Indoor positioning using Particle Filter

Lior Kissos

Contents

1 Introduction

This report summarizes the short investigation of the indoor positioning problem faced by ? and suggests a possible solution; the Particle Filter . The chosen technology is the Ultra Wideband (UWB) and the device is the almost industry standard Decawave's DW1000.

2 Trilateration [1]

The basic information provided by the DW1000 is the Time Of Flight (TOF) from a given tag to an anchor. This time of flight is assumed to reflect the distance according to:

$$r^n[m] = c[\frac{m}{sec}] \cdot TOF^n[sec] \tag{2.1}$$

Where; $c = 3 \cdot 10^8[m]$, is the speed of light, and n is the anchor's index (not a power)

The basic positioning derivation technique is the Trilateration; TOF between a tag and 3 or more anchors allows to intersect 3 imaginary spheres having each a radius r^n. In fact, since the heights of both tag and anchor are known and constant, we may conduct all the calculation in a 2-dimensional plane (Pythagorean triplet);

$$(d^n)^2 = (r^n)^2 - (h^n - h^{tag})^2 \tag{2.2}$$

Where;

$$(d_k^n)^2 = \left(x_k^n - x_k^{tag}\right)^2 + \left(y_k^n - y_k^{tag}\right)^2; n = 1,2,3. k = time\ step \tag{2.3}$$

An illustration of the intersection is brought in Figure 1;

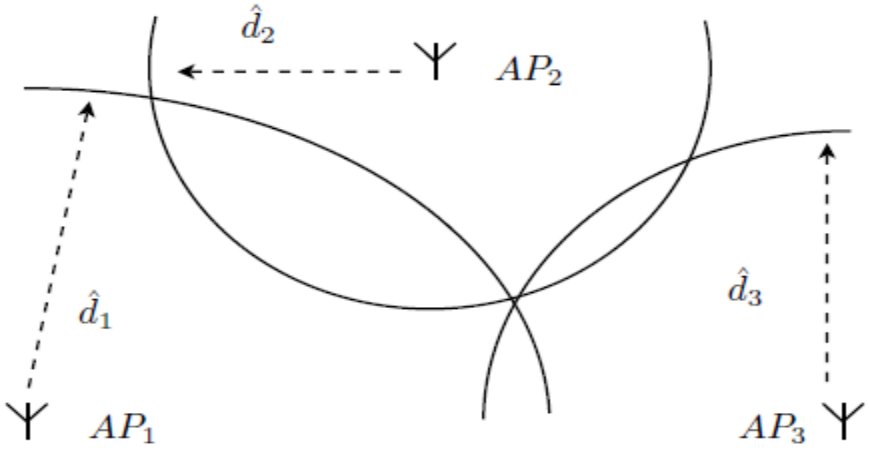

Figure 1: Trilateration

The Trilateration is a “memoryless” technique, as it uses only the current observation to derive the location. Another memoryless technique exists, Multilateration, but is much less used for the reasons mentioned in [2].

3 Particle Filter

3.1 Problem formulation

The Particle Filter has been widely used in motion tracking problems, whose observations are noisy. In our case, the estimated parameter is the current position and velocity of an object, the state vector, and the observations are the distances (or the square distances) to the anchors. In general, the state and the observation are vectors. We regard the state and its observation as a Hidden Markovian Random Process (HMM):

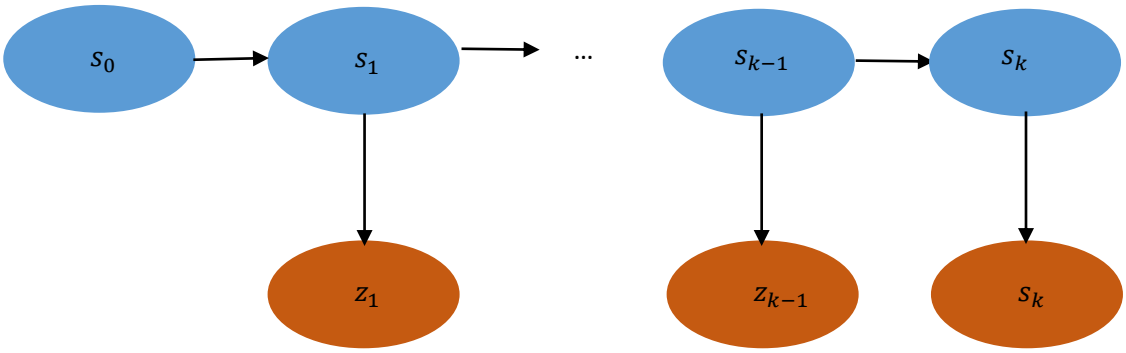

Figure 2: HMM process model

Hence, known algebraic and probabilistic relationships between current state and observation to the previous;

$s_k \triangleq$ Current state

$z_k \triangleq$ Current observation

Probabilistic relationship;

$$P(s_k|s_{1:k-1}) = P(s_k|s_{k-1})$$
$$P(z_k|s_{1:k}, s_{1:k-1}) = P(z_k|s_k) \quad (3.1)$$

The physical model assumed by the Particle Filter is expressed by the algebraic relationship:

The State equation $$s_k = f(s_{k-1}, w_{k-1}) \quad (3.2)$$

The Measurement equation $$z_k = h(s_k, v_k) \quad (3.3)$$

Where;

- w_{k-1} is the previous step’s added noise; the state equation noise.

- v_k is the current step's added noise; the measurement equation noise.

The functions f, h are not necessarily linear, and the noises are not necessarily Gaussian thus unfit to be treated by the Basic or the Extended Kalman filter.

3.2 The need: Estimate conditional probability- $p(s_k|z_{1:k})$

The particle filter aims to estimate the conditional probability $p(s_k|z_{1:k})$ and further use it to generate 2 possible types of estimators:

- MAP estimator: $\hat{s}_{k,MAP} = \underset{s_k \in \mathbb{R}^{nx1}}{argmax}\, p(s_k|z_{1:k})$
- MMSE estimator: $\hat{s}_{k,MMSE} = E(s_k|z_{1:k}) = \sum_{s_k} s_k \cdot p(s_k|z_{1:k})$

This will be done in a sequential manner

3.3 Monte Carlo approximation

If we wish to evaluate the expectation based on a known probability function;

$$E(g(x)) = \int g(x)p(x)dx \tag{3.4}$$

A Monte Carlo approximation of the expectation is based on the averaging over a histogram of randomly generated x samples;

1. Drawing N_s samples of x: $\{x^i\}_{i=1}^{N_s}$ from a known probability function $p(x)$
2. Calculation of: $\{g(x^i)\}_{i=1}^{N_s}$ - the histogram
3. Monte Carlo estimation of the expectation:

$$E\big(g(x)\big) \cong \frac{1}{N_s}\sum_{i=1}^{N_s} g(x^i) \tag{3.5}$$

The underlying assumption is that the occurrence rate of the x^i reflects well the probability function $p(x)$ and that's what makes 3.5 an approximation of 3.4;

$$\frac{1}{N_s}\sum_{i=1}^{N_s} g\big(x^i\big) \underset{N_s \to \infty}{\longrightarrow} E\big(g(x)\big)$$

3.4 Importance Sampling [3]

When $p(x)$ is difficult to draw samples from or is unknown, a bypass is suggested. We choose a probability density function $q(x)$ easy to draw samples from (e.g Normal), and combine it in 3.4 such that;

$$E(g(x)) = \int g(x)\frac{p(x)}{q(x)}q(x)dx$$

In that case the Monte Carlo estimation is;

$$E\big(g(x)\big) \cong \frac{1}{N_s}\sum_{i}^{N_s} g(x^i)\frac{p(x^i)}{q(x^i)}$$

This will allow us to draw Monte Carlo samples $\{x^i\}_{i=1}^{N_s}$ from $q(x^i)$ instead of $p(x^i)$, and compensate for that with the Importance weight, that will be defined below ([4],page 6, between eq.14 and eq.15);

We can define importance weights;

$$w^i \triangleq \frac{1}{N_s}\frac{p(x^i)}{q(x^i)}$$

In our case;

$$w_k^i = \frac{1}{N_s}\frac{p(s_k^i|z_{1:k})}{q(s_k^i|z_{1:k})} \qquad 3.6$$

Using Bayes identity;

$$w_k^i = \frac{1}{N_s}\frac{p(z_{1:k}|s_k^i)p(s_k^i)}{p(z_{1:k})}\frac{1}{q(s_k^i|z_{1:k})}$$

If we define the UnNormalized weights as :

$$\widetilde{w}_k^i \triangleq \frac{p(z_{1:k}|s_k^i)p(s_k^i)}{q(s_k^i|z_{1:k})} \qquad 3.7$$

And count on a further normalization to correct that ([4],page 6, between eq.14 and eq.15; just as with the introduction of $q(x)$)

We normalize by $\sum_i^{N_s}\widetilde{w}^i$, then we get;

$$E(g(s_k)|z_k) \cong \frac{1}{\sum_i^{N_s}\widetilde{w}^i}\sum_{i=1}^{N_s} g(s_k^i)\frac{p(z_{1:k}|s_k^i)p(s_k^i)}{q(s_k^i|z_{1:k})} = \frac{1}{\sum_i^{N_s}\widetilde{w}^i}\sum_{i=1}^{N_s}\widetilde{w}_k^i \cdot g(s_k^i) \qquad 3.8$$

3.5 The Particle principle: Grid-based approximation

We intend to approximate the continuous probability distribution mentioned in 3.2 based on observations ("Monte Carlo"). The grid based approximation suggests that a discrete N_s-long grid of $nx1$ vectors belonging to the continuous probability space be sampled (to become a grid) and that the required distribution be evaluated on that grid. Namely, s_k is the continuous n-dimensional vector for which we define;

The grid at step k;

$$\{s_k^i\}_{i=1}^{N_s}$$

Since on a "continuous grid";

$$E(g(s_k)|z_k) = \sum_{s_k} g(s_k)P(s_k|z_k)$$

The conditional probability on the discrete grid at step k following 3.8 is:

$$P(s_k = s_k^i|\, z_{1:k}) = P(s_k^i|\, z_{1:k}) = \frac{w_k^i}{\sum_i^{N_s} w^i} \qquad 3.9$$

And the probability of the continuous s_k is then approximated with the aid of 3.9;

$$P(s_k|\, z_{1:k}) \cong \sum_{i=1}^{N_s} P(s_k^i|\, z_{1:k})\delta(s_k - s_k^i) \propto \sum_{i=1}^{N_s} w_k^i \delta(s_k - s_k^i)$$

The N_s terms of the grid are called "Particles", and the conditional probabilities are called "Weights".

It can be shown that;

$$\sum_{i=1}^{N_s} P(s_k^i|\, z_{1:k})\delta(s_k - s_k^i) \underset{N_s\to\infty}{\longrightarrow} P(s_k|\, z_{1:k})$$

The Particle Filtering algorithm is a sequential procedure of;

1. Particle vector $\{s_k^i\}_{i=1}^{N_s}$ realization (based on a known probability distribution predefined by the model)
2. Observation z_k acquisition
3. Weights $\{w_k^i\}_{i=1}^{N_s}$ calculation

3.6 Recursion ([5], p.5)

Basic relationships:

1. $p(a|b) = \frac{p(b|a)p(a)}{p(b)}$
2. $p(a,b) = p(a|b)p(b)$
3. $p(a,b|c) = p(a|b,c)p(b|c)$

We thrive to obtain a sequential algorithm of the calculation of the importance weights, $\{w_k^i\}_{i=1}^{N_s}$. We depart from 3.6;

$$w_k^i \propto \frac{p(s_k^i|z_{1:k})}{q(s_k^i|z_{1:k})} \qquad 3.10$$

The numerator:

$$p(s_{0:k}|z_{1:k}) \overset{(1)}{=} \frac{p(z_{1:k}|s_{0:k})p(s_{0:k})}{p(z_{1:k})}$$
$$= \frac{p(z_k, z_{1:k-1}|s_{0:k})p(s_{0:k})}{p(z_k, z_{1:k-1})} \overset{(2)\&(3)}{=} \frac{p(z_k|z_{1:k-1}, s_{0:k})p(z_{1:k-1}|s_{0:k})p(s_{0:k})}{p(z_k|z_{1:k-1})p(z_{1:k-1})} \overset{HMM}{=} \frac{p(z_k|s_k)p(z_{1:k-1}|s_{0:k})p(s_{0:k})}{p(z_k|z_{1:k-1})p(z_{1:k-1})}$$
$$= \frac{p(z_k|s_k)p(s_{0:k})}{p(z_k|z_{1:k-1})p(z_{1:k-1})} \frac{p(\, s_{0:k}|z_{1:k-1})p(z_{1:k-1})}{p(s_{0:k})}$$
$$= \frac{p(z_k|s_k)p(\, s_{0:k}|z_{1:k-1})}{p(z_k|z_{1:k-1})} \overset{(3)}{=} \frac{p(z_k|s_k)}{p(z_k|z_{1:k-1})} p(\, s_k|z_{1:k-1}, s_{0:k-1})p(\, s_{0:k-1}|z_{1:k-1}) \overset{HMM}{=}$$
$$\frac{p(z_k|s_k)}{p(z_k|z_{1:k-1})} p(\, s_k|s_{k-1})p(\, s_{0:k-1}|z_{1:k-1})$$

The denominator:

$$q(s^i_{0:k}|z_{1:k}) \overset{(3)}{=} q(s^i_k|s^i_{0:k-1}, z_{1:k})q(s^i_{0:k-1}|z_{1:k}) \overset{causality}{=} q(s^i_k|s^i_{0:k-1}, z_{1:k})q(s^i_{0:k-1}|z_{1:k-1})$$

Hence,

$$w^i_k \propto \frac{p(z_k|s_k)p(\, s_k|s_{k-1})p(\, s_{0:k-1}|z_{1:k-1})}{q(s^i_k|s^i_{0:k-1}, z_{1:k})q(s^i_{0:k-1}|z_{1:k-1})} = \frac{p(z_k|s_k)p(\, s_k|s_{k-1})}{q(s^i_k|s^i_{0:k-1}, z_{1:k})} \frac{p(\, s_{0:k-1}|z_{1:k-1})}{q(s^i_{0:k-1}|z_{1:k-1})}$$

However (see 3.10);

$$w^i_{k-1} \propto \frac{p(\, s_{0:k-1}|z_{1:k-1})}{q(s^i_{0:k-1}|z_{1:k-1})}$$

Therefore, we obtain the recursion;

$$w^i_k \propto \frac{p(z_k|s_k)p(\, s_k|s_{k-1})}{q(s^i_k|s^i_{0:k-1}, z_{1:k})} w^i_{k-1} \qquad 3.11$$

3.7 Basic algorithm ([5], p.5)

Initialization:

$s^i_0 \sim p(s_0)$, the apriori distribution

$\{w^i_o\}_{i=1}^{N_s} = \frac{1}{N_s}$, a uniform distribution

1. **Prediction**: draw new grid/set of particles

$$x^i_k \sim q(x_k|x_{0:k-1}, z_{1:k}) \qquad 3.12$$

2. **Update**: calculate new weights;

$$w^i_k \propto \frac{p(z_k|s_k)p(\, s_k|s_{k-1})}{q(s^i_k|s^i_{0:k-1}, z_{1:k})} w^i_{k-1} \qquad 3.13$$

3.8 Resampling

The basic algorithm suffers from degeneration; the weights series $\{w_k^i\}_{i=1}^{N_s}$ converges towards a single non-zero value as k increases. The solution is "Resampling". We define -

$$N_{s,eff} \triangleq \frac{1}{\sum_i^{N_s}(w_k^i)^2}$$

An effective number of non zero-probability particles. Once a certain threshold is crossed;

- Extract a new grid of N_s terms whose weights are higher than a certain threshold. Of course, some of the new particles appear more than once in the new grid
- Assign the new grid/particles series a uniform weight; $\{w_k^i\}_{i=1}^{N_s} = \frac{1}{N_s}$

3.9 The choice of q

A thorough discussion is devoted to the issue in [4] and [5], as this has a serious impact on performance. It is recommended to investigate and dedicate a specific q function for the given problem.

In general, it is preferable to choose a q such that:

$$q(s_k^i|s_{0:k-1}^i, z_{1:k}) = q(s_k^i|s_{k-1}^i, z_k)$$

The simplest choice is;

$$q(s_k^i|s_{k-1}^i, z_k) = p(s_k^i|s_{k-1}^i)$$

Where $p(s_k^i|s_{k-1}^i)$ is part of the HMM given problem model. Besides, it simplifies the expression 3.13.

3.10 Final algorithm

1. Initialization:
   $s_0^i \sim p(s_0)$, the apriori distribution

   $\{w_o^i\}_{i=1}^{N_s} = \frac{1}{N_s}$, a uniform distribution
2. Prediction: draw new grid/set of particles

   $$x_k^i \sim p(s_k^i|s_{k-1}^i)$$
3. Update: if $i < N_s$ calculate new weights (see 3.13);

   $$w_k^i = p(z_k|s_k) \cdot w_{k-1}^i$$

And return to 2. Else continue to 4

4. Normalize:

$$w_k^i = \frac{w_k^i}{\sum_{i=1}^{N_s} w_k^i}$$

5. Calculate

$$N_{s,eff} \triangleq \frac{1}{\sum_i^{N_s} \left(w_k^i\right)^2}$$

If $N_{s,eff} < N_{th}$; Resample. Else go to 2.

4 Application of PF to Indoor Positioning [6]

4.1 State equation

In the indoor positioning case, the state, s_k, is a $4x1$ $(n = 4)$ vector;

$$\underset{4x1}{s_k} = \left[x_k, y_k, v_k^x, v_k^y\right]^T \tag{4.1}$$

Where x_k, y_k, v_k^x, v_k^y are the 2D position and velocity at time k respectively.

4.1.1 Known acceleration

The state equation describes an accelerated movement, where the 2D acceleration is known in principle;

$$s_k = \begin{pmatrix} x_k \\ y_k \\ v_k^x \\ v_k^y \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ v_{k-1}^x \\ v_{k-1}^y \end{pmatrix} + \begin{pmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \begin{pmatrix} a_k^x \\ a_k^y \end{pmatrix} + \begin{pmatrix} w_{k-1}^x \\ w_{k-1}^y \\ w_{k-1}^{v_x} \\ w_{k-1}^{v_y} \end{pmatrix} \tag{4.2}$$

And is thus linear:

$$s_k = F s_{k-1} + G u_k + w_{k-1} \tag{4.3}$$

(Not to confuse w_{k-1} here, a noise, with the particles series $\left\{w_k^i\right\}_{i=1}^{N_s}$) The acceleration is treated as an input and is supposed to be provided by an acceleration sensor (accelerometer). The noise is modeling the accelerometer inaccuracy and is supposed to be white;

$$\begin{pmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \begin{pmatrix} a_k^x + n_a^x \\ a_k^y + n_a^y \end{pmatrix} = \begin{pmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \begin{pmatrix} a_k^x \\ a_k^y \end{pmatrix} + \underbrace{\begin{pmatrix} n_{acc}^x \cdot \Delta t^2/2 \\ n_{acc}^y \cdot \Delta t^2/2 \\ n_{acc}^x \cdot \Delta t \\ n_{acc}^y \cdot \Delta t \end{pmatrix}}_{\underline{w}}$$

Hence;

$$Q = E\left(\underline{w}_k \underline{w}_k^T\right)$$
$$= \begin{pmatrix} (\sigma_{nacc}^x)^2 \cdot \Delta t^4/4 & 0 & (\sigma_{nacc}^x)^2 \cdot \Delta t^3/2 & 0 \\ 0 & \left(\sigma_{nacc}^y\right)^2 \cdot \Delta t^4/4 & 0 & \left(\sigma_{nacc}^y\right)^2 \cdot \Delta t^3/2 \\ (\sigma_{nacc}^x)^2 \cdot \Delta t^3/2 & 0 & (\sigma_{nacc}^x)^2 \cdot \Delta t^2 & 0 \\ 0 & \left(\sigma_{nacc}^y\right)^2 \cdot \Delta t^3/2 & 0 & \left(\sigma_{nacc}^y\right)^2 \cdot \Delta t^2 \end{pmatrix} \quad 4.4$$

The value of σ_{nacc}^x and σ_{nacc}^y should be provided by the accelerometer device's specifications.

4.1.2 Unknown acceleration

When the acceleration is unknown, we can regard it as a white random process. The state equation in this case is;

$$s_k = \begin{pmatrix} x_k \\ y_k \\ v_k^x \\ v_k^y \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ v_{k-1}^x \\ v_{k-1}^y \end{pmatrix} + \begin{pmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \begin{pmatrix} a_k^x \\ a_k^y \end{pmatrix}$$

$$s_k = F s_{k-1} + w_{k-1} \quad 4.5$$

Whose noise vector is:

$$\underline{w}_{k-1} = \begin{pmatrix} w_{k-1}^x \\ w_{k-1}^y \\ w_{k-1}^{v_x} \\ w_{k-1}^{v_y} \end{pmatrix} = \begin{pmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \begin{pmatrix} a_k^x \\ a_k^y \end{pmatrix} = \begin{pmatrix} a_k^x \cdot \Delta t^2/2 \\ a_k^y \cdot \Delta t^2/2 \\ a_k^x \Delta t \\ a_k^y \Delta t \end{pmatrix}$$

The noise covariance matrix is:

$$Q = E\left(\underline{w}_k \underline{w}_k^T\right) = \begin{pmatrix} (\sigma^x)^2 \cdot \Delta t^4/4 & 0 & (\sigma^x)^2 \cdot \Delta t^3/2 & 0 \\ 0 & (\sigma^y)^2 \cdot \Delta t^4/4 & 0 & (\sigma^y)^2 \cdot \Delta t^3/2 \\ (\sigma^x)^2 \cdot \Delta t^3/2 & 0 & (\sigma^x)^2 \cdot \Delta t^2 & 0 \\ 0 & (\sigma^y)^2 \cdot \Delta t^3/2 & 0 & (\sigma^y)^2 \cdot \Delta t^2 \end{pmatrix} \quad 4.6$$

σ^x and $\sigma^y$are $E(a_k^x - E(a_k^x))^2$ and $E\left(a_k^y - E(a_k^y)\right)^2$ respectively ($E(a_k^x), E\left(a_k^y\right) = 0$). A good choice would be the maximum possible acceleration on each of the axes. The null terms in the matrix are the result of lack of correlation between $a^x$and a^y.

4.1.3 $p(s_k|s_{k-1})$

In the case of 4.1.2, we chose the Normal distribution. Accordingly,

- Expectation: Fs_{k-1} (see equation 4.5)
- Variance: Q (see equation 4.6)

In general;

$$s_k|s_{k-1}\sim N(Fs_{k-1},Q)$$

4.2 Measurement equation

The observation vector is the squared distance;

$$\underset{3x1}{z_k}=\left[\left(d_k^1\right)^2,\left(d_k^2\right)^2,\left(d_k^3\right)^2\right]^T \tag{4.7}$$

And the observation equation is;

$$z_k=\begin{pmatrix}\left(d_k^1\right)^2\\ \left(d_k^2\right)^2\\ \left(d_k^3\right)^2\end{pmatrix}=\begin{pmatrix}\left(x_k^1-x_k\right)^2+\left(y_k^1-y_k\right)^2\\ \left(x_k^1-x_k\right)^2+\left(y_k^1-y_k\right)^2\\ \left(x_k^1-x_k\right)^2+\left(y_k^1-y_k\right)^2\end{pmatrix}+\begin{pmatrix}v_k^1\\ v_k^2\\ v_k^3\end{pmatrix} \tag{4.8}$$

Which is obviously nonlinear.

The noise is modeling the TOF measurement inaccuracy and is presumed white;

$$R=E\left(\underline{v}^T\underline{v}\right)=)(\sigma^v)^2\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix} \tag{4.9}$$

4.2.1 $p(z_k|x_k)$

We chose to model the sensor inaccuracy as a Normal distribution. Accordingly,

- Expectation: $\begin{pmatrix}\left(x_k^1-x_k\right)^2+\left(y_k^1-y_k\right)^2\\ \left(x_k^1-x_k\right)^2+\left(y_k^1-y_k\right)^2\\ \left(x_k^1-x_k\right)^2+\left(y_k^1-y_k\right)^2\end{pmatrix}$ (see equation 4.8)
- Variance: R (see equation 4.64.9)

In general;

$$z_k|s_k\sim N\left(\begin{pmatrix}\left(x_k^1-x_k\right)^2+\left(y_k^1-y_k\right)^2\\ \left(x_k^1-x_k\right)^2+\left(y_k^1-y_k\right)^2\\ \left(x_k^1-x_k\right)^2+\left(y_k^1-y_k\right)^2\end{pmatrix},R\right)$$

4.3 Initial state [6]

4.3.1 Least Square estimation

The PF is a sequential algorithm, and it is therefore useful to have an initial state that is as accurate as possible. One option is to solve the nonlinear system 2.3. It is also possible to formulate a linear least squares problem (true for all k, namely for $k = 1$);

$$m_k^2 \triangleq \left(d_k^2\right)^2 - \left(d_k^1\right)^2 - \left(\left(x_k^2\right)^2 - \left(x_k^1\right)^2 - \left(y_k^2\right)^2 + \left(y_k^1\right)^2\right)$$

$$m_k^3 \triangleq \left(d_k^3\right)^2 - \left(d_k^1\right)^2 - \left(\left(x_k^3\right)^2 - \left(x_k^1\right)^2 - \left(y_k^3\right)^2 + \left(y_k^1\right)^2\right)$$

Which becomes;

$$m_k^2 = (x_k^1 - x_k^2) \cdot x_k + (y_k^1 - y_k^2) \cdot y_k$$

$$m_k^3 = (x_k^1 - x_k^3) \cdot x_k + (y_k^1 - y_k^3) \cdot y_k$$

And in matrix form;

$$\begin{pmatrix} m_k^2 \\ m_k^3 \end{pmatrix} = \begin{pmatrix} (x_k^1 - x_k^2) & (y_k^1 - y_k^2) \\ (x_k^1 - x_k^3) & (y_k^1 - y_k^3) \end{pmatrix} \begin{pmatrix} x_k \\ y_k \end{pmatrix}$$

Solving for $k = 1$, we may derive $\hat{x}_1, \hat{y}_1$ to give (assuming initial velocities are zero):

$$\hat{s}_0 = \begin{pmatrix} \hat{x}_1 \\ \hat{y}_1 \\ 0 \\ 0 \end{pmatrix} \qquad 4.10$$

4.3.2 Initial state particles

We must avoid a situation of voluntary degeneration right from the beginning. An initial state of

$$\left\{w_o^i\right\}_{i=1}^{N_s} = \delta(i - i_0)$$

Where ;

$s_0^{i_0} = \hat{s}_0$ (see 4.10), is what must not happen.

Therefore, we need begin with a particles vector that includes $\hat{s}_0$, but allocates non zero probabilities to the rest of the $N_s - 1$ particles.

We chose;

$$s_0^{i_0} \sim \begin{pmatrix} N(\hat{x}_1, 0.2) \\ N(\hat{y}_1, 0.2) \\ N(0,0.1) \\ N(0,0.1) \end{pmatrix}$$

And drew N_s particles out of that. In addition, we assumed that the initial velocities are very low, hence the values of terms 3 and 4 of $s_0^{i_0}$.

5 Simulations

We have a recording of a moving tag along a known grid ($60\ cm$ between each grid); ranges from 3 anchors sampled at a $280msec$ interval.

We have estimated the track using Trilateration and PF. The PF supposed unknown acceleration. The PF's model tunable parameters are;

- The measurement noise: σ^v (measured in units of m^2)
- The state noise: σ^x, σ^y (measured in units of $m/second^2$)
- The particles number: N_s
- The resampling threshold: $N_{s,eff,th}$

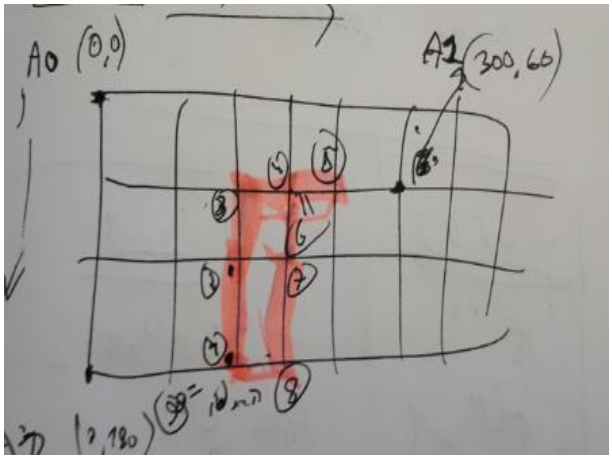

Figure 3: Ground Truth. Grid=60 cm

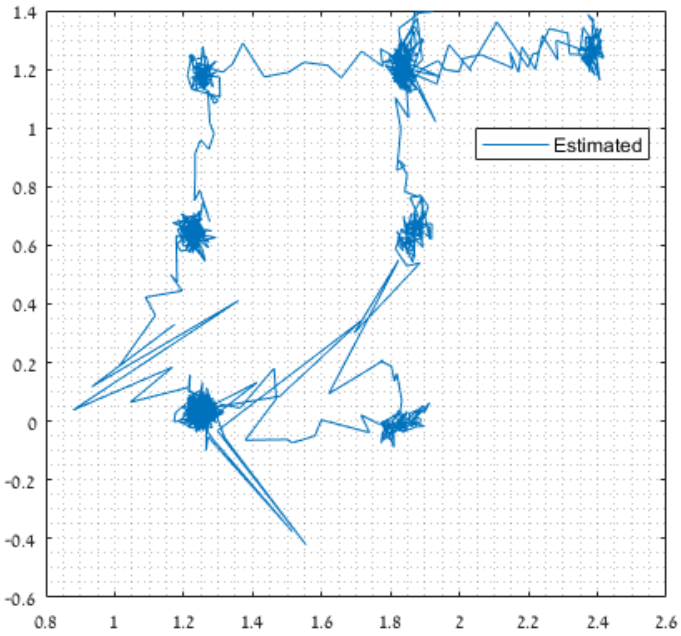

Figure 4: Trilateration

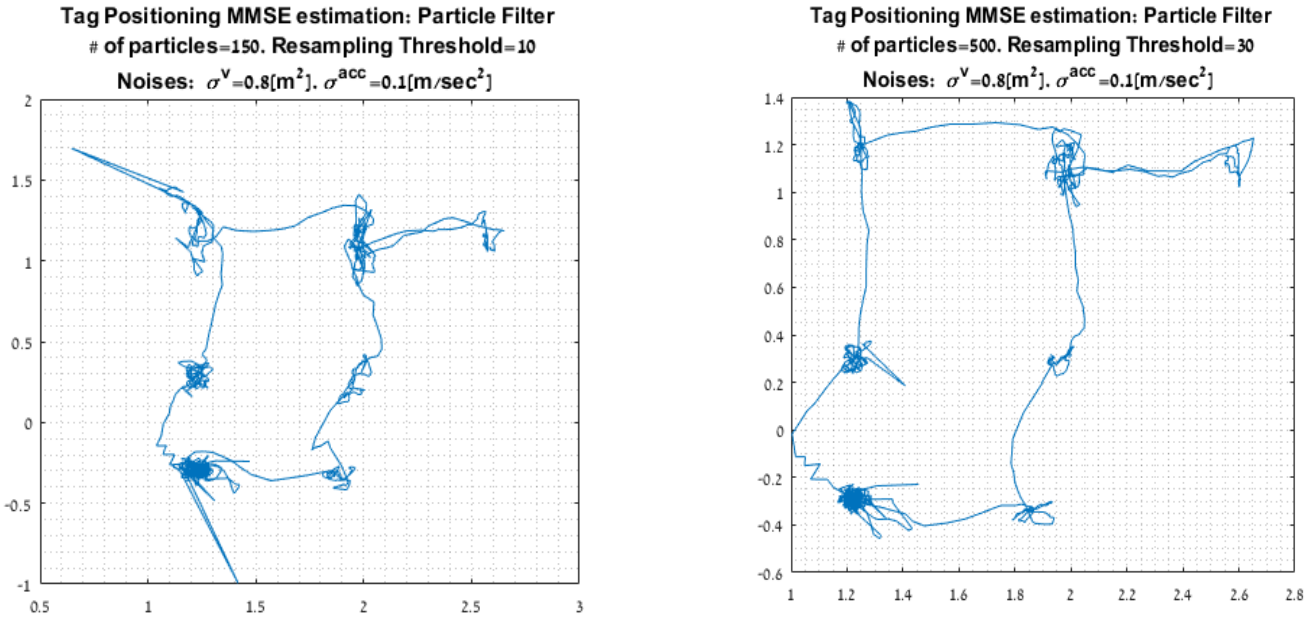

Figure 5: PF- performance versus number of particles and resampling threshold. Small& small (left), large& large (right)

We may notice that the increase of number of particles has a positive effect on the correctness of the estimation, as predicted. However, this comes with higher complexity

6 Appendix

6.1 Bayesian Recursion ([4])

The basis of the sequential algorithm is the Bayesian recursion, composed of 2 basic stages; Prediction and Measurement update/Correction. The original expressions may be found in [4] (eq. 6) and [5] (eq. 3-5). The grid based version is brought below:

1. **Prediction** ([4], equation 11)

$$p(s_k|z_{1:k-1}) = \sum_{j=1}^{N} p(s_k^i|s_{k-1}^i)p(s_{k-1}^i|z_{1:k-1}) = \sum_{j=1}^{N} p(s_k^i|s_{k-1}^i)w_{k-1|k-1}^i$$

Equivalent to:

$$w_{k|k-1}^i = \sum_{j=1}^{N} p(s_k^i|s_{k-1}^i)w_{k-1|k-1}^i \qquad 6.1$$

2. **Update** ([4] equation 10a);

$$p(s_k|z_{1:k}) = \frac{p(z_k|s_k)p(s_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \cong \frac{p(z_k|s_k)w_{k|k-1}^i}{p(z_k|z_{1:k-1})}$$

Equivalent to:

$$w_{k|k}^i = \frac{p(z_k|s_k)w_{k|k-1}^i}{p(z_k|z_{1:k-1})} \quad 6.2$$

Where ([4] equation 10b);

$$p(z_k|z_{1:k-1}) \cong \sum_{j=1}^{N} p(z_k|s_k^j)p(s_k^j|z_{1:k-1}) = \sum_{j=1}^{N} p(z_k|s_k^j)w_{k|k-1}^i \quad 6.3$$

7 References

[1] B. Gaffneey, "Considerations and Challenges in Real Time Locating Systems Design".

[2] PIXIE, "TOF VS TDOA IN LOCATION SYSTEMS," PIXIE, [Online]. Available: https://pixie-technology.com/tof-vs-dtoa/. [Accessed 7 2 2018].

[3] S. Särkkä, "Particle Filtering — Sequential Importance Resampling and Rao-Blackwellized Particle Filtering," 2012.

[4] F. Gustafsson, "Particle Filter Theory and Practice with Positioning Applications," Linköping University, 2010.

[5] M. S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE TRANSACTIONS ON SIGNAL PROCESSING,,* vol. 50, no. 2, 2002.

[6] K. Guo, Z. Qiu, C. Miao and A. H. Zaini, "Ultra-Wideband-Based Localization for Quadcopter Navigation," *Unmanned Systems,* vol. 4, no. 1, 2016.