II.5 Sensor Selection & Tracking

Assoc. Prof. Tham Chen Khong
Dept of ECE, NUS

National University of Singapore

Overview

- What is CSIP?
- IDSQ
  - Bayesian estimation
  - Observation model
  - Information utility
  - Sensor selection
- Results
- Discussion of issues

References

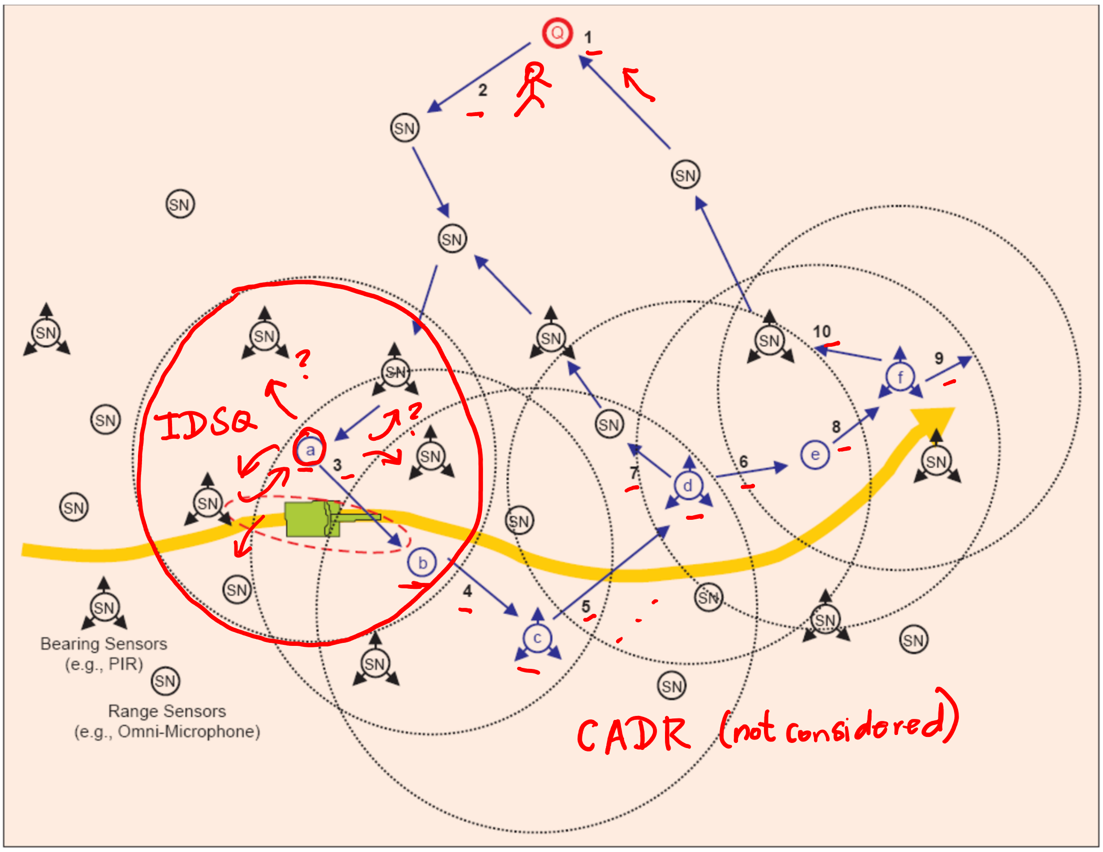
- [Zhao03] F. Zhao, J. Liu, J. Liu, L. Guibas and J. Reich, “Collaborative Signal and Information Processing: An Information-Directed Approach”, Proceedings of the IEEE, Vol. 91, no. 8, Aug 2003.

- [Chu01] M. Chu, H. Haussecker, and F. Zhao, “Scalable Information-Driven Sensor Querying and Routing for ad hoc Heterogeneous Sensor Networks”, Xerox Palo Alto Research Center Technical Report P2001-10113, May 2001.

CSIP

- Sense and perform signal processing locally and in collaboration with other sensor nodes
- limited power: so, try to minimize amount and range of communication as well

- we will consider the issue of **selecting sensors to participate in estimation of the state of interest**, e.g. location of a moving target in a tracking application



1. *A tracking scenario illustrating how the decision of sensor collaboration is accomplished using a measure of information utility as well as a measure of cost. Here, a vehicle moves through the sensor field from left to right. A user query is initially routed from node Q to node a which performs an initial estimate of the vehicle position. Node a then selects the next sensor, b, which it believes will provide the best measurement for the next estimation at a reasonable cost, and hands the current estimate to b. This process of sensor-to-sensor hand-off continues as the vehicle moves through the field. Periodically, the state estimation is sent back to the user using a shortest path routing algorithm such as directed diffusion routing.*

Information-Driven approach

- Use information-driven criterion to guide sensor selection
- Static and Dynamic versions
- Static case [IDSQ:Information Driven Sensor Query]
  - Querying node selects optimal sensors to request data from using information utility measures
  - Querying node tries to determine which node can provide the most useful information while balancing the energy cost, without the need to have the sensor data first !
- Dynamic case [CADR (not covered)]
  - Current sensor node updates the belief with its measurement and sends its estimation to the next neighbor that it determines can best improve the estimation

Constrained Anisotropic Diffusion Routing

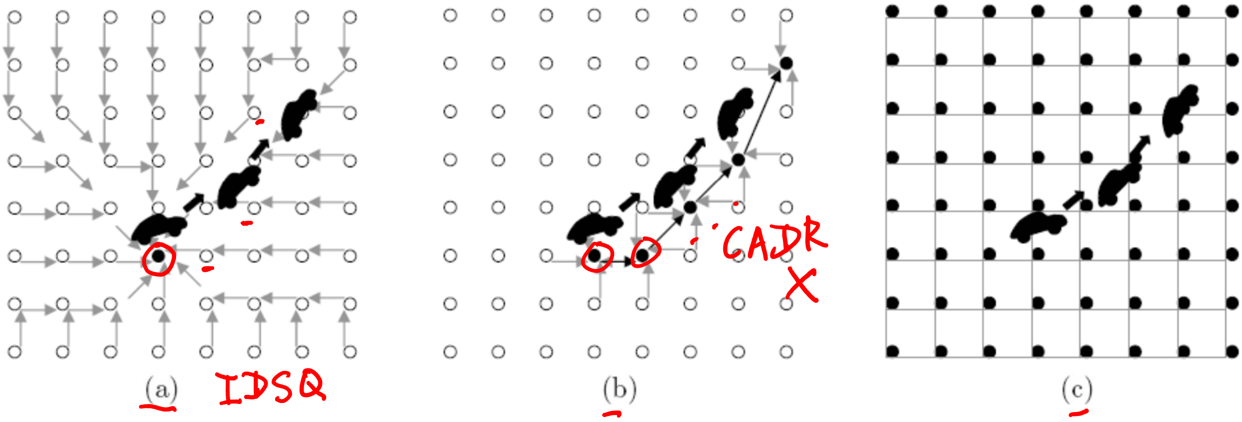


Fig. 2. Storage and communication of target state information in a networked distributed tracker. Circles on the grid represent sensor nodes, and some of the nodes, denoted by solid circles, store target state information. Thin, faded arrows or lines denote communication paths among the neighbor nodes. Thin, dark arrows denote sensor hand-offs. A target moves through the sensor field, indicated by thick arrows. (a) A fixed single leader node has the target state. (b) A succession of leader nodes are selected according to information such as vehicle movement. (c) Every node in the network stores and updates target state information.

Figure 10: AAVP7A1 tracked, armored assault vehicle.

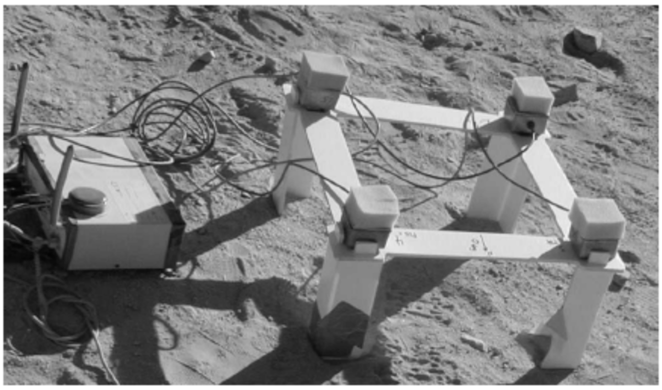


Figure 11: WINS 2.0 node with DOA-sensing microphone array.

IDSQ paper

- Considers how to: (1) query sensors, and (2) route data, in a network so that information gain is maximized while power and bandwidth consumption is minimized

IDSQ

CADR

- Two algorithms:
  - * – Information-Driven Sensor Querying (IDSQ)
  - – *Constrained Anisotropic Diffusion Routing (CADR)* (not covered)

Bayesian Estimation

Bayes Thm $P(B|A) = \frac{P(\overset{z}{A}|\overset{x}{B}) \cdot P(\overset{x}{B})}{P(A)}$ ($B = x$, $A = z$)

The basic task of tracking a moving target in a sensor field is to determine and report the underlying target state $\mathbf{x}^{(\mathbf{t})}$, such as its position and velocity, based on the sensor measurements up to time t, denoted as $\overline{\mathbf{z}^{(t)}} = \{\mathbf{z}^{(0)}, \mathbf{z}^{(1)}, \cdots, \mathbf{z}^{(t)}\}$. Many approaches have been developed over the last half century, including Kalman filters, which assume a Gaussian observation model and linear state dynamics, and, more generally, sequential Bayesian filtering, which computes the posterior belief at time $t+1$ based on the new measurement $\mathbf{z}^{(t+1)}$ and the belief $p(\mathbf{x}^{(t)}|\overline{\mathbf{z}^{(t)}})$ inherited from time t:

Posterior Belief Distribution — likelihood — prior $P(x^{(t+1)} | \bar{z}^{(t)})$

$$p(\mathbf{x}^{(t+1)}|\overline{\mathbf{z}^{(t+1)}}) \propto p(\mathbf{z}^{(t+1)}|\mathbf{x}^{(t+1)}) \cdot \int p(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}) \cdot p(\mathbf{x}^{(t)}|\overline{\mathbf{z}^{(t)}})\, d\mathbf{x}^{(t)}.$$

z?

36 36.7 37

x=36.7

Here $p(\mathbf{z}^{(t+1)}|\mathbf{x}^{(t+1)})$ denotes the observation model, and $p(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})$ the state dynamics model. As more data is gathered over time, the belief $p(\mathbf{x}^{(t)}|\overline{\mathbf{z}^{(t)}})$ is successively refined.

Definitions

state

- **Belief**: representation of the current *a posteriori* distribution of **x** given measurements **z**$_1$, …, **z**$_N$: $p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)$
- **Expectation** is considered the estimate
  $\bar{\mathbf{x}} = \int \mathbf{x}\, p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)\, d\mathbf{x}$ continuous prob distribution
- **Covariance** approximates residual uncertainty
  $\Sigma = \int (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)\, d\mathbf{x}$

Measurement/Observation model

$p(z|x)$

- **Measurement / Observation model:**
- $\mathbf{z}_i(t) = \mathbf{h}(\mathbf{x}(t), \lambda_i(t))$, (1)
  parameterized by $\lambda_i(t)$ and $\mathbf{z}_i(t)$, the characteristics and measurement of sensor *i* respectively. e.g. gain
- for sensors measuring sound amplitude

x x_i

- ○ $\lambda_i = [\, x_i , \sigma_i^{\,2}]^T$ acoustic sensor/mic (2)
- ○ x_i is the known sensor position and $\sigma_i^{\,2}$ is the known additive noise variance
- ○ $z_i = a \,/\, \|\, \mathbf{x}_i - \mathbf{x} \,\|^{\alpha/2} + w_i$, (3)
- ○ *a* is target amplitude, α is attenuation coefficient, w_i is Gaussian noise with variance $\sigma_i^{\,2}$

usefulness

Information Utility

The Information Utility function is defined as

psi $\psi : P(R^d) \rightarrow R$ real number

① $\psi_1 = 0.1$
② $\psi_2 = 0.6$ ✓ bigger is better

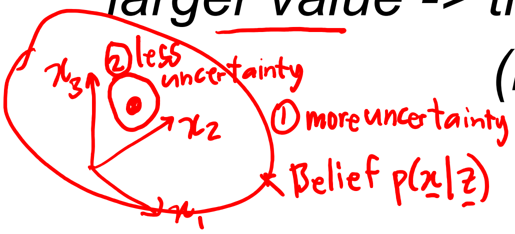
d is the dimension of x, P is all probability distributions in R^d

ψ assigns value to each element of $P(R^d)$ indicating the uncertainty of the distribution

smaller value ψ_1 *-> more spread out distribution*

larger value ψ_2 *-> tighter distribution*

(higher certainty, better)



Sensor Selection

- In theory:
- $j_0 = \arg_{j\in A} \max \psi(p(\mathbf{x}|\{\mathbf{z}_i\}_{i\in U} \cup \{\mathbf{z}_j\}))$ Belief dist^n U : set of sensor readings that have been incorporated
  - $A = \{1, \ldots, N\} - U$ is set of sensors whose measurements are not yet incorporated into belief
  - ψ is information utility function defined on the class of all probability distributions of $\mathbf{x}$
  - intuitively, select sensor j for querying such that information utility function of the distribution updated by $\mathbf{z}_j$ is maximized
- In practice:
  - $\mathbf{z}_j$ is unknown before it is sent back (we only know $\mathbf{h}$ and λ_i values)
  - e.g. best average case

$j' = \arg_{j\in A} \max E_{\mathbf{z}_j}[\psi(p(\mathbf{x}|\{\mathbf{z}_i\}_{i\in U} \cup \{\mathbf{z}_j\}))| \{\mathbf{z}_i\}_{i\in U}]$

Information Utility Measures

- When belief state is well-approximated by Gaussian distribution (with mean and covariance)

- *Mahalonobis distance*
  - effectively a weighted Euclidean distance where the weighting is determined by the covariance matrix
  - usefulness of sensor data is measured by how close the sensor is to the mean of the belief state under the Mahalanobis metric
  - a sensor geometry based measure:

information utility ψ is function of sensor location x_j only
(e.g. amplitude measuring sensors) (j is the candidate sensor node)

Mahalonobis Distance

$\psi(p_X) = -\boxed{(\mathbf{x}_j-\mathbf{x}_0)^T\Sigma^{-1}(\mathbf{x}_j-\mathbf{x}_0)}$ where $\mathbf{x}_0$ is the mean of the current estimate of target location

ref. Chu01 paper for details

Assume that estimation uncertainty can be approximated by Gaussian distribution

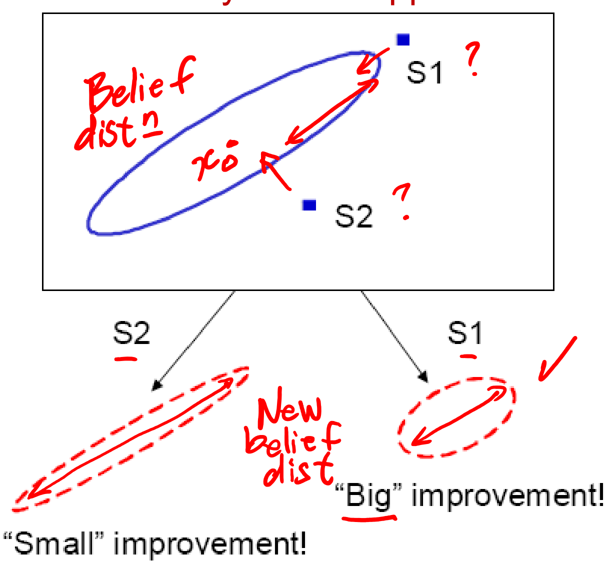


Fig. 3. Sensor selection based on information gain of individual sensor contributions. The information gain is measured by the reduction in the error ellipsoid. In the figure, reduction along the longest axis of the error ellipsoid produces a larger improvement in reducing uncertainty. Sensor placement geometry and sensing modality can be used to compare the possible information gain from each possible sensor selection, S1 or S2.

S1 has smaller Mahalanobis distance to $\boldsymbol{x}_0$ than S2

larger ψ

IDSQ Algorithm

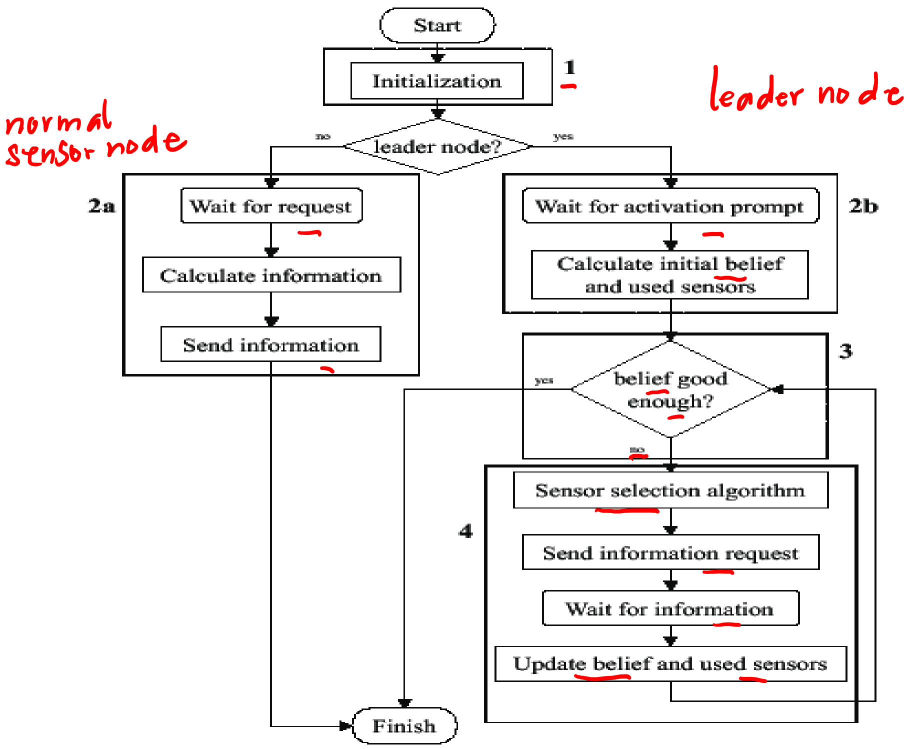


Figure 2: Flowchart of the information-driven sensor querying algorithm for each sensor.

IDSQ Algorithm

- Initialization
  - pick a leader l
  - leader has knowledge of certain characteristics $\{\lambda_i\}_{i=1}^{N}$ of the sensors
- Initial sensor reading
  - when a target is present in the range of the sensor cluster, the cluster leader will become activated

n. Assume we have a cluster of N sensors each labelled by a unique integer in $\{1, \ldots, N\}$. *A priori*, each sensor i only has knowledge of its own position $\mathbf{x}_i \in \mathcal{R}^2$. Figure 2 shows the flowchart of this algorithm which is identical for every sensor in the cluster. The algorithm works as follows:

1 Initialization Assuming all sensors are synchronized so that they are running the initialization routine at the same time, the first computation is to pick a leader from the cluster of N sensors. Depending on how the leader is determined, the sensors will have to communicate information about their position. For example, [Gao et al. 2001] describes a clustering algorithm with mobile centers. Leaving out the details of this leader selection, let us assume the sensor node labelled l is the leader. Assume also that the leader node has knowledge of certain characteristics $\{\lambda_i\}_{i=1}^N$ of the sensors in the network such as the positions of the sensor nodes.

2a Follower Nodes If the sensor node is not the leader, then the algorithm follows the left branch in Figure 2. These nodes will wait for the leader node to query them, and if they are queried, they will process their measurements and transmit the queried information back to the leader.

2b Initial Sensor Reading If the sensor node is the leader, then the algorithm follows the right branch in Figure 2. When a target is present in the range of the sensor cluster, the cluster leader will become activated (e.g. the amplitude reading at the leader node is greater than some threshold). The leader node will then

1. calculate a representation of the belief state with its own measurement, $p(\mathbf{x} \mid \mathbf{z}_l)$, and
2. begin to keep track of which sensors' measurements have been incorporated into the belief state, $U = \{l\} \subset \{1, \ldots, N\}$.

Again, it is assumed that the leader node has knowledge of the characteristics $\{\lambda_i\}_{i=1}^N$ of all the sensors within the cluster.

3 Belief Quality Test If the belief is good enough, based on some measure of goodness, the leader node is finished processing. Otherwise, it will continue with sensor selection.

4 Sensor Selection Based on the belief state, $p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U})$, and sensor characteristics, $\{\lambda_i\}_{i=1}^{N}$, pick a sensor node from $\{1, \ldots, N\} - U$ which satisfies some information criterion $\psi(\cdot)$. Say that node is j. Then, the leader will send a request for sensor j's measurement, and when the leader receives the requested information, it will

1. update the belief state with $\mathbf{z}_j$ to get a representation of
$$p(\mathbf{x} \mid \{\mathbf{z}_i\}_{i \in U} \cup \mathbf{z}_j) \text{ , and}$$
2. add j to the set of sensors whose measurements have already been incorporated
$$U := U \cup \{j\} \text{ .}$$

Now, go back to step 3 until the belief state is good enough.

Note: we have not discussed specific belief and covariance update methods

At the end of this algorithm, the leader node contains all the information about the belief from the sensor nodes by intelligently querying a subset of the nodes which provide the majority of the information. This reduces unnecessary power consumption by transmitting only the most useful information to the leader node. This computation can be thought of as a local computation for this cluster. The belief stored by the leader can then be passed up for processing at higher levels.

Incremental Update of Belief

New Belief

- $p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)$

likelihood/obs mdl prior

$= c\, p(\mathbf{z}_N \mid \mathbf{x})\, p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_{N-1})$
  - $\mathbf{z}_N$ *is the new measurement*
  - $p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_{N-1})$ is previous belief
  - $p(\mathbf{x} \mid \mathbf{z}_1, \ldots, \mathbf{z}_N)$ is updated belief
  - c is a normalizing constant
- for linear system with Gaussian distribution, *Kalman filter* is used
- mean & Covariance ~~is also~~ are updated

IDSQ Experiments
- Sensor Selection Criteria

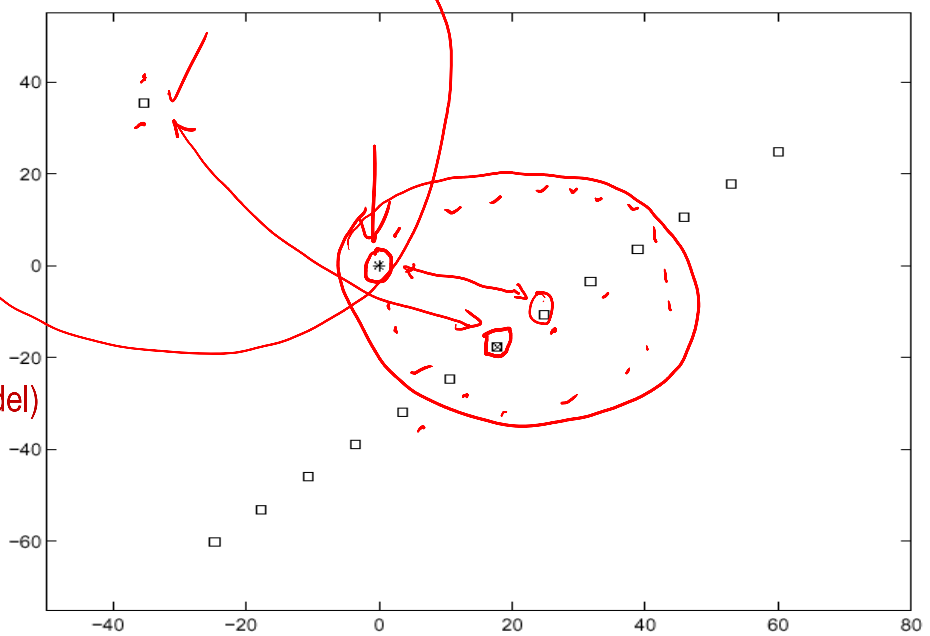
- A. *Nearest Neighbor data diffusion*

  $j_0 = \arg_{j \in \{1, \ldots, N\}-U} \min \|\mathbf{x}_l - \mathbf{x}_j\|$

- B. *Mahalanobis distance*

  $j_0 = \arg_{j \in \{1, \ldots, N\}-U} \min (\mathbf{x}_j - \mathbf{x}_0)\Sigma^{-1}(\mathbf{x}_j - \mathbf{x}_0)$

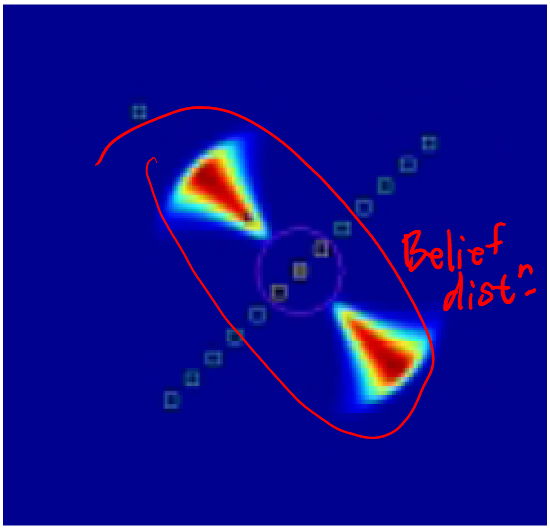
IDSQ Experiments



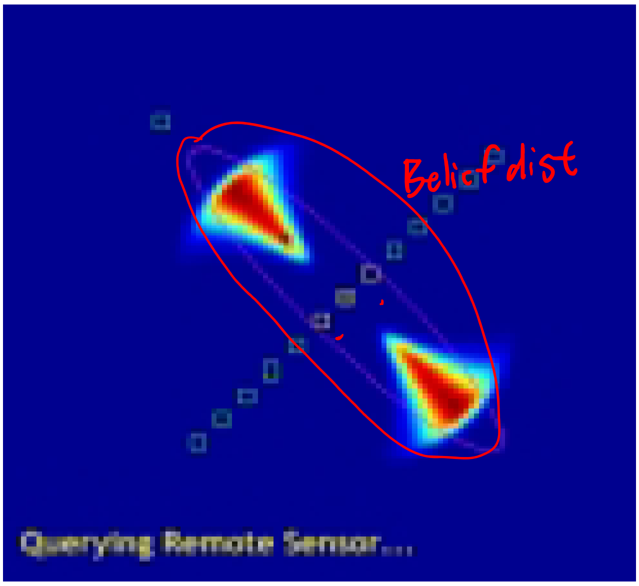
14 acoustic sensors (mics)(see earlier model)

Figure 3: Layout of all-but-one-colinear sensors (squares) and target (asterisk). The leader node is denoted by an $\times$.

IDSQ - Comparison of NN and Mahalanobis distance method

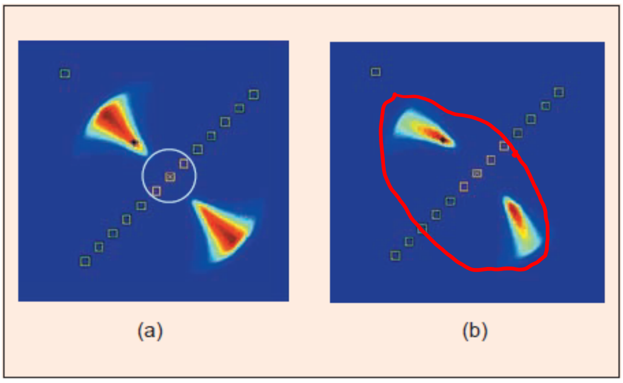


NN distance method



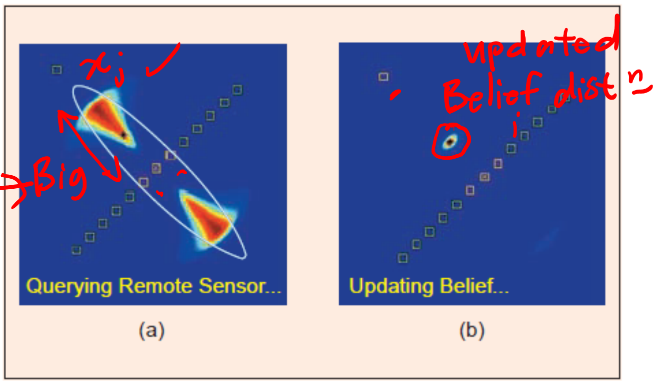
Mahalanobis distance method

Posterior distributions (bimodal)



5. *Sensor selection based on the nearest neighbor method. The estimation task here is to localize a stationary target labeled "*." Squares denote sensors. (a) select the nearest sensor and (b) incorporate the new measurement from the selected sensor.*

Σ^{-1}



6. *Sensor selection based on the Mahalanobis distance measure of information utility. The localization problem is the same as that in Fig. 5.*

IDSQ Experiments

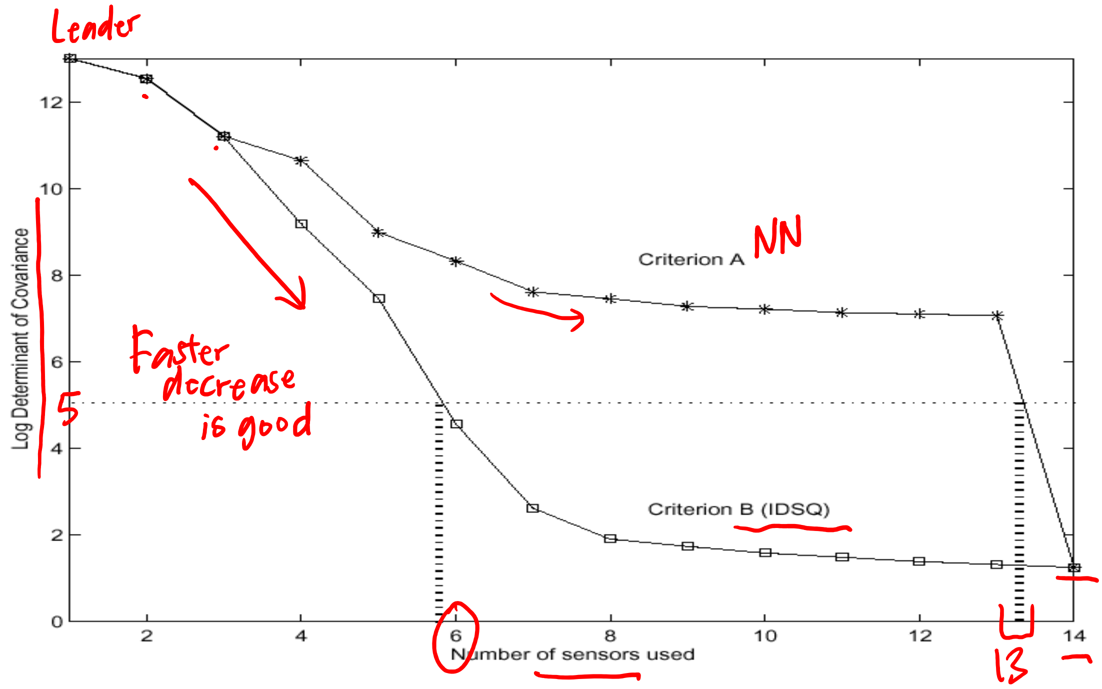


Figure 6: Determinant of the error covariance for selection criteria A NN and B (IDSQ) for the all-but-one-colinear sensor layout. A tasks 14 sensors while B tasks 6 sensors to be below an error threshold of 5 units.

IDSQ Experiments

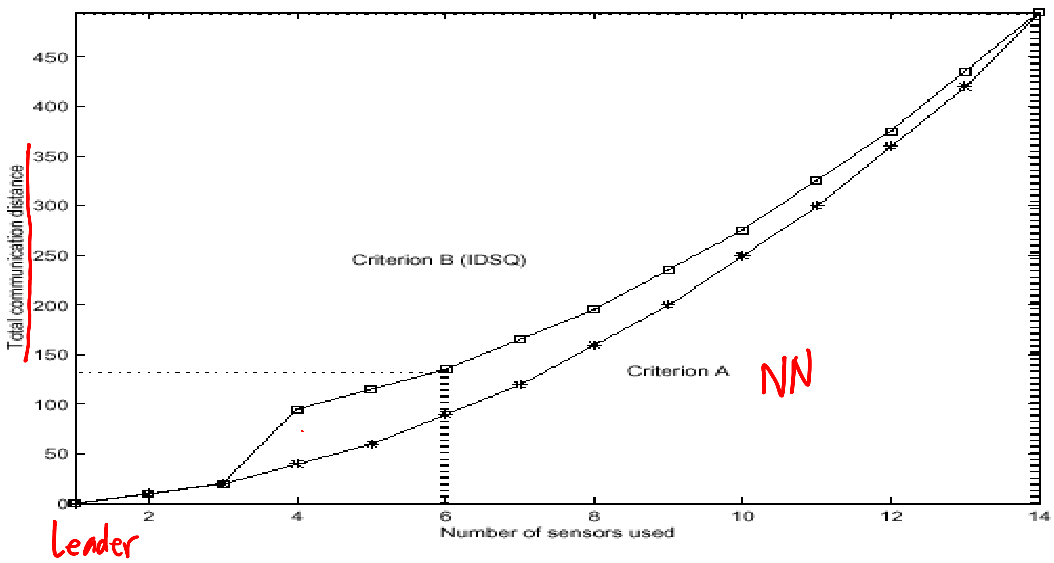


Figure 7: Total communication distance NN vs. the number of sensors queried for selection criteria A and B (IDSQ) for the all-but-one-colinear sensor layout. For achieving the same threshold of the error, A tasks 14 sensors and uses nearly 500 units of communication distance whereas B tasks 6 sensors and uses less than 150 units of communication distance.

IDSQ Experiments

II.

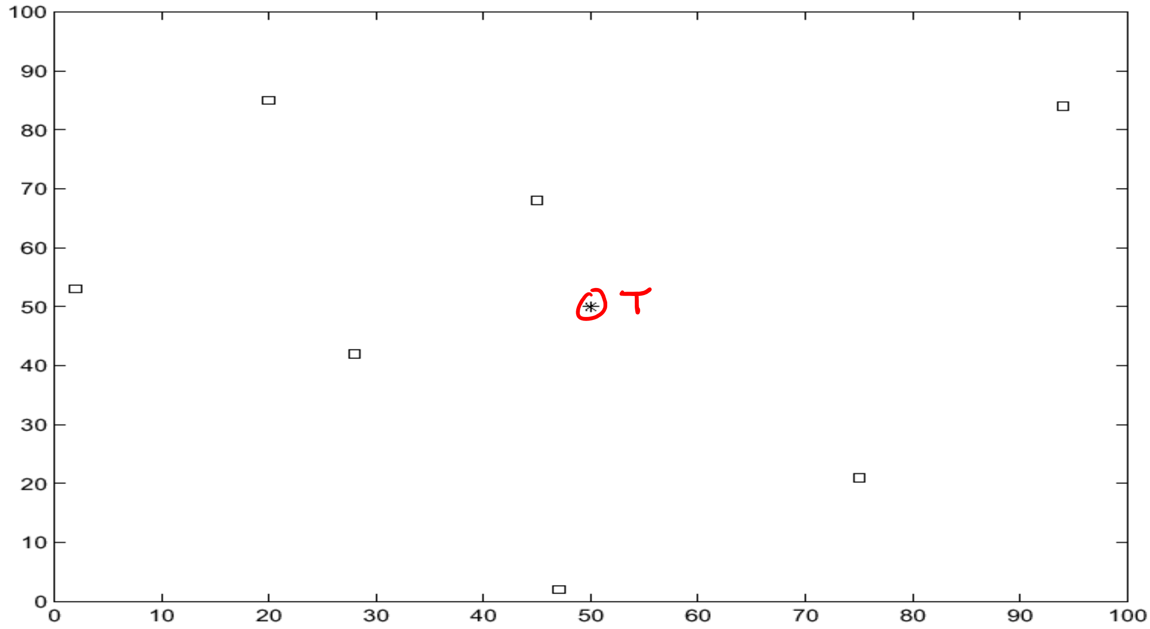


Figure 8: Layout of seven randomly placed sensors (squares) with target in the middle (asterisk).

IDSQ Experiments

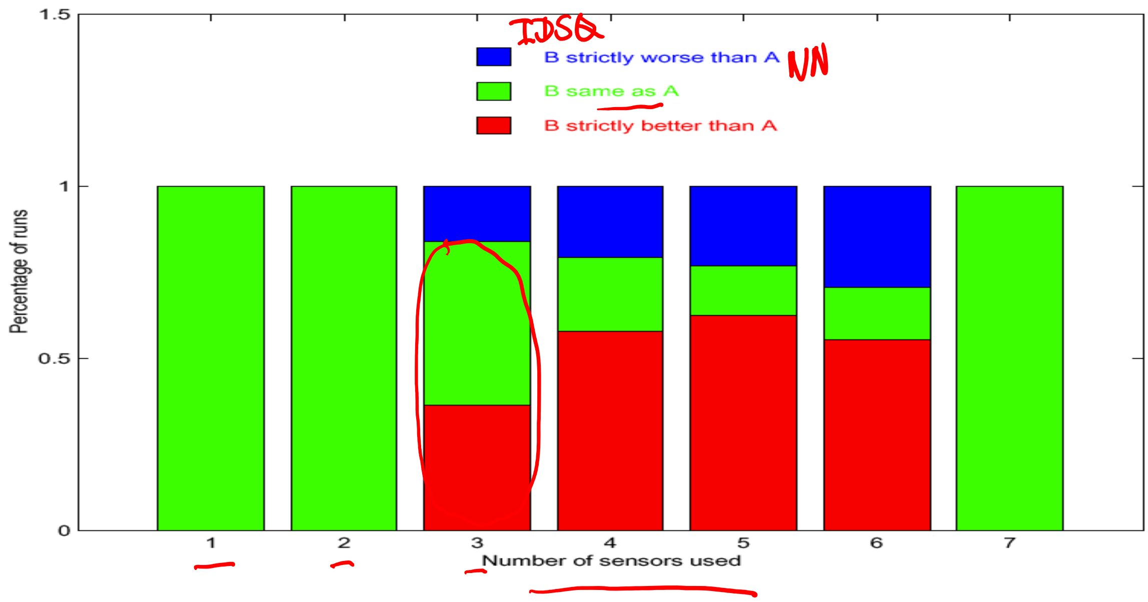


Figure 9: Percentage of runs where **B** performs better than **A** for seven randomly placed sensors.

IDSQ

B uses the Mahalanobis distance, A is Nearest Neighbor Diffusion. "Performs Better" means smaller error for given number of sensors used.

Discussion

- Basic IDSQ ignores communication cost
- Can enhance with composite objective function: Bigger is better
- $M_c(\lambda_l, \lambda_j, p(\mathbf{x}|\{\mathbf{z}_i\}_{i\in U}) =$
  $\gamma M_u(p(\mathbf{x}|\{\mathbf{z}_i\}_{i\in U}, \lambda_j) - (1 - \gamma)M_a(\lambda_l, \lambda_j)$
  - l is leader node
  - M_u is information utility measure ψ
  - M_a is communication cost measure
  - $\gamma \in [0, 1]$ balances their contributions 0.5
  - λ_j is characteristics of current sensor j
- $j_0 = \arg_{j\in A} \max M_c(\lambda_l, \lambda_j, p(\mathbf{x}|\{\mathbf{z}_i\}_{i\in U})$ $\equiv x_0$: mean of current Belief distⁿ

  e.g.
  $$M_u(\mathbf{x}_j, \hat{\mathbf{x}}, \Sigma) = -[(\mathbf{x}_j - \hat{\mathbf{x}})^T \Sigma^{-1} (\mathbf{x}_j - \hat{\mathbf{x}})] \text{ Mahalanobis Dist}$$
  $$M_a(\mathbf{x}_j, \mathbf{x}_l) = (\mathbf{x}_j - \mathbf{x}_l)^T (\mathbf{x}_j - \mathbf{x}_l)$$

Questions?