In the name of GOD

DISTRIBUTED CONSENSUS-BASED TRACKING IN WIRELESS SENSOR NETWORKS: A PRACTICAL APPROACH

Electrical Engineering Department of
Shahid Beheshti University

Course name: Mobile Communication

Advisor: Dr.Ghorashi
Lecturer: A.Hajihoseini

April 2015

ABSTRACT

Tracking is a key application in Wireless Sensor and Ad-hoc Networks. The development of distributed, energy-efficient strategies for target tracking is of practical importance.

In this contribution we propose a distributed tracking strategy based on cooperative localization using consensus.

Nodes do not need to care about the network topology or the number of nodes present in the network.

They use their own measurements to track target.

The proposed approach appears to be robust against small errors in the node positions .

Introduction

The main features of Wireless Sensor Networks (WSN)

- low-cost nodes
- limited computational and power resources.

WSNs must also be robust against changes in topology and energy efficient.

These limitations also make centralized approaches not very suitable for being used in WSNs.

Introduction

Methods for acquiring the position of a target node :

Time-of-Arrival (TOA)

Time-difference of Arrival (TDOA)

Angle of Arrival (AOA)

Received Signal Strength Indicator (RSSI)

In this paper we focus on the use of RSSI measurements for the localization task.

One of the main challenges when using RSSI measurements is that the mapping between the measurement and target's position is nonlinear and hence, classical tracking strategies like the Kalman filter are not suitable.
So some particle filtering approaches proposed.
In general, particle filtering approaches have shown very good performance when dealing with RSSI measurements but they are centralized and suffer from a high computational cost and hence, their applicability in a real scenario is questionable.

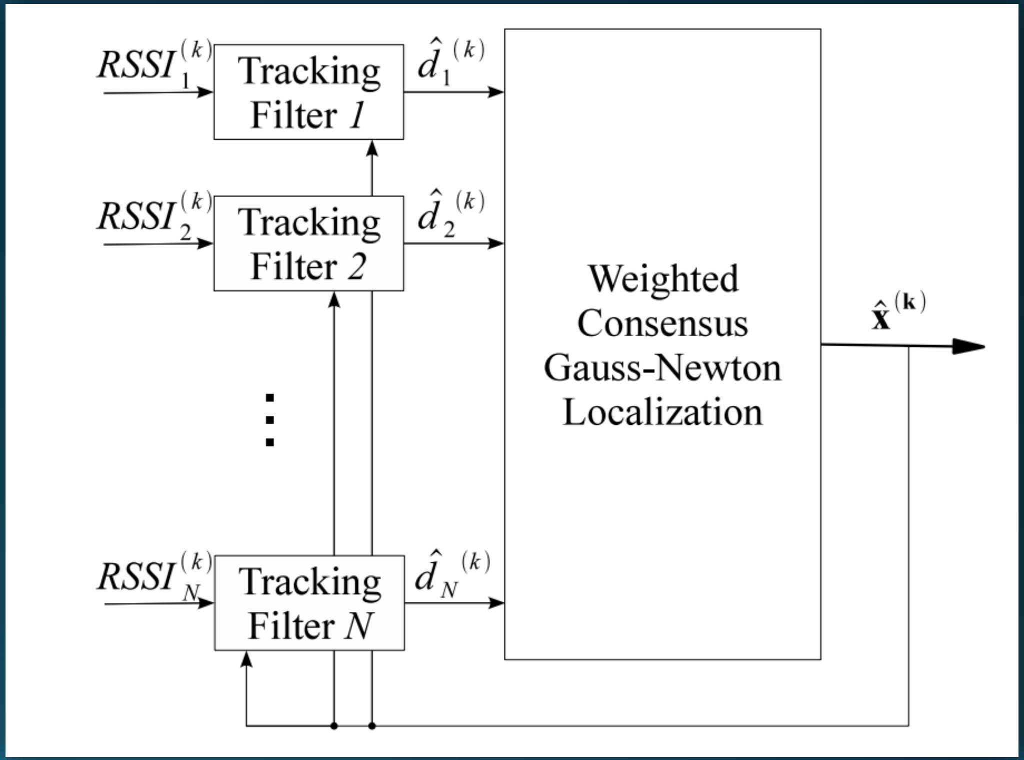
Introduction

In this work we propose a two-step tracking strategy.

Nodes perform local target tracking using the Unscented Kalman Filter and then combine the smoothed data (coming from the tracking fileters) in a distributed fashion using consensus-based localization.

The proposed algorithm has the following desirable properties:

- it is scalable (i.e. nodes do not care about the network topology)
- computationally simple (nodes only need to run a local UKF)
- energy efficient



Further, the use of weights in the joint estimation process make the proposed approach robust against uncertainties in nodes' positions and/or bias in the measurements.

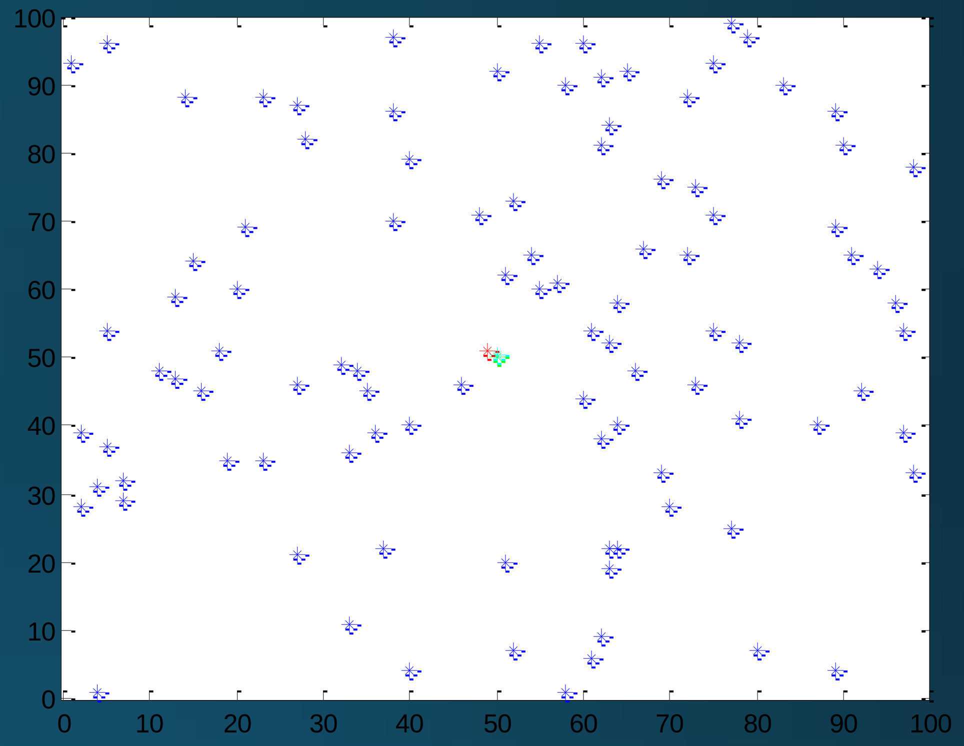
PROBLEM FORMULATION AND DEFINITIONS

Consider a Wireless Sensor Network of N nodes randomly deployed on a certain area.

Nodes are static and able to communicate with adjacent nodes that lie within a given range for communications.

Assume the presence of a target node that moves within the network.

The goal is to determine the location of the target node and be capable of tracking its position as time evolves.



Network

For getting estimates of the target position, nodes employ RSSI measurements.

A common assumption and references therein, is that the received power follows a lognormal distribution with a distance-dependent mean as

$$P_R[\mathrm{dB}] = P_0 - 10 n_p \log_{10}\left(\frac{d}{d_0}\right) + X,$$

$$\hat{d}_n = d_0\, 10^{\left(\frac{P_0 - P_{R,n}}{10 n_p}\right)}.$$

Target

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \mathbf{v}^{(k)} T + \frac{1}{2}\mathbf{a}^{(k+1)} T^2 \\ \mathbf{v}^{(k+1)} &= \mathbf{v}^{(k)} + \mathbf{a}^{(k+1)} T \ , \end{aligned}$$

DISTRIBUTED LOCALIZATION

$$
\begin{aligned}
d_1^2 &= (x_1 - x_t)^2 + (y_1 - y_t)^2 \\
d_2^2 &= (x_2 - x_t)^2 + (y_2 - y_t)^2 \\
&\vdots \\
d_N^2 &= (x_N - x_t)^2 + (y_N - y_t)^2
\end{aligned}
$$

$$
\begin{bmatrix} d_1^2 - (x_1^2 + y_1^2) \\ \vdots \\ d_N^2 - (x_N^2 + y_N^2) \end{bmatrix} = \left(\mathbf{x}^\mathsf{T}\mathbf{x}\right) \cdot \mathbf{1} - 2 \underbrace{\begin{bmatrix} x_1 & y_1 \\ \vdots & \\ x_N & y_N \end{bmatrix}}_{\mathbf{C}=[\mathbf{c}_1, \ldots, \mathbf{c}_N]^\mathsf{T}} \mathbf{x}
$$

$$
\mathbf{b} = \left[\hat{d}_1^2 - (x_1^2 + y_1^2), \ldots, \hat{d}_N^2 - (x_N^2 + y_N^2)\right]^\mathsf{T}
$$

$$
\tilde{\mathbf{f}}(\mathbf{x}) = \left(\mathbf{x}^\mathsf{T}\mathbf{x}\right) \cdot \mathbf{1} - 2\mathbf{C}\,\mathbf{x} - \mathbf{b}
$$

$\rightarrow$ Cost function

DISTRIBUTED LOCALIZATION

In order to incorporate robustness and make the localization task more applicable to realistic scenarios we propose to use a weighted version of the cost function.

In a WSN it may happen that some of the nodes exhibit irregular behavior (i.e. bias in their measurements).

Additionally, nodes may not have precise information about their own locations instead, some inaccuracies may be present.

The incorporation of weights will mitigate the effects of misbehaving or biased nodes and uncertainties in nodes' positions.

$$\mathbf{f}(\mathbf{x}) = \boldsymbol{\Gamma}\tilde{\mathbf{f}}(\mathbf{x})$$

$$\boldsymbol{\Gamma} = \mathrm{diag}\left(1/\hat{d}_1, \ldots, 1/\hat{d}_N\right)$$

DISTRIBUTED LOCALIZATION

Algorithm 1 Distributed Gauss-Newton localization

$\hat{\mathbf{x}}^{(0)} \leftarrow$ same initial value $\forall\, n \in N$

for $k = 0$ to $K-1$ **do**

$\quad \mathbf{J}_n^{(k)} \leftarrow \frac{2}{\hat{d}_n} \left[\hat{x}_t^{(k)} - x_n \quad \hat{y}_t^{(k)} - y_n \right]$

$\quad f_n(\hat{\mathbf{x}}^{(k)}) \leftarrow \frac{\hat{\mathbf{x}}^{(k)\mathsf{T}}\hat{\mathbf{x}}^{(k)} - 2\,\mathbf{c}_n^{\mathsf{T}}\hat{\mathbf{x}}^{(k)} + x_n^2 + y_n^2 - \hat{d}_n^2}{\hat{d}_n}$

$\quad \boldsymbol{\Delta}_n^{(k)} \leftarrow \mathbf{J}_n^{(k)\mathsf{T}}\mathbf{J}_n^{(k)}$

$\quad \boldsymbol{\gamma}_n^{(k)} \leftarrow \mathbf{J}_n^{(k)\mathsf{T}} f_n(\hat{\mathbf{x}}^{(k)})$

$\quad$ **consensus**

$\quad\quad \boldsymbol{\Delta}_*^{(k)} \leftarrow \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\Delta}_n^{(k)} = \frac{1}{N}\mathbf{J}^{(k)\mathsf{T}}\mathbf{J}^{(k)}$

$\quad\quad \boldsymbol{\gamma}_*^{(k)} \leftarrow \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\gamma}_n^{(k)} = \frac{1}{N}\mathbf{J}^{(k)\mathsf{T}} f(\hat{\mathbf{x}}^{(k)})$

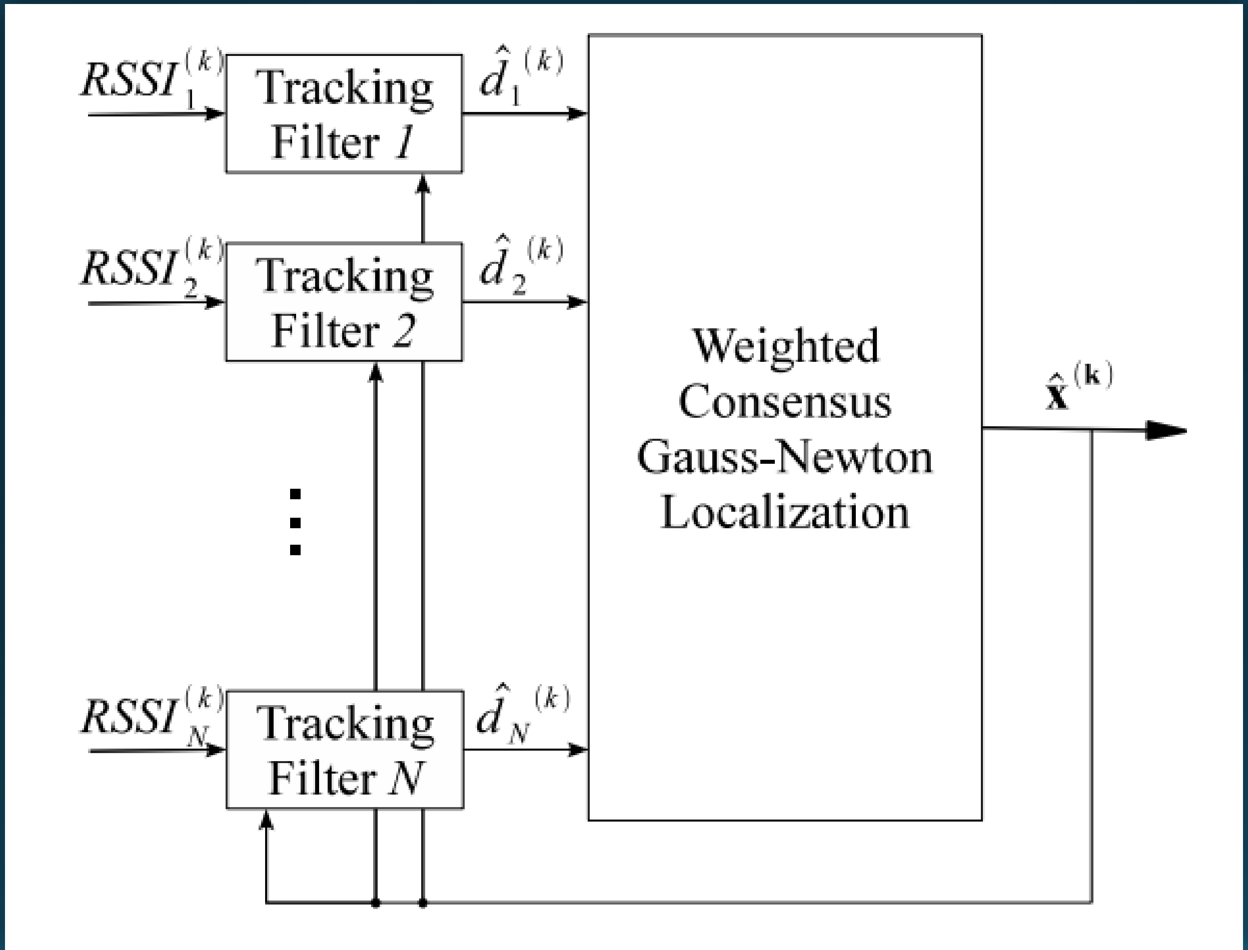
$\quad$ **end consensus**

$\quad \mathbf{h}^{(k)} \leftarrow \boldsymbol{\Delta}_*^{(k)^{-1}}\boldsymbol{\gamma}_*^{(k)} = \left(\mathbf{J}^{(k)\mathsf{T}}\mathbf{J}^{(k)}\right)^{-1}\mathbf{J}^{(k)\mathsf{T}}\mathbf{f}(\hat{\mathbf{x}}^{(k)})$

$\quad \hat{\mathbf{x}}^{(k+1)} \leftarrow \hat{\mathbf{x}}^{(k)} + \mathbf{h}^{(k)}$

end for

Distributed Tracking



Simulations

A network of 100 nodes scattered over a 100×100 [m2] area

We have also generated 100 different target trajectories over the network area

For comparison purposes we also consider in our simulations a centralized version of the UKF where a central entity is assumed to collect all data coming from the nodes.

A perfect consensus is assumed among the nodes.