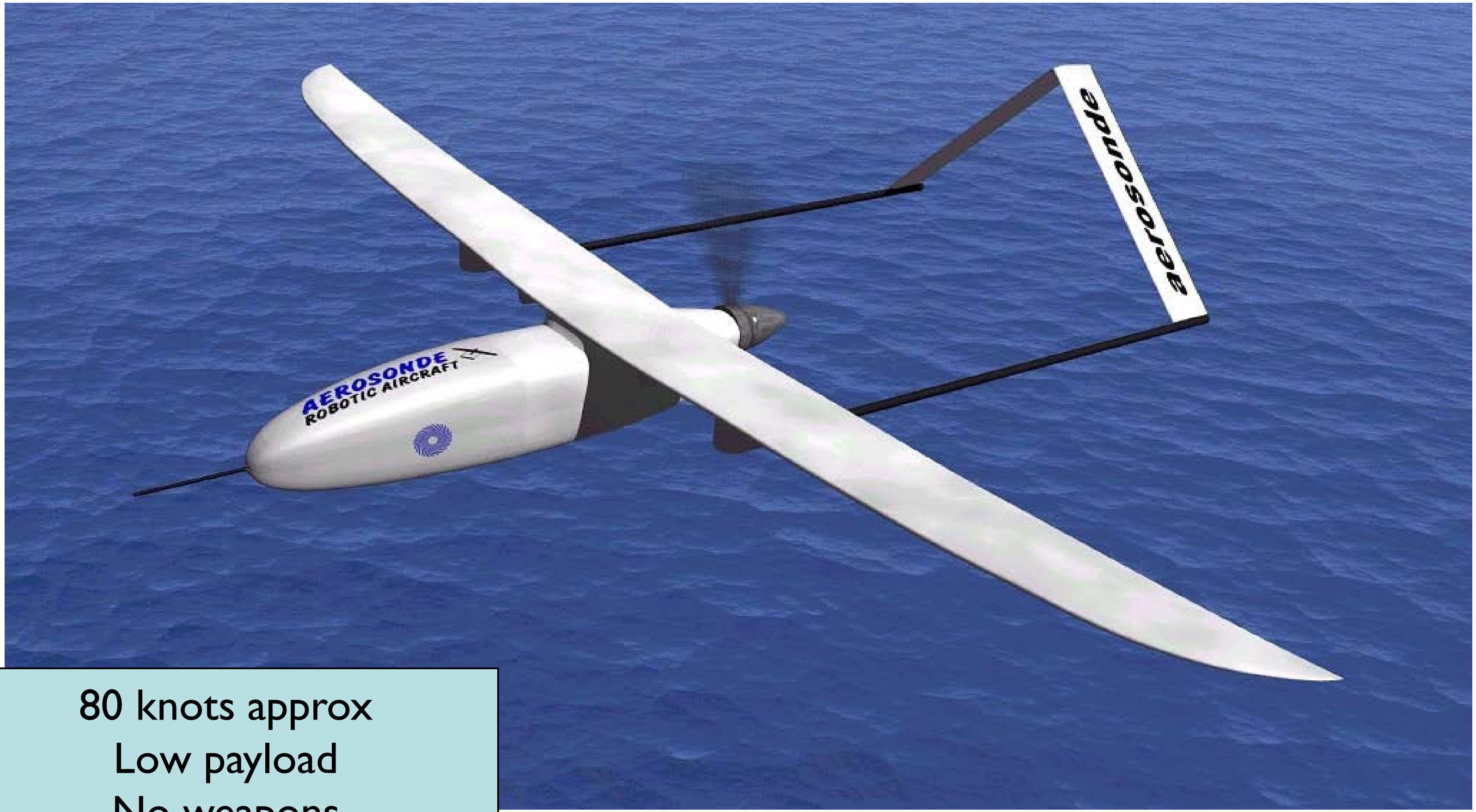
- To organizers for inviting me
- To colleagues involved in this project:
  - Bomin Jiang (MIT)
  - Hatem Hmam (Australia Defence Science and Technology Group)



Outline

- Problem Scenario
- Learning from Past Contributions
- First Solution using Semidefinite Programming
- Final Solution with Maximum Likelihood (with tests)
- Multiagent Problems
- Conclusions

UAVs



80 knots approx
Low payload
No weapons
Can cross Atlantic
2 meter wingspan

UAVs

Firemen may search a burning building using a canister of 100 micro airborne vehicles

Problem Scenario

- Situational context: a networked group of UAVs needing to localize their positions.
    - Example: Formations doing surveillance

- GPS is only available to a small number of agents in the network, possibly just one. (Building canyon operation, indoor operation, jamming, spoofing)

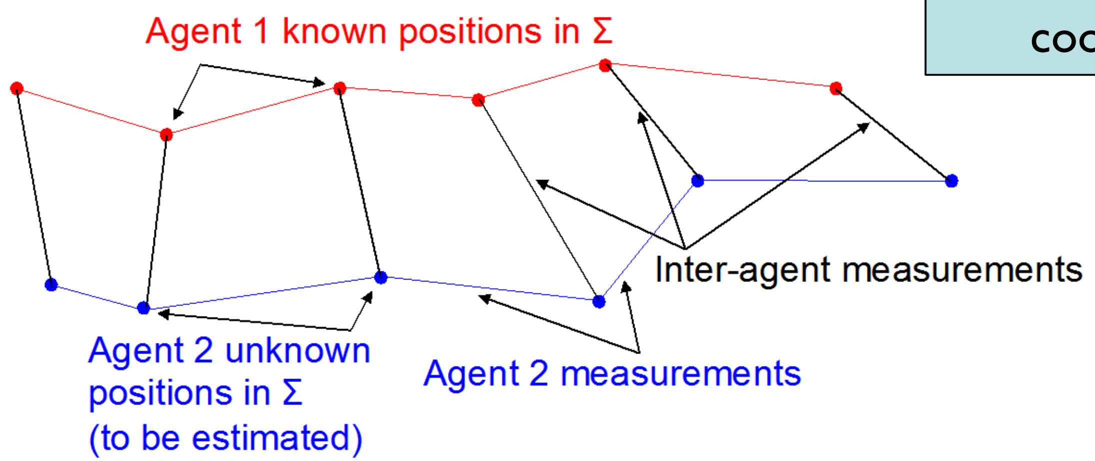
- The other agent ... h will drift over a long period, eve ...
    - Positions in global coordinates will be lost. Positions in local coordinates are consistent over a limited interval.

It is also relevant to look at using bearing-only measurements—but not here.

- Objective: **Payload is at a premium**!!! Seek to use inter-vehicle **distance-only measurements** to align each agent's local coordinate system with the global one.

Distance-only measurements in R^3

Σ denotes global coordinates



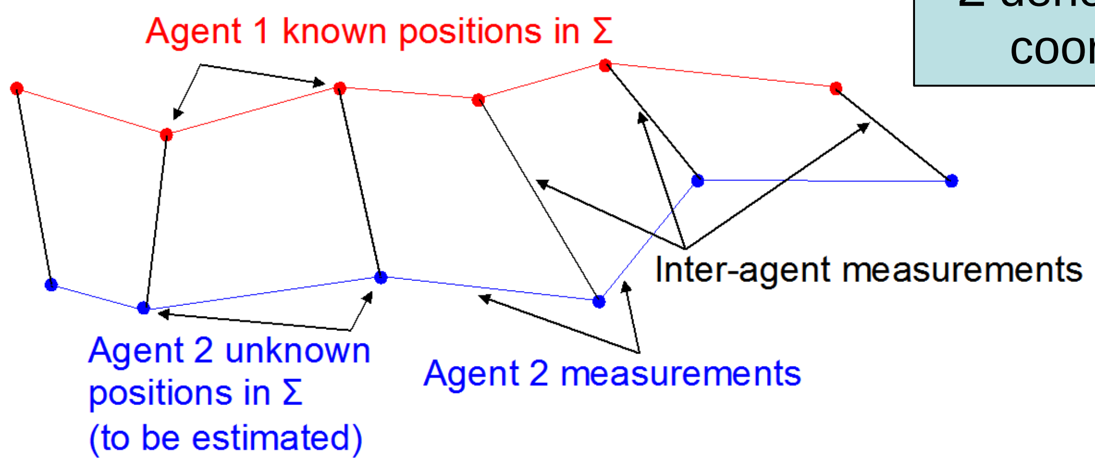
Inter-agent distance measurements may be noisy!

Agent 2 knows all waypoints and so has knowledge of its 'network' **in its own coordinate basis**

Treat two agent probblem first. Later treat multiple agents.

Distance-only measurements in R^3

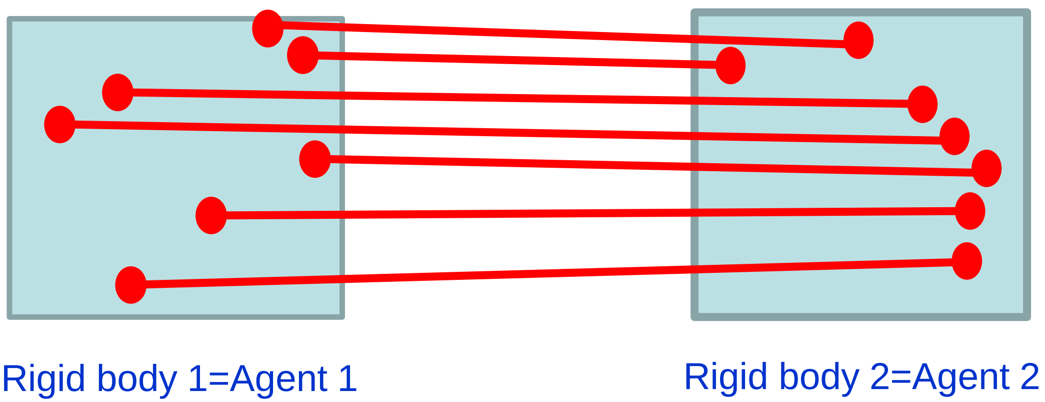
Σ denotes global coordinates



- Assumption: Measurement points on both trajectories form a rigid body of nonzero volume associated with each agent
- Knowledge of trajectories in consistent coordinate basis means: the two rigid bodies are specified up to congruence.

Treat two agent probblem first. Later treat multiple agents.

Distance-only measurements in R^3



- In global coordinates, rigid body 1 is localized; rigid body 2 is known up to congruence.
- Equivalently, rigid body 2 is localized in its own coordinate basis
- The two coordinate bases are related by a rotation matrix and translation vector.
- Question: can distance measurements fully localize rigid body 2?
  - How many will be needed?

Sensor Network View

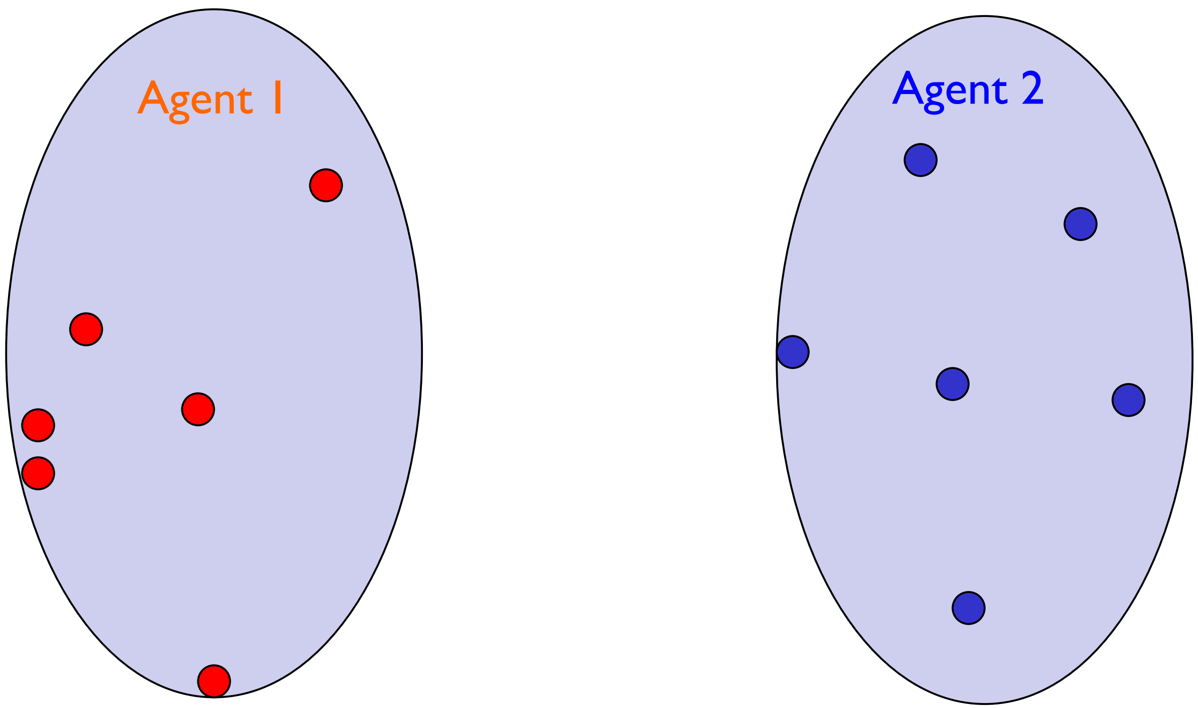
Agent 1 equivalent sensors are here

Agent 2 equivalent sensors are here

Agent 1 equivalence: Red
Sensor Network—fully localized globally

Agent 2 equivalence: Blue
Sensor Network—localized in own 'map' but could turn and translate as a whole

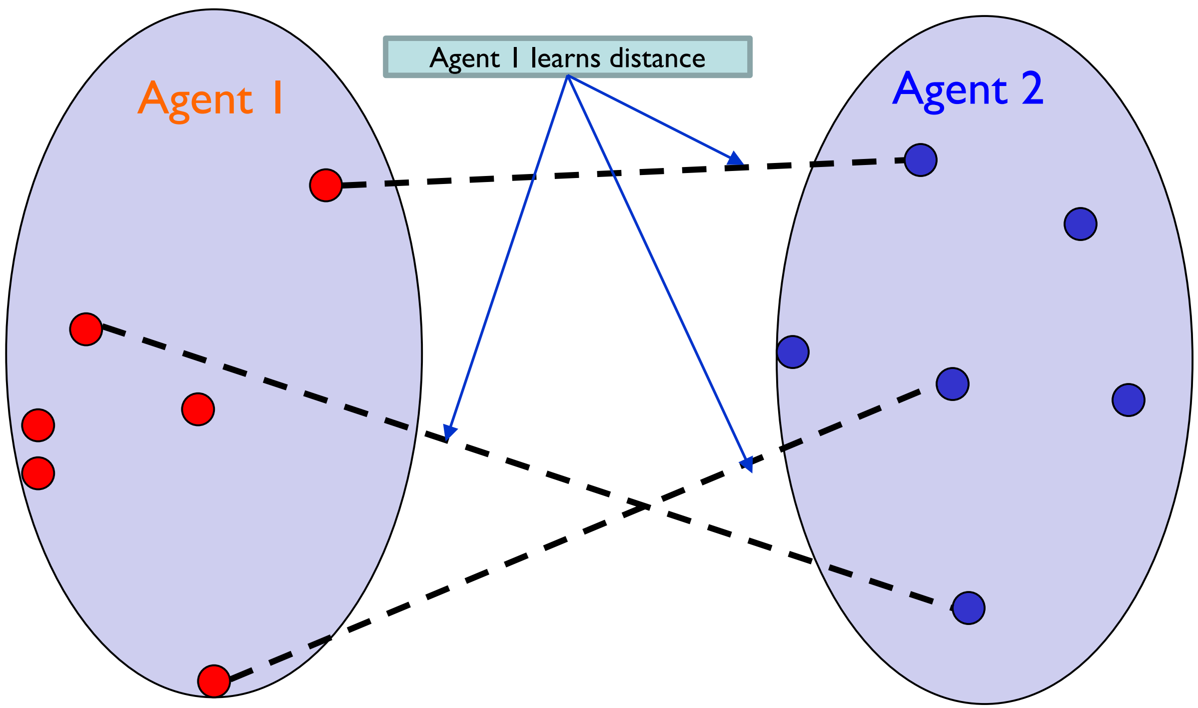
Sensor Network View-showing sensors



Sensor Network—fully localized globally

Sensor Network—localized in own 'map' but could turn and translate as a whole

Sensor Network View-showing sensors



Sensor Network—fully localized globally

Sensor Network—localized in own 'map' but could turn and translate as a whole

Problem scenario summary

- "Anchor sensors" (GPS-equipped agent at different times): know their location in global coordinates

- "Partially localized sensors" (GPS-denied agent at different times): know their location in local coordinates only

- Distance measurements: available between sensors

- Objective: **alignment of coordinate systems=identification of rotation matrix and translation vector**

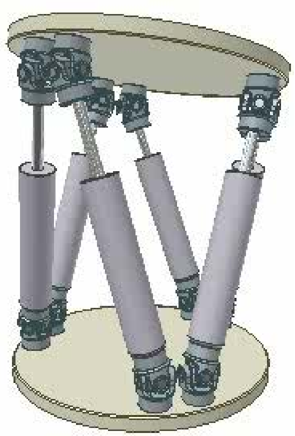
Outline

- Problem Scenario
- Learning from Past Contributions
- First Solution using Semidefinite Programming
- Final Solution with Maximum Likelihood (with tests)
- Multiagent Problems
- Conclusions

Some history: Stewart-Gough

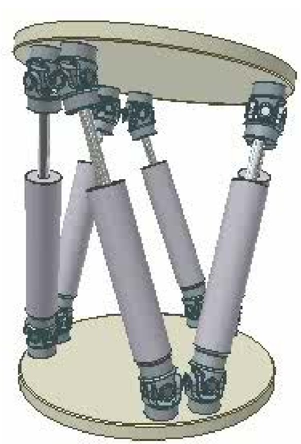
- Stewart-Gough platform—1966
- Two platforms: ground platform, moving platform
- Moving platform has 6 degrees of freedom
- 6 prismatic actuators

D. Stewart, “A platform with six degrees of freedom,” Aircraft Engineering and Aerospace Technology, vol. 38, no. 4, pp. 30–35, 1966



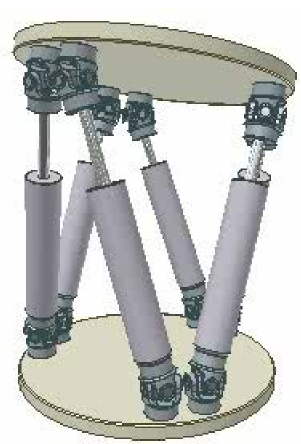
Some history: Stewart-Gough

- 3D localization problem for upper platform.
- Two coordinate systems—global, body-centered for upper.
- Localization problem has 6 degrees of freedom
- 6 distance measurements



Some history: Stewart-Gough

- 6 distance measurements
- **CENTRAL QUESTION: How can one obtain orientation and centroid position of upper platform from distance measurements?**



Stewart-Gough Platform

- Suppose:
  - $p_x = [x_1, x_2, x_3]^\top$ is the position of an 'anchor agent' in the global coordinates. (known)
  - $p_y = [y_1, y_2, y_3]^\top$ is the position of an agent localized in the local coordinates to which the anchor agent is connected. (known)
  - $R = \{r_{ij}\}$ and $T = [t_1, t_2, t_3]^\top$ are the rotation matrix and translation from global coordinates to local coordinates. (unknown)
  - z is the measured distance between the two connected agents. (known)

Stewart-Gough Platform

- Suppose:
  - $p_x = [x_1, x_2, x_3]^\top$ is the position of an 'anchor agent' in the global coordinates. (known)
  - $p_y = [y_1, y_2, y_3]^\top$ is the position of an agent localized in the local coordinates to which the anchor agent is connected. (known)
  - $R = \{r_{ij}\}$ and $T = [t_1, t_2, t_3]^\top$ are the rotation matrix and translation from global coordinates to local coordinates. (unknown)
  - z is the measured distance between the two connected agents. (known)
- There holds for each distance measurement (length of connection):

$$z = \|Rp_y + T - p_x\|$$

  - *R* is 3×3 orthogonal, *T* is 3-vector (6 parameters in all)

Stewart-Gough Platform

- Suppose:
  - $p_x = [x_1, x_2, x_3]^\top$ is the position of an 'anchor agent' in the global coordinates. (known)
  - $p_y = [y_1, y_2, y_3]^\top$ is the position of an agent localized in the local coordinates to which the anchor agent is connected. (known)
  - $R = \{r_{ij}\}$ and $T = [t_1, t_2, t_3]^\top$ are the rotation matrix and translation from global coordinates to local coordinates. (unknown)
  - z is the measured distance between the two connected agents. (known)

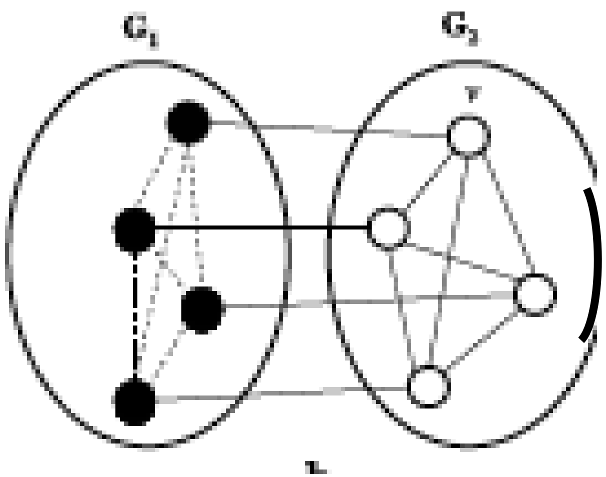
- There holds for each distance measurement (length of connection):

$$z = ||Rp_y + T - p_x||$$

- Finding *R,T* using six distance measurements gives rise to fourtieth order polynomial equation. One solution is correct!
  - Could hope to disambiguate with seven measurements, but could not freely set lengths of seven rods. System is overdetermined.
  - Side remark: 9 measurements and planar platform is easy.

2D sensor network merging problem

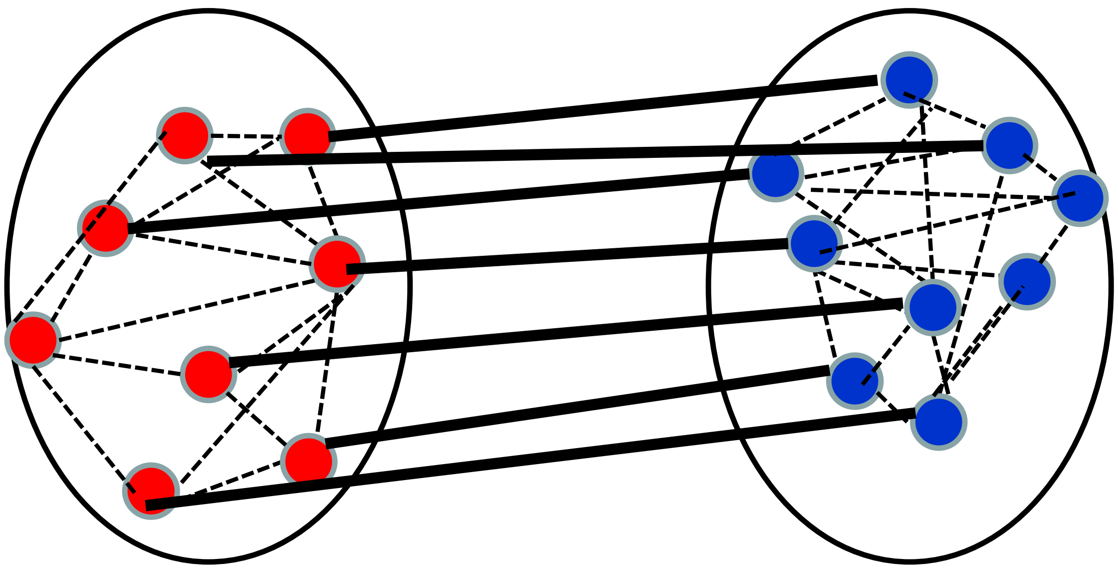
- Consider two **globally rigid** networks G_1, G_2 in 2D (globally rigid = given distances determine map up to a congruence)
- G_1 localized in global coordinates, G_2 in own coordinates.
- **Four** or more **inter-network** distance measurements available.
- **FACT:** one can localize whole network.
- Construction of algorithm is tricky.
  - Relied on geometry of four-bar chain.



S. Motevallian, L. Xia, and B. Anderson, "A new splitting-merging paradigm for distributed localization in wireless sensor networks," in IEEE International Conference on Communications, June 2013, pp.1454–1458.

Formation merging in R^3

- Data: Two globally rigid formations in R^3



C. Yu, B. Fidan, and B. D. Anderson, "Principles to control autonomous formation merging," in Proceedings of the American Control Conference. IEEE, 2006, pp. 7–pp.12.

- How many edges need to be added to make the whole merged formation globally rigid? (Globally rigid means formation shape is determined up to congruence)

Formation merging in R^3

- Problem: How many edges need to be added to make the whole formation globally rigid? Can allow for common vertices.
- Principles for merging set out
- > 2 formations also studied

C. Yu, B. Fidan, and B. D. Anderson, "Principles to control autonomous formation merging," in Proceedings of the American Control Conference. IEEE, 2006, pp. 7–pp.12.

TABLE I

OPTIMAL MERGING OF TWO GLOBALLY RIGID FORMATIONS

Dim.	$\lvert V_c \rvert$	$\lvert E_{new} \rvert$	$\lvert V_1^{new} \rvert = \lvert V_2^{new} \rvert$
$\Re^2$	0	4	3
$\Re^2$	1	2	2
$\Re^2$	2	1	1
$\Re^2$	3 or more	0	0
$\Re^3$	0	7	4
$\Re^3$	1	4	3
$\Re^3$	2	2	2
$\Re^3$	3	1	1
$\Re^3$	4 or more	0	0

Frame orientation of two robots

- Two robots (= rigid bodies) are given, with distances between different point pairs, one on each robot.
- Describe second robot by centroid (3 variables) and frame orientation (quaternion, 4 variables, unit length constraint).
- Obtain polynomial system in unknowns with 40 solutions given six measurements. Each equation has linear and quadratic terms.
- Hard to solve with > 6 measurements, especially given noise.
- By treating some of the quadratic terms as independent variables, and by using 10 measurements, an easily solved set of equations results.
- Solution can be used as initialization of a weighted least squares algorithm for use in noisy situation.

- Simulation results show **even a small amount of noise may be fatal.**

N. Trawny, X. S. Zhou, K. X. Zhou, and S. I. Roumeliotis, "3d relative pose estimation from distance-only measurements," in IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2007, pp. 1071–1078.

Summary of what is known

- Solvability of (2-agent) localization problem requires at least 7 measurements
- Very little insight into algorithms, even for the noiseless case
- Noisy case gives overdetermined set of equations which have no solution generically.

Outline

- Problem Scenario
- Learning from Past Contributions
- First Solution using Semidefinite Programming
- Final Solution with Maximum Likelihood (with tests)
- Multiagent Problems
- Conclusions

Setting up an SDP problem 1

- Suppose:
  - $p_x = [x_1, x_2, x_3]^\top$ is the position of an GPS-equpped agent in the global coordinates. (known)
  - $p_y = [y_1, y_2, y_3]^\top$ is the position of to-be-localized agent in its own (INS) coordinates. (known)
  - $R = \{r_{ij}\}$ and $T = [t_1, t_2, t_3]^\top$ are the rotation matrix and translation from global coordinates to INS coordinates. (unknown)
  - z is the measured distance between the two connected agents.:(known)

$$z = \|Rp_y + T - p_x\|$$

Setting up an SDP problem 1

- Suppose:
  - $p_x = [x_1, x_2, x_3]^\top$ is the position of an GPS-equpped agent in the global coordinates. (known)
  - $p_y = [y_1, y_2, y_3]^\top$ is the position of to-be-localized agent in its own (INS) coordinates. (known)
  - $R = \{r_{ij}\}$ and $T = [t_1, t_2, t_3]^\top$ are the rotation matrix and translation from global coordinates to INS coordinates. (unknown)
  - z is the measured distance between the two connected agents. (known)
- Use squared version of $z = \|Rp_y + T - p_x\|$ and the quadratic equations satisfied by entries of *R*. Then with

$$\Theta = [\theta_1, \theta_2, \ldots, \theta_{16}]$$

where the θ_l are unknowns include all r_{ij}, t_i, together with

$$\sum r_{i1}t_i, \sum r_{i2}t_i, \sum r_{i3}t_i, \text{ and } \sum t_i^2$$

there results for *k*-th measurement, with known row-vector, scalar *A(k), b(k)*

$$A(k)\Theta = b(k)$$

Setting up an SDP problem 2

- Use squared version of $z = \|Rp_y + T - p_x\|$ and the quadratic equations satisfied by entries of *R*. Then with

$$\Theta = [\theta_1, \theta_2, \ldots, \theta_{16}]$$

where the θ_l are unknowns include all r_{ij}, t_i, together with

$$\sum r_{i1}t_i, \sum r_{i2}t_i, \sum r_{i3}t_i, \text{ and } \sum t_i^2$$

there results for the *k*-th measurement, with known row-vector, scalar *A(k), b(k)*

$$A(k)\Theta = b(k)$$

Setting up an SDP problem 2

- Use squared version of $z = \|Rp_y + T - p_x\|$ and the quadratic equations satisfied by entries of *R*. Then with

$$\Theta = [\theta_1, \theta_2, \dots, \theta_{16}]$$

where the θ_l are unknowns include all r_{ij}, t_i, together with

z

$$\sum r_{i1}t_i, \sum r_{i2}t_i, \sum r_{i3}t_i, \text{ and } \sum t_i^2$$

there results for the *k*-th measurement, with known row-vector, scalar *A(k), b(k)*

$$A(k)\Theta = b(k)$$

- Collect all the measurement equations together, using matrix and vector *A*,*b*:

$$A\Theta = b$$

- Only 6 of the entries of Θ are functionally independent. There are 10 constraints.
- With 16 measurements, no noise and *A* nonsingular, we obtain Θ and thus *R* and *T*.
- This equation assumes no noise. But we have noise!
- We need: **a least squares solution reflecting the constraints on Θ entries!**

Setting up an SDP problem 3

- Collect all the measurement equations together, using matrix and vector *A*, *b*:

$$A\Theta = b$$

- Only 6 of the entries of Θ are functionally independent. There are 10 constraints.
- We need: **a least squares solution reflecting the constraints on Θ entries!**
- **WONDERFUL FACT: The constraints on the entries of *Θ* are all quadratic**
  - Θ contains entries of *R*,*T*.
  - Some constraints come from the orthogonal matrix constraint.
  - Remainder come from inclusion as entries $\sum r_{i1}t_i, \sum r_{i2}t_i, \sum r_{i3}t_i, \text{ and } \sum t_i^2$
- So we have a quadratic minimization problem:

Choose Θ to minimize $||A\Theta - b||^2$ subject to some quadratic constraints.

- This is **almost standard.**

Setting up SDP Problem 4

Define

$$X = \begin{bmatrix} \Theta \\ 1 \end{bmatrix} \begin{bmatrix} \Theta \\ 1 \end{bmatrix}^\top$$

Then

$$X \succcurlyeq 0$$

$$\text{rank } X = 1$$

$$X_{17,17} = 1$$

Note that $\boldsymbol{\Theta}$ determines X uniquely and X determines $\boldsymbol{\Theta}$ uniquely!

- The objective function $||\boldsymbol{A\Theta} - \boldsymbol{b}||^2$ can be written for some P as

$$\langle P, X \rangle$$

- The i-th quadratic constraint can be written for some Q_i as

$$\langle Q_i, X \rangle = q_i$$

The SDP Problem itself

$$\begin{aligned} \text{Find } \operatorname{argmin}_X \langle P, X\rangle \\ \text{subject to } \langle Q_i, X\rangle = q_i \\ X_{17,17} = 1 \\ X \succcurlyeq 0 \\ \operatorname{rank} X = 1 \end{aligned}$$

- Finding *X* is equivalent to finding ***Θ***
- Rank constraint means it is not strictly SDP
- **Relaxation: temporarily forget rank constraint, use standard SDP**

How can we deal with rank constraint?

- In most cases it holds automatically
- You can always get a rank 2 solution
- Use SVD and approximate *X* with nearest rank 1 matrix

A side issue with approximating

- If X is obtained by SVD of an SDP solution, it may not satisfy the quadratic constraints exactly.
- Then the entries of $\boldsymbol{\Theta}$ yielding the rotation matrix entries may not correspond to an orthogonal matrix.
- This is easily fixed by a **Procrustes algorithm,** which finds closest orthogonal matrix with determinant =1 to a given matrix.
- In summary, SDP, [perhaps with relaxation, SVD approximation and Procrustes], gives values for R,T.

Outline

- Problem Scenario
- Learning from Past Contributions
- First Solution using Semidefinite Programming
- Final Solution with Maximum Likelihood (with tests)
- Multiagent Problems
- Conclusions

Maximum Likelihood 1

- The performance function for SDP is not related to the noise—it is mathematically convenient only.
- If the noise contaminating distance measurements has known probability density, the gold standard solution is the **maximum likelihood values** of R, T.
- These minimize a different performance index.
- Analytic minimization is impossible. Typically there are multiple minima. **Good initial guess is critical.**
- **Use SDP values to initialize gradient descent ML algorithm.**

Maximum likelihood minimization 2

- Assumption: Inter-vehicle distance measurements are contaminated with zero mean white noise of variance σ^2. Position measurements are noiseless.
- Instead of $z = ||Rp_y + T - p_x||$ there holds

$$\tilde{z}(k) = ||Rp_y + T - p_x|| + \xi, \xi \sim N(0, \sigma^2)$$

- We aim to solve

$$\tilde{R}, \tilde{T} = \arg\min_{R,T} \sum_{k=1}^{N} (\tilde{z}(k) - ||Rp_y(k) + T - p_x(k)||)^2$$

$$\text{subject to } RR^\top = I \;\; detR = 1$$

Maximum Likelihood Estimate 3

$$\tilde{R}, \tilde{T} = \arg\min_{R,T} \sum_{k=1}^{N} (\tilde{z}(k) - ||Rp_y(k) + T - p_x(k)||)^2$$

$$\text{subject to } RR^\top = I \quad detR = 1$$

- Computing gradient is simple, and implementing gradient is simple, **except** for the orthogonality constraint.
- We need gradient descent **on a manifold**.
- This works by computing the usual gradient at a point on the manifold and then **projecting onto the tangent space.**

Maximum Likelihood Estimate 4

$$\tilde{R}, \tilde{T} = \arg\min_{R,T} \sum_{k=1}^{N} (\tilde{z}(k) - ||Rp_y(k) + T - p_x(k)||)^2$$

$$\text{subject to } RR^\top = I \quad detR = 1$$

- This works by computing the usual gradient at a point on the manifold and then **projecting onto the tangent space.**
- Suppose *M* is gradient w.r.t. *R* of the index at *R,T*.
- The projection onto the tangent space is:

$$M_T = \frac{1}{2}M - \frac{1}{2}RM^\top R$$

Maximum Likelihood Estimate 5

$$\tilde{R}, \tilde{T} = \arg\min_{R,T} \sum_{k=1}^{N} (\tilde{z}(k) - ||Rp_y(k) + T - p_x(k)||)^2$$

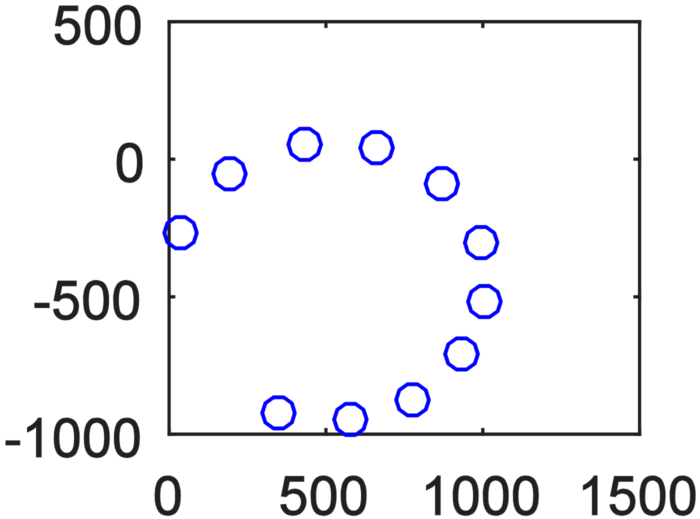
$$\text{subject to } RR^\top = I \;\; detR = 1$$

- A gradient descent algorithm on a smooth manifold (as here) **almost always converges** to a minimum—**but** it might be a local minimum.
- Try to get to global minimum by initializing using SDP solution.
- Systematic and small correction for bias can be made. (improvement of about 3% in simulations)

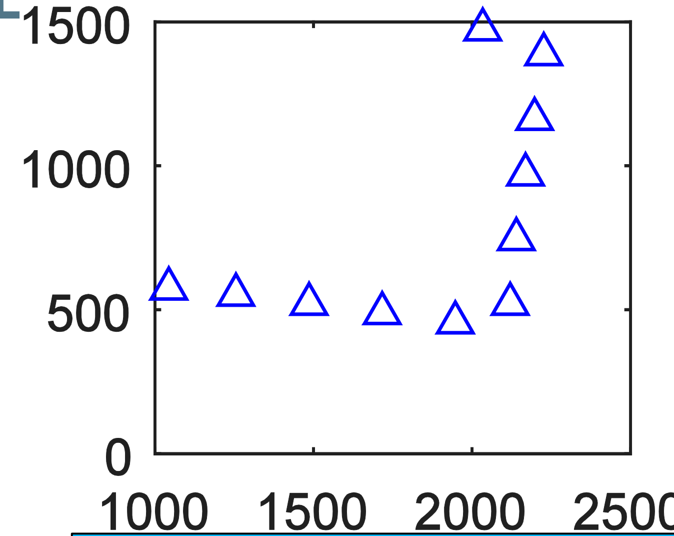
Summary

- Regard localization problem as one of finding a rotation matrix R and translation vector T.
- Establish a least squares problem with unknowns including the entries of R, T.
- Solve using SDP.
  - If necessary, relax, and tidy up with Procrustes
- Use SDP solution to initialize gradient descent algorithm for MLE
  - Gracefully handle gradient descent on a manifold.

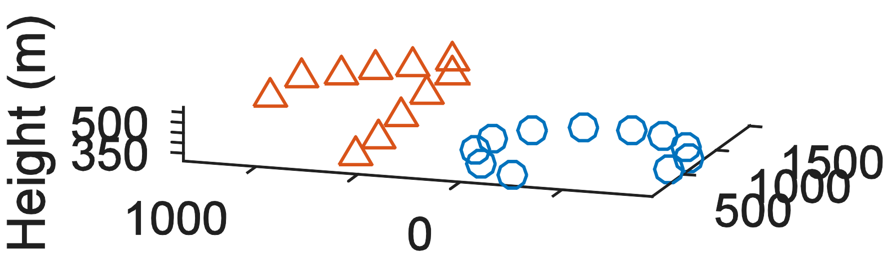
Real data and outcome 1



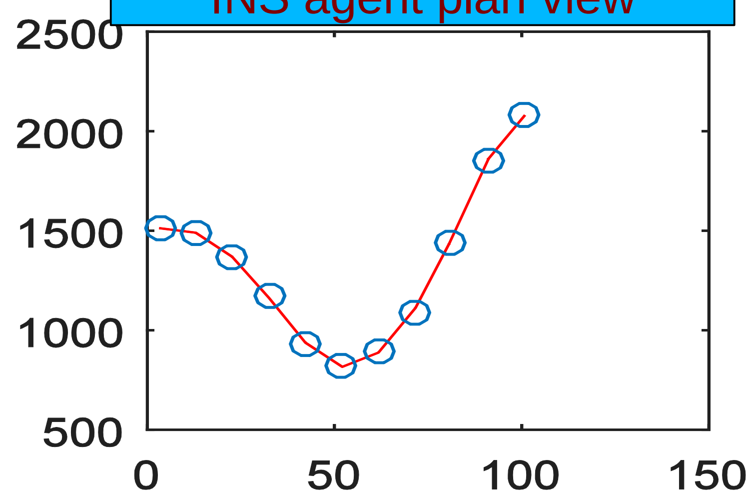
Anchor agent plan view



INS agent plan view

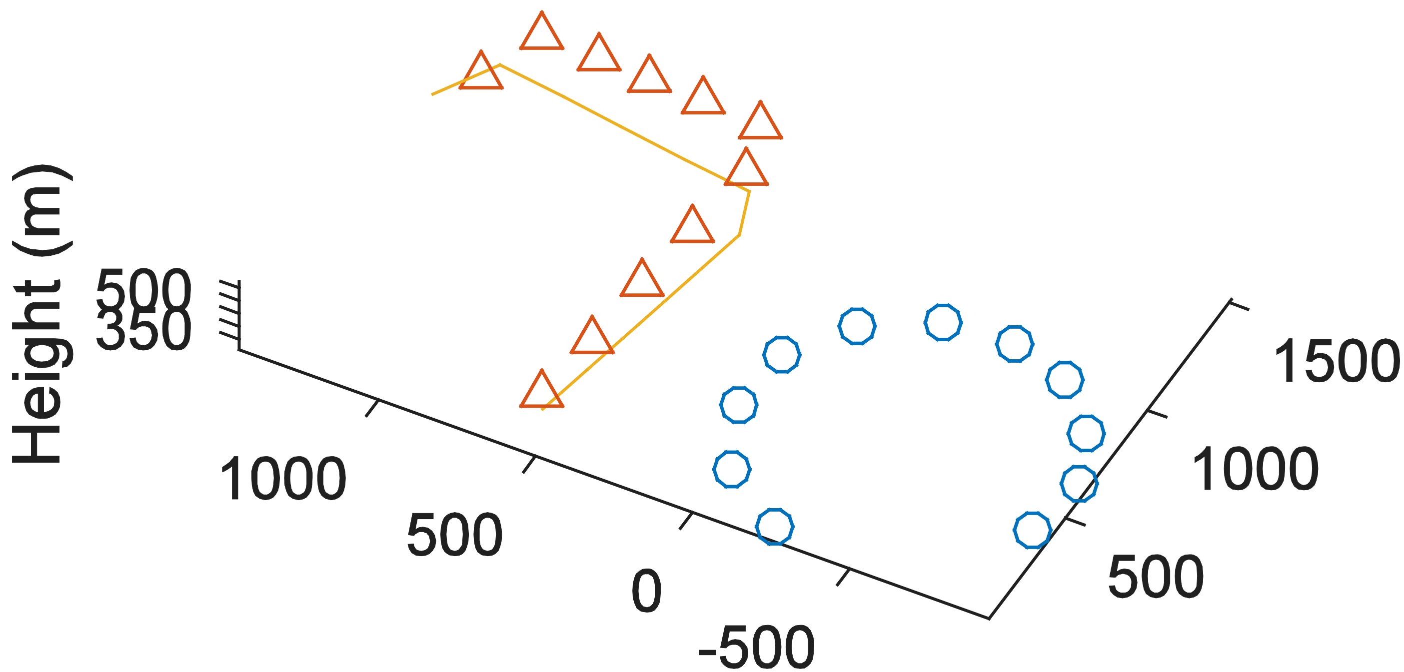


3D estimated trajectories



Reconstructed Interagent distance

Real data and outcome 2



Circles: GPS-equipped UAV.
Triangles: 'recovered' positions of GPS-denied UAV.
Solid line: actual trajectory of GPS-denied UAV.

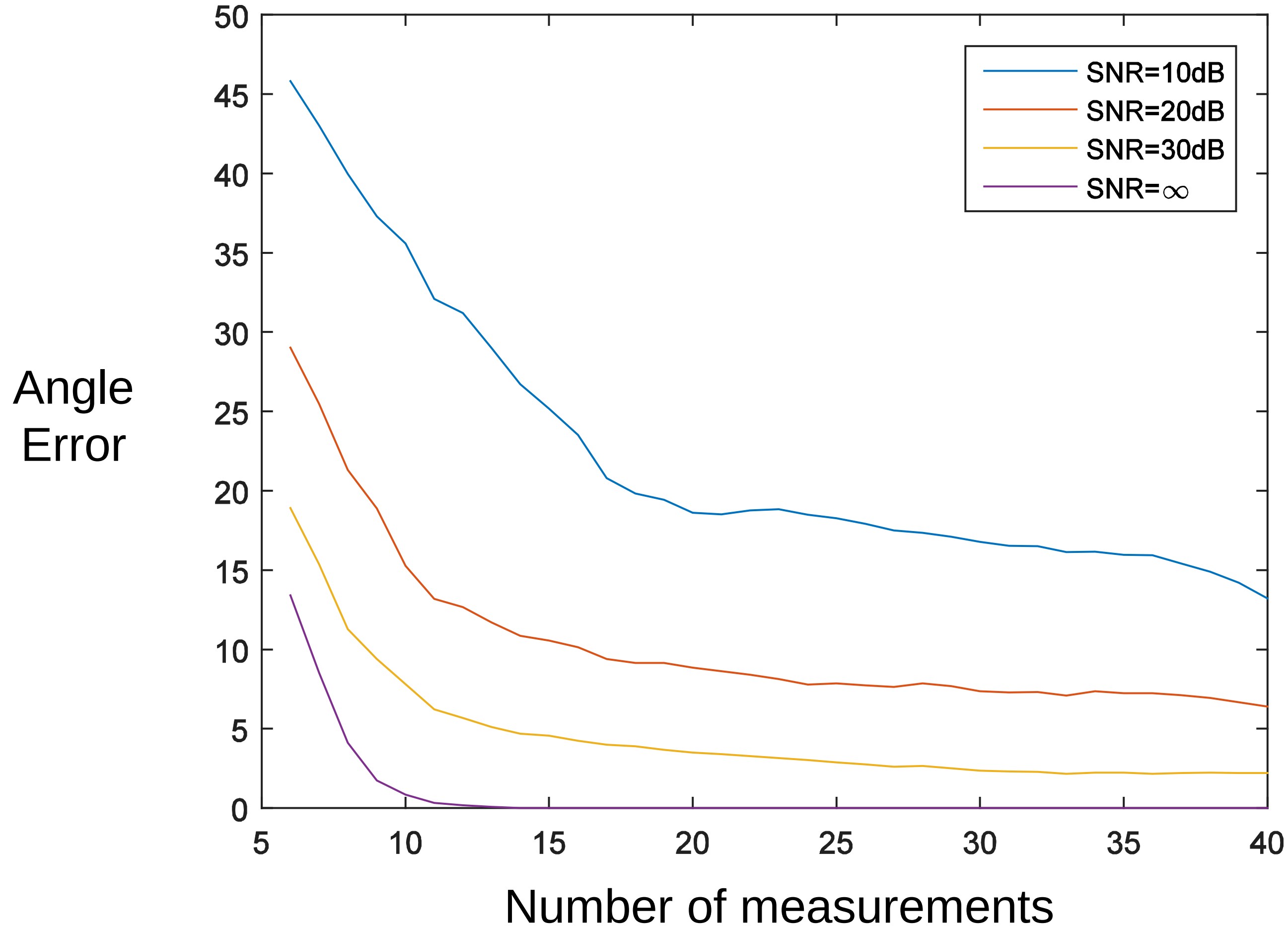
Real data and simulations

- Further simulations show proposed method works with 10 dB SNR and only 7 measurements.

- Performance shows a balance between number of measurements (quantity of measurements) and SNR (quality of measurements)

Real data and simulations

Angle error in degrees vs. number of measurements: SNR=inf, 30, 20 and 10



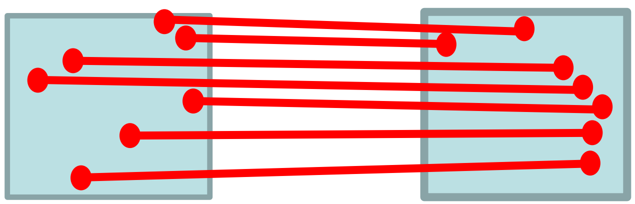
Outline

- Problem Scenario
- Learning from Past Contributions
- First Solution using Semidefinite Programming
- Final Solution with Maximum Likelihood (with tests)
- **Multiagent Problems**
- Conclusions

Multiagent Problems: The key questions

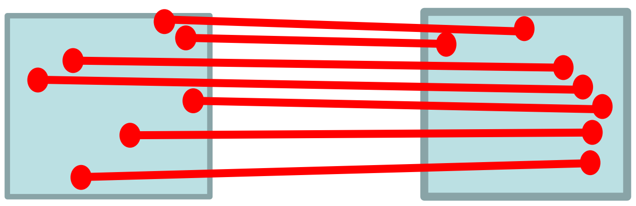
- Given $N>2$ agents, where one has GPS, and the others INS, what are the rules on the number of distance measurements and their locations to ensure global rigidity (uniqueness of the agent locations)?
- What sort of algorithms can be used to achieve localization, i.e. alignment of coordinate bases?
  - Can the algorithms be made recursive (localize agents one at a time)?

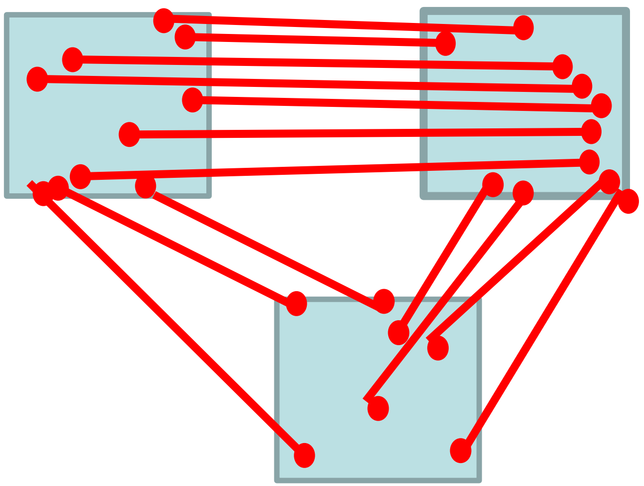
Body Bar Graphs



- “Body” is 3-dimensional rigid body, formed from waypoints of one agent. (Waypoint: where distance measurement is taken)
- “Bars” correspond to distance measurements **between** agents.

Body Bar Graphs





- "Body" is 3-dimensional rigid body, formed from waypoints of one agent. (Waypoint: where distance measurement is taken)
- "Bars" correspond to distance measurements **between** agents.

- Sample questions:
  - Is this three-agent body-bar graph globally rigid?
  - Equivalently, if one agent has GPS and the other two INS, can we localize them?
  - How would we localize?

Counting results are available.

Minimal Rigidity (1984)

- With k rigid bodies, need 6(k-1) bars and no more than 6(m-1) between any subset of m

Global rigidity (1984)

- Set up is redundantly rigid, i.e. remains rigid (=minimally rigid plus extra bars allowed) if remove any single bar

Globally rigid building (2013)

All globally rigid can be built using following steps:

- Add an edge anywhere
- Add a new body with 7 joining edges to other bodies
- Break k edges, $k \in \{1,2,3,4,5\}$ and connect breakpoints to a new body together with 6-k edges to old bodies (adds in all 6 edges)

Example of Step 3 (Connelly et al 2013)

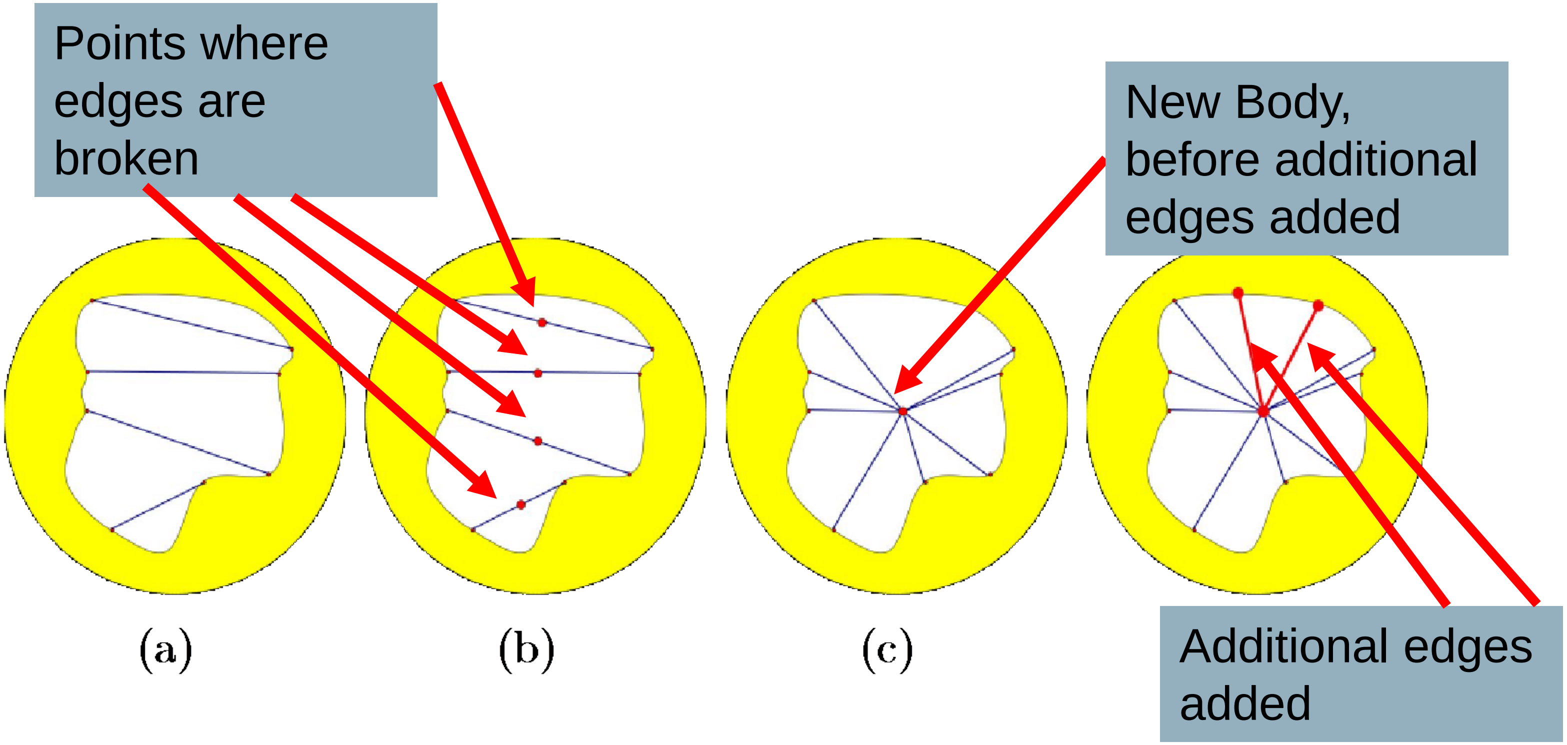


Figure 2: A 6-split on 4 edges. The chosen edges (a), the pinch (b,c), and the addition of 2 edges (d).

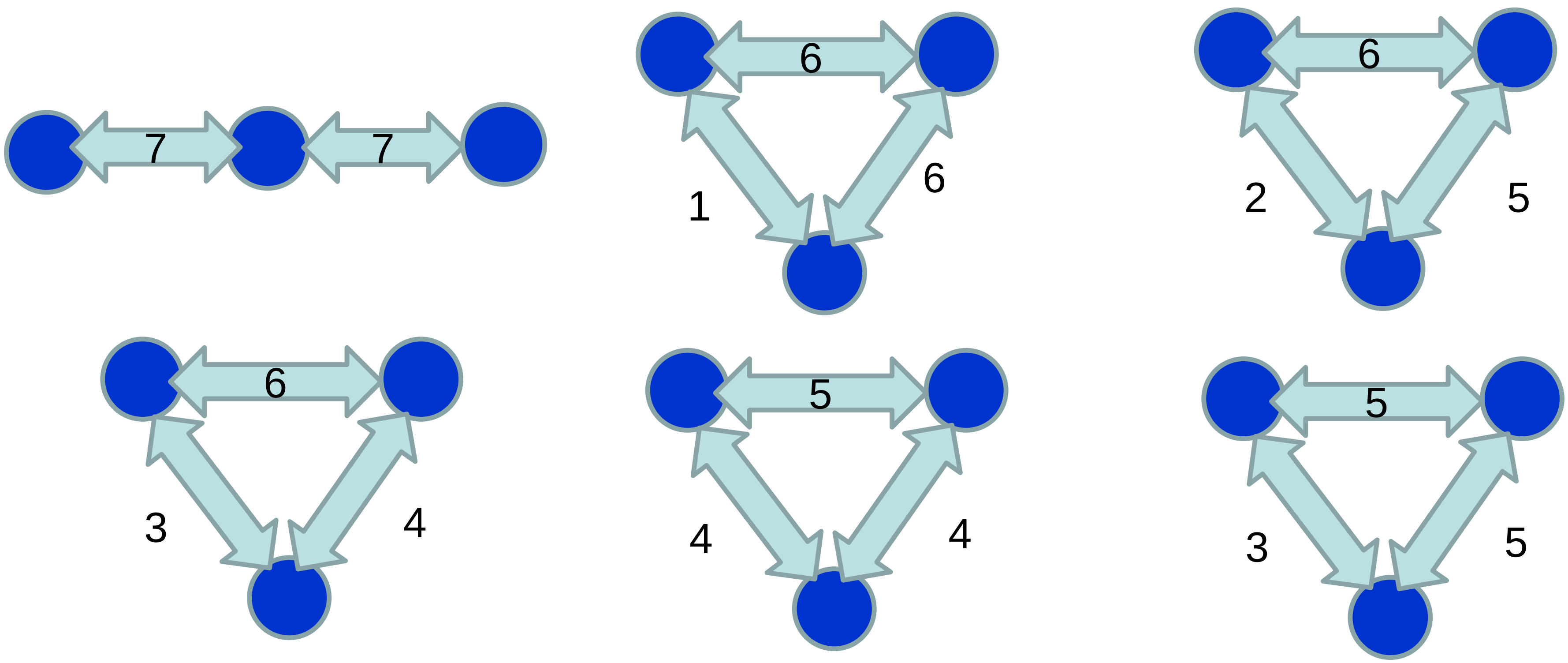
Three body (3 agent) case

- Counting conditions yield easily the following:

For a bar-body graph with three bodies, necessary and sufficient conditions for global rigidity are:

1. Each body has at least 7 connected bars
2. There are at least 13 bars in all.

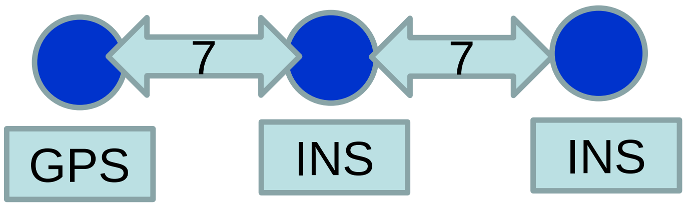
Three body (three agent) case

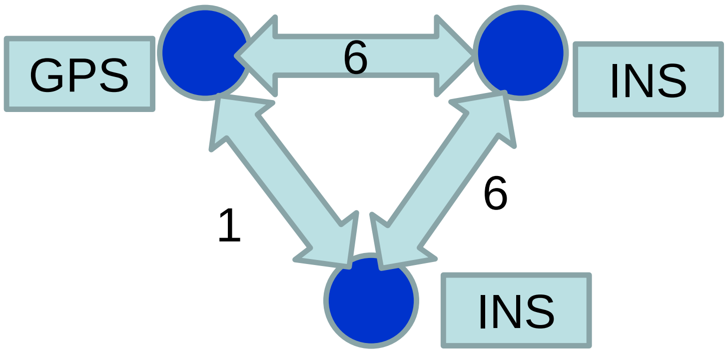


In every case, more bars could be added. In no case can any be removed.

Each body must have at least 7 incident edges; total count must be at least 13.

Three body (three agent) case





- Recursive localization possible using two-agent scheme

- **Cannot recursively localize**
- Use 2 agent approach but have larger dimension

Three agent localization problem

- Assume one agent has GPS. Other two described by rotations and translations R_i, T_i for i=1,2.
- Collect all the measurement equations together, using matrix and vector A, b:

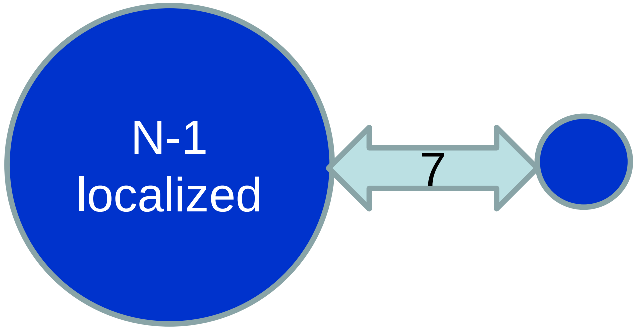
$$A\Theta = b$$

- Only 12 of the 32 entries of Θ are functionally independent. There are 20 **quadratic** constraints, 12 from orthogonality and 8 from new quantities.
- Hence SDP can (in principle) be used. MLE can follow.
- Dimensionality explosion (between linear and quadratic in agent number) if increase agent number even more. (Consistent with sensor network localization problems.)

Recursive Localization for special networks

N -1 agents are localized

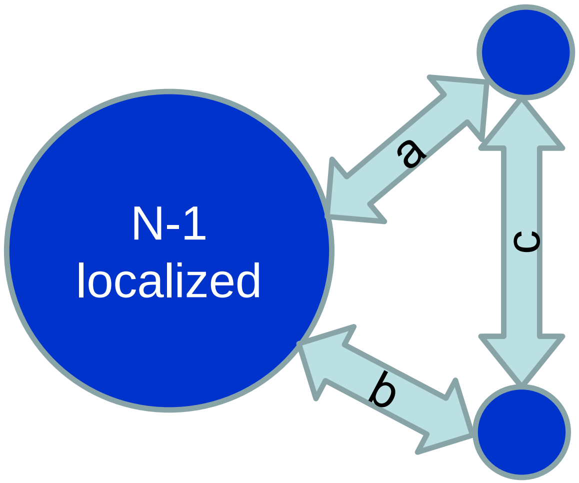
- **One** additional agent is to be localized in one step.
- In Body-Bar view, suppose it is connected by **7 or more** bars to any of N-1 already localized.
- Regarding the N-1 localized agents as a "super-agent", agent N can be treated as the second of a two-agent group.



Recursive Localization for special networks

N -1 agents are localized

- **Two** additional agents are to be localized in one step.
- In body-bar view, each of two agents must have 7 or more incident bars, N-1 agents collectively must have 7 or more bars to the new 2, and total bars incident on new 2 must be at least 13.
- Again regard the N-1 localized agents as a “super-agent”, and do three-agent localization.



$$a + b + c \geq 13$$

$$a + b \geq 7$$

$$a + c \geq 7$$

$$b + c \geq 7$$

Outline

- Problem Scenario
- Learning from Past Contributions
- First Solution using Semidefinite Programming
- Final Solution with Maximum Likelihood (with tests)
- Multiagent Problems
- Conclusions

Key Messages 1

- Localization of GPS-denied agents is a practical issue
- It is desirable to solve it with minimum sensing modes
- Practical solutions require 7 inter-agent distance measurements while INS frame of reference is stable
- Noisy measurements can be accommodated
- Key algorithm components include:
  - Semidefinite programming
  - Maximum likelihood estimation via gradient descent on a manifold
  - Procrustes algorithm for obtaining orthogonal matrix
  - Systematic bias reduction

Key Messages 2

- Two agents certainly can be handled. Three agents can probably be handled
- Four or more agents can probably only be handled for those graphs which can be built up by adding one or two agents per step to an already localized system of agents
- There are special (nongeneric) trajectories for which localization is not possible.

Future Work 1

- Same problem but with bearing-only measurements.
- It appears easier!
- For two agent case, agent A has GPS, transmits its position to agent B, who direction-finds in its basis.
- Use same approach of seeking rotation matrix *R* and translation vector *T*.
- One bearing measurement in R^3 (=unit vector in R^3) gives **two** scalar pieces of information.
- The information concerning *R* and *T* is **linear** in the unknown entries. **This is a helpful simplification**.

Future Work 2

- Deal with slowly changing INS coordinate frames
  - Requires some form of filter (particle filter, Kalman filter, etc).
  - Procedure described here would initialize the filter.
  - MLE with exponential forgetting of old measurements.
- Handling **partial** GPS data.

Thank You

Questions?