Lecture 20: Particle Filter

These lecture notes are based on [1, Sec.15.2] and [2, Ch.4.3].

Grid- Versus Particle-based Recursive Bayesian Estimation. The grid-based recursive Bayesian estimator of the last lecture assumes that the state-space is a discretized grid which remains fixed over time. One shortcoming of this approach is that it limits the resolution of the estimate (based on the spacing of the grid-points defining the state space) and it requires the formulation of discrete-state transition dynamics which can be challenging when working with a continuous state space. An alternative but related approach based on Bayesian estimation is the *particle filter*. The particle filter numerically implements the recursive Bayesian estimator (RBE) discussed above by approximating the p.d.f. $p(\boldsymbol{x}_k)$ by a set of N state vectors called *particles*. These particles are similar to the grid cells representing states in the grid-based RBE but they are allowed to take on continuous values in the state-space and their location is dynamically adjusted as the estimation proceeds. A particle can be considered a hypothesis about the true state of the system and the collection of all N particles is an approximation of the states p.d.f.

Particle Filter

System Model Assumptions. The particle filter applies to discrete-time non-linear system and measurement equations:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \tag{1}$$
$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \tag{2}$$

where $\{\boldsymbol{w}_k\}$ and $\{\boldsymbol{v}_k\}$ are independent noise processes with known p.d.f.'s $p(\boldsymbol{w}_k)$ and $p(\boldsymbol{v}_k)$. Notice that this description of a system is very general—the dynamics and measurements are non-linear and the noise can be non-parametric. This generality is one reason the particle filter is very flexible. In fact the discrete dynamics above do not require a mathematical representation — it can be an outcome of a black-box simulation or it could represent the continuous dynamics integrated with a numerical solver over a single time-step.

Likelihood Function. As with the grid-based RBE, the particle filter requires a likelihood function that describes the probability of a state given a measurement $\mathcal{L}(\boldsymbol{x};\boldsymbol{y})$. This function is equivalent to the the probability of the measurement given the state $p(\boldsymbol{y}|\boldsymbol{x})$. That is,

$$\mathcal{L}(\boldsymbol{x};\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x}) \tag{3}$$

As discussed before, the likelihood function views the data as fixed and the state as the independent variable. For the special case of a nonlinear measurement equation with *additive* Gaussian noise $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \tag{4}$$
$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{5}$$

the likelihood function is the equation for a multivariate Gaussian p.d.f.:

$$\mathcal{L}_k(\boldsymbol{x};\boldsymbol{y}) = p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{h}_k(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{h}_k(\boldsymbol{x}))\right)}{(2\pi)^{n/2}\sqrt{|\boldsymbol{R}|}} \tag{6}$$

Initialization. As with all filtering approaches, we begin by initializing the filter with a prior guess of the system's p.d.f. $p(\boldsymbol{x}_{0|0})$. Let N denote the total number of particles that will be used to represent the state during the filtering process. These N particles can be initialized by sampling $p(\boldsymbol{x}_{0|0})$ appropriately. For example, $p(\boldsymbol{x}_{0|0})$ can be a Gaussian, a uniform distribution, or any parametric or non-parametric probability distribution that is appropriate for the problem at hand. The process of sampling the distribution can be achieved in various ways depending on its type — for Gaussians we can generate a unit-variance random vector of length N and transform it through an appropriate linear system to approximate a desired mean and covariance (as discussed in an earlier lecture). For a uniform distribution we can define a hyper-rectangle and sample it uniformly (e.g., using a Cartesian product of a grid defined over each axis). Regardless of the method used the initialization process produces N state vectors particles that are denoted

$$\mathcal{X}_{0|0} = \{\boldsymbol{x}_{0|0,i} \ \text{ for } \ i = 1, \ldots, N\} \tag{7}$$

This set of particles becomes the prior for time-step $k = 0$. Each time a measurement is received the time-step k is incremented and the following steps are performed.

Motion Update. For each of the following time-steps $k = 1, 2, 3, \ldots$ the posterior set of particles from the previous $(k-1)$th time-step

$$\mathcal{X}_{k-1|k-1} = \{\boldsymbol{x}_{k-1|k-1,i} \ \text{ for } \ i = 1, \ldots, N\} \tag{8}$$

is propagated through the motion model (4). That is, the ith particle is obtained as:

$$\boldsymbol{x}_{k|k-1,i} = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1|k-1,i}, \boldsymbol{w}_{k-1,i}) \tag{9}$$

where $\boldsymbol{w}_{k-1,i}$ is the process noise term that is sampled (independently for each particle) from the known p.d.f. $p(\boldsymbol{w}_k)$. After propagating the entire set of N particles we have a new set that represents the motion prediction

$$\mathcal{X}_{k|k-1} = \{\boldsymbol{x}_{k|k-1,i} \ \text{ for } \ i = 1, \ldots, N\} \tag{10}$$

Compute Likelihood. Given a new measurement $\boldsymbol{y}_k$, the relative likelihood of each motion-prediction particle conditioned on the measurement received $\boldsymbol{y}_k$ is computed:

$$\{q_i = \mathcal{L}_k(\boldsymbol{x}_{k|k-1,i}; \boldsymbol{y}_k) \ \text{ for } \ i = 1, \ldots, N\} \tag{11}$$

and these likelihood are normalized to sum to one as follows

$$\left\{ \bar{q}_i = \frac{q_i}{\sum_{i=1}^{N} q_i} \ \text{ for } \ i = 1, \ldots, N \right\} \tag{12}$$

The likelihood represent the weight or importance of a particular particle in the set. Note that initially the particles contained in $\mathcal{X}_{k-1|k-1}$ all had equal weight but regions of higher state probability were populated by more particles then regions that were less probable. In other words, the density of particles is used to indicate the regions of higher probability when approximating the p.d.f. with a particle filter. The likelihood or weight computed above is used only temporarily as a means to adjust this density through the resampling process described next.

Resampling. The resampling step adjusts the population of N particles by generating N new particles from the weights/likelihoods. One of the simplest approaches is as follows. Accumulate the likelihood weights $\bar{q}_i$ into a cumulative sum weight vector Q_i such that $Q_1 = q_1$, $Q_2 = q_1 + q_2$, and so on until $Q_N = 1$. For each $i = 1, 2, \ldots, N$, draw a uniformly random number $r \in [0, 1]$ and increment the integer $u = 1$ by ones until $Q_u > r$. Set the resampled particle as

$$\boldsymbol{x}_{k|k,i} = \boldsymbol{x}_{k|k-1,r} \tag{13}$$

and repeat the process for the remaining particles. This procedure leads to a set of posterior particles

$$\tilde{\mathcal{X}}_{k|k} = \{\boldsymbol{x}_{k|k,i} \text{ for } i = 1, \ldots, N\} \tag{14}$$

The process above will produce duplicate particles in regions with high likelihood and will omit low likelihood particles in the resampled set. For duplicate particles the process noise term in the motion-update step of the next iteration will cause the particles to take on unique states in the next iteration of the particle filter. The resampled set is now approximately distributed according to $p(\boldsymbol{x}_k|\boldsymbol{y}_k)$.

Compute statistics. At this point several statistics can be computed from the population of particles $\tilde{\mathcal{X}}_{k|k}$ such as the mean

$$\hat{\boldsymbol{x}}_k = 1/N \sum_{i=1}^{N} \boldsymbol{x}_{k|k,i} \,, \tag{15}$$

and covariance

$$\boldsymbol{P}_k = \sum_{i=1}^{N} \boldsymbol{x}_{k|k,i} (\boldsymbol{x}_{k|k,i})^{\mathrm{T}} - \hat{\boldsymbol{x}}_k \hat{\boldsymbol{x}}_k^{\mathrm{T}} \,. \tag{16}$$

Sample Impoverishment. When the regions of the state space where the likelihood function and the p.d.f. of the motion prediction do not have significant overlap the issue of sample impoverishment can occur. In essence, the particles are not "in the right place" to attain a high likelihood and so the filter will not produce a good estimate. Moreover, after several iterations the diversity of particles may continue to diminish producing many duplicates in regions that seem to have high-likelihood (relative to other particles) but are in fact low-likelihood relative to the overall state space. At worst the particles can collapse to a single state (i.e., all N particles being identical). One way to combat this issue to increase the number of particles, however this is computationally intensive. Other approaches are based on improving the previously described resampling step through regularization or roughening.

Regularizaton. In *regularized particle filtering* the set $\mathcal{X}_{k|k-1}$ along with the weights q_i is used to generate a continuous approximation of the p.d.f. (e.g., based on a kernel density estimation algorithm) that can then be resampled. The advantage of this approach is that the posterior particles $\mathcal{X}_{k|k}$ are not restricted to duplicate the set of motion predictions $\mathcal{X}_{k|k-1}$.

Roughening. Roughening is a mechanism to inject artificial noise into the particle population to promote the particles taking on a diverse set of states and prevent sample impoverishment. Generally this roughening step occurs after we've resampled our particle set and now have potential duplicates or particle clusters. One approach to roughening begins by computing the maximum difference across particles,

$$\boldsymbol{m}_k(l) = \max_{i,j\in\{1,\ldots,N\}} |\boldsymbol{x}_{k|k,i}(l) - \boldsymbol{x}_{k|k,i}(l)| + m_{\min} \tag{17}$$

where $m_{\min}$ is a small additive constant and $\boldsymbol{m}_k(l)$ denotes the lth element of $\boldsymbol{m}_k$ and $l \in \{1,\ldots,n\}$ is each dimension of the particle's state (the size of the state vector). Then, for each particles $i = 1,2,\ldots,N$ noise is added to give a roughened posterior set of particles

$$\mathcal{X}_{k|k} = \{\boldsymbol{x}_{k|k,i} + \Delta_i \;\text{ for }\; i = 1,\ldots,N\} \tag{18}$$

where

$$\Delta_i \sim \mathcal{N}(\mathbf{0}, K\text{diag}[\mathbf{m}_k]) \tag{19}$$

and K is a roughening gain.

Algorithm: Particle Filter

1. **System Model.** Assume the discrete-time nonlinear system and measurement equations are:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \\ \boldsymbol{y}_k &= \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{v}_k) \end{aligned}$$

   where $\{\boldsymbol{w}_k\}$ and $\{\boldsymbol{v}_k\}$ are independent noise processes with known p.d.f.'s $p(\boldsymbol{w}_k)$ and $p(\boldsymbol{v}_k)$.

2. **Initialize.** Assume that the p.d.f. of the initial state $p(\boldsymbol{x}_0)$ is known. Initialize the estimator with N particles

$$\mathcal{X}_{0|0} = \{\boldsymbol{x}_{0|0,i} \;\text{ for }\; i = 1,\ldots,N\} \tag{20}$$

   that are randomly sampled from a user-defined distribution $p(\boldsymbol{x}_0)$. In the absence of other information a uniform distribution over the state-space (or a subset of the state space) can be assumed.

3. For $k = 1,2,\ldots,$ do the following:

   (a) Obtain the measurement $\boldsymbol{y}_k$ from the sensor/data stream.

   (b) Retrieve the posterior $\mathcal{X}_{k-1|k-1}$ from the last step. This becomes the prior.

   (c) Motion-Update. Propagate the prior (posterior from the previous time-step) through the system dynamics to give a new set of particles:

$$\mathcal{X}_{k-1|k-1} = \left\{\boldsymbol{x}_{k|k-1,i} = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1|k-1,i}, \boldsymbol{w}_{k-1,i}) \;\text{ for }\; i = 1,\ldots,N\right\} \tag{21}$$

   where $\boldsymbol{w}_{k-1,i}$ is randomly generated based on the known p.d.f. of the process noise.

   (d) Compute Likelihood. After receiving measurement $\boldsymbol{y}_k$, compute the relative likelihood of each particle as

$$\{q_i = \mathcal{L}_k(\boldsymbol{x}_{k|k-1,i}; \boldsymbol{y}_k) \;\text{ for }\; i = 1,\ldots,N\} \tag{22}$$

   and normalize these likelihoods

$$\left\{\bar{q}_i = \frac{q_i}{\sum_{i=1}^{N} q_i} \;\text{ for }\; i = 1,\ldots,N\right\} \tag{23}$$

(e) Measurement Update (Resampling). Accumulate the likelihood weights $\bar{q}_i$ into a cumulative sum weight vector Q_i such that $Q_1 = q_1$ and $Q_N = 1$. For each $i = 1, 2, \ldots, N$, draw a uniformly random number $r \in [0, 1]$ and increment the integer $u = 1$ by ones until $Q_u > r$. Set the resampled particle as

$$\boldsymbol{x}_{k|k,i} = \boldsymbol{x}_{k|k-1,u} \tag{24}$$

and repeat the process for the remaining particles to give the set

$$\tilde{\mathcal{X}}_{k|k} = \{\boldsymbol{x}_{k|k,i} \text{ for } i = 1, \ldots, N\} \tag{25}$$

(f) Compute Statistics. Compute statistics as desired. For example, the mean

$$\hat{\boldsymbol{x}}_k = 1/N \sum_{i=1}^{N} \boldsymbol{x}_{k|k,i} \ , \tag{26}$$

and covariance

$$\boldsymbol{P}_k = \sum_{i=1}^{N} \boldsymbol{x}_{k|k,i} (\boldsymbol{x}_{k|k,i})^{\mathrm{T}} - \hat{\boldsymbol{x}}_k \hat{\boldsymbol{x}}_k^{\mathrm{T}} \ . \tag{27}$$

(g) Roughening (Recommended). Use (18) to roughen the particle set to give $\mathcal{X}_{k|k}$. If not using roughening, then set $\mathcal{X}_{k|k} = \tilde{\mathcal{X}}_{k|k}$.

(h) Store the posterior $\mathcal{X}_{k|k}$ to be used as the prior in the next iteration.

Example [3]

This example describes a particle-filter based tracking algorithm that enables an autonomous underwater vehicle (AUV) to detect and track nearby surface vessels. A planar, hull-mounted hydrophone array provides acoustic data to the passive sonar that uses a time-delay-and-sum beamformer to generate multiple bearing-only contacts. A tracker assimilates the contacts using a particle filter with a single-hypothesis data association strategy to estimate the position, speed, and course of nearby targets. Tracks are periodically reported to an onboard database along with a qualitative status label.

Motion Model. The target is assumed to move with approximately constant velocity according to a second-order kinematic model subject to position and speed disturbances. The target state at the kth timestep is $\mathbf{x}_k \triangleq [n_k \ \dot{n}_k \ e_k \ \dot{e}_k]^{\mathrm{T}}$ where (n_k, e_k) is the northing and easting of the target, and $(\dot{n}_k, \dot{e}_k)$ are the corresponding target speeds. The discrete-time state equation is

$$\begin{aligned}
\mathbf{x}_k &= f_k(\mathbf{x}_{k-1}, \mathbf{w}_k) && (28) \\
&= \underbrace{\begin{bmatrix} 1 & T_k & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T_k \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{F}_k} \underbrace{\begin{bmatrix} n_{k-1} \\ \dot{n}_{k-1} \\ e_{k-1} \\ \dot{e}_{k-1} \end{bmatrix}}_{\mathbf{x}_{k-1}} + \underbrace{\begin{bmatrix} (w_n)_k \\ (w_{\dot{n}})_k \\ (w_e)_k \\ (w_{\dot{e}})_k \end{bmatrix}}_{\mathbf{w}_k} \ , && (29)
\end{aligned}$$

where $\mathbf{F}_k$ is the state-transition matrix and $T_k \triangleq t_k - t_{k-1}$ is the sampling time. The sampling time is constant and corresponds to the total time to record a fixed time interval of data, referred to as a frame, from the passive sonar. The zero-mean Gaussian random disturbance vector $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ has covariance matrix $\mathbf{Q}_k \triangleq \mathrm{diag}([\sigma_p^2\ \sigma_v^2\ \sigma_p^2\ \sigma_v^2]^\mathrm{T})T_k$, where σ_p and σ_v are tuning parameters. The speed of the target $v_k \triangleq \sqrt{\dot{n}_k^2 + \dot{e}_k^2}$ is bounded by a minimum and maximum speed, i.e., $v_k \in \mathcal{V}$, where $\mathcal{V} \triangleq [v_{\min}, v_{\max}]$.

Sensor Model. The ownship state $\mathbf{p}_k \triangleq [n_k^o\ e_k^o\ \psi_k^o]^\mathrm{T}$ at time t_k is given by its northing n_k^o, easting e_k^o, and heading ψ_k^o. The bearing of a target relative to the ownship is

$$\beta_k(\mathbf{x}_k, \mathbf{p}_k) \triangleq \mathrm{atan}(\Delta E_k / \Delta N_k)\ , \tag{30}$$

where $\Delta N_k(\mathbf{x}_k, \mathbf{p}_k) \triangleq n_k - n_k^o$ and $\Delta E_k(\mathbf{x}_k, \mathbf{p}_k) \triangleq e_k - e_k^o$ are relative displacements. Both the target bearing and ownship heading are measured clockwise from north. The aspect angle of the target,

$$\alpha_k(\mathbf{x}_k, \mathbf{p}_k) \triangleq d_\measuredangle(\psi_k^o, \beta_k(\mathbf{x}_k, \mathbf{p}_k))\ , \tag{31}$$

is the conical angle measured relative to vehicle's heading (see Fig. 1), where $d_\measuredangle(\cdot,\cdot) : \mathbb{S} \times \mathbb{S} \to [0, \pi]$ returns the shortest (unsigned) angular distance between two input angles. According to

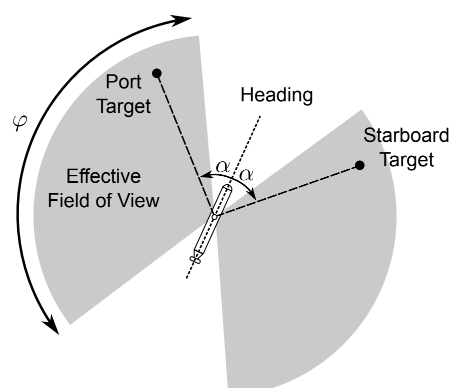


Figure 1: Geometry of the passive sonar model. The vehicle has blind spots at endfire (i.e., in front and behind). The effective field of view angle is φ. The sonar reports an aspect angle α that is port/starboard ambiguous.

(31), if a target is fore (i.e., in front of the ownship) the aspect reads zero, and if the target is aft (i.e., behind the ownship) it reads π. The measurement is port-starboard ambiguous and does not distinguish between a noise source on the port or starboard sides of an axial, planar array. For convenience, define the starboard and port bearings, $\beta_{\mathrm{star}}(\alpha, \psi) \triangleq \psi + \alpha$ and $\beta_{\mathrm{port}}(\alpha, \psi) \triangleq \psi - \alpha$, corresponding to a given aspect angle and ownship heading. The measured aspect angle is corrupted by additive, zero-mean Gaussian noise, $v_k \sim \mathcal{N}(0, \sigma_v^2)$, and subject to saturation limits due to endfire. Thus, the aspect angle sensor reports

$$y_k \triangleq h_k(\mathbf{x}_k, v_k; \mathbf{p}_k) \tag{32}$$
$$= \mathrm{sat}(\alpha_k(\mathbf{x}_k, \mathbf{p}_k) + v_k; \alpha_f, \alpha_a)\ , \tag{33}$$

where $\mathrm{sat}(x; x_{\mathrm{lb}}, x_{\mathrm{ub}})$ is a saturation function that bounds an input x between a lower bound x_{lb} and an upper bound x_{ub}. The constants $\alpha_f \triangleq (\pi - \varphi)/2$ and $\alpha_a \triangleq \pi - \alpha_f$ are the minimum (fore) and maximum (aft) aspect angles, respectively, where φ is the effective field of view angle (see Fig. 1). Targets at a range

$$r_k \triangleq \sqrt{\Delta E_k^2 + \Delta N_k^2}\ , \tag{34}$$

are detected if $r_k \in \mathcal{R}$, where $\mathcal{R} \triangleq [r_{\min}, r_{\max}]$ defines the sonar's sensing range.

Likelihood Function. A likelihood function gives the relative probability of obtaining an aspect angle measurement $y \in [0, \pi]$ for all target states $\mathbf{x} \in \mathbb{R}^4$. However, the likelihood is not necessarily a probability density function—for example, it need not integrate to unity across the target state space. The likelihood function is designed from the measurement equation (33) and other practical considerations on the target speed and sensor range limits as follows.

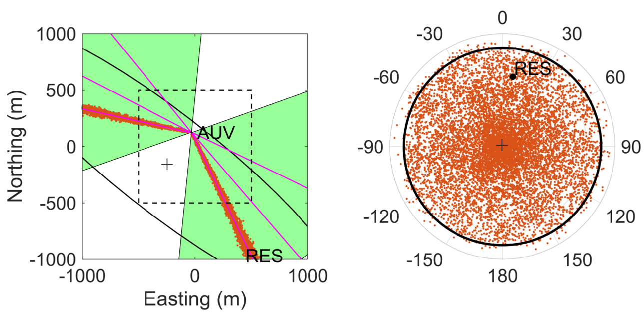
The measurement space of aspect angles is partitioned into three regions $\mathcal{E}_f \cup \mathcal{F} \cup \mathcal{E}_a = [0, \pi]$, where $\mathcal{E}_f \triangleq [0, \alpha_f]$ is the fore endfire region, $\mathcal{F} \triangleq (\alpha_f, \alpha_a)$ is the field of view, and $\mathcal{E}_a \triangleq [\alpha_a, \pi]$ is the aft endfire region. Measurements in endfire, $\mathcal{E}_a$ or $\mathcal{E}_f$, are assumed to be saturated and handled as a special case by the multi-target tracker (not described here).

States that lie outside the speed or range interval, $\mathcal{V}$ or $\mathcal{R}$, respectively, have zero likelihood. To express this condition an indicator function is used, $I_A : X \to \{0, 1\}$, that maps an element $x \in X$ to 1 if $x \in A$ and to zero otherwise. For a given state, the corresponding aspect angle and range are computed using (31) and (34), respectively. The indicator function and a Gaussian distribution about the measurement defines the likelihood

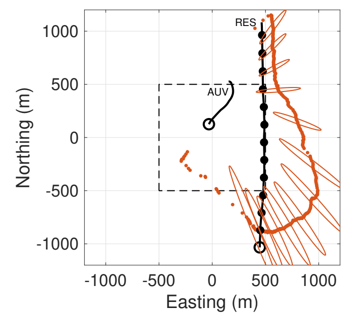
$$\mathcal{L}(y|\mathbf{x};\mathbf{p}) \triangleq I_{\mathcal{V}\times\mathcal{R}}(v, r) g(\alpha; y, \sigma_y^2)\ , \tag{35}$$

where $g(\cdot)$ denote a normally distributed random variable with probability density function $g(x; \mu, \sigma^2) \triangleq (\sigma\sqrt{2\pi})^{-1}\exp(-1/2\,(x - \mu/\sigma)^2)$ where μ is the mean and σ^2 is the variance. The ownship state is considered a known parameter. This likelihood function is used for the measurement update step of the particle filter for single-target tracking.

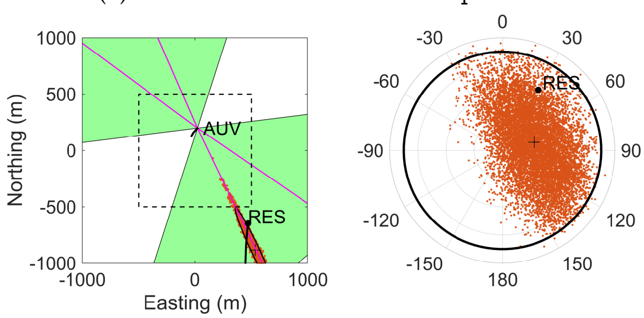
A single track is examined (Fig. 2) to illustrate the target state estimator. The data corresponds to a seven minute interval for track no. 202. The particles on the left in Fig. 2 give insight into the state of the particle filter as the track status changes. Particles are uniformly dispersed in speed and course when the track is `spawned` in Fig. 2a. As more measurements are assimilated the target is confirmed and the status becomes `detected`. The track status becomes `resolved` once the particles resolve the port/starboard ambiguity and estimate a north-east course (Fig. 2c). Later, the filter mean correctly estimates that the target is moving north and the uncertainty ellipse becomes sufficiently small so that the track status becomes `converged` (Fig. 2e). At the point of closest approach (near five minutes) the error in the estimated target state is at a minimum value (Fig. 2b, 2d, 2f).



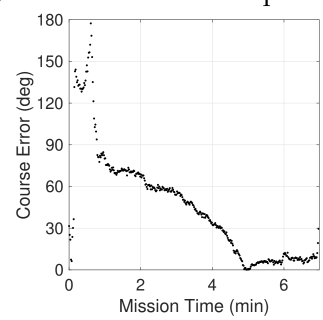
(a) Track No. 202 with status `spawned`.



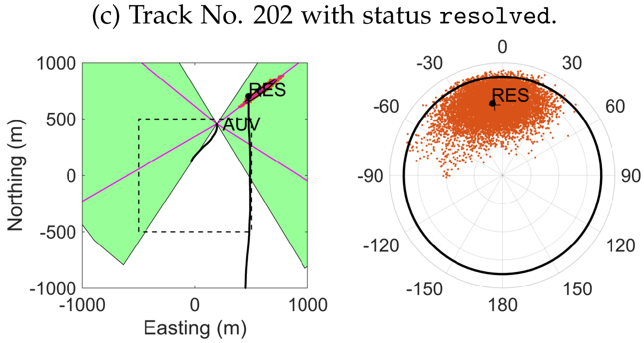
(b) Actual vs. estimated position.



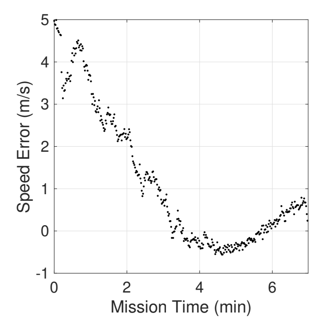
(c) Track No. 202 with status `resolved`.



(d) Error in estimated course angle.



(e) Track No. 202 with status `converged`.



(f) Error in estimated speed

Figure 2: View of internal particle filter state and performance for track no. 202. Left: Snapshots of particle filter during different stages of tracking. The green fan-shaped areas indicate the field of view of the AUV. Particles are encircled by a 95% confidence ellipse. The target in this track is the R/V Resolution, labeled "RES". Middle: Polar plot of corresponding speed and course of particles—maximum speed bound of 8 m/s is indicated by a black circle. Right: Corresponding position of AUV and target with uncertainty ellipses and error in course and speed with time.

References

[1] Dan Simon. *Optimal State Estimation: Kalman, H infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.

[2] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[3] Artur Wolek, James McMahon, Benjamin R Dzikowicz, and Brian H Houston. Tracking multiple surface vessels with an autonomous underwater vehicle: Field results. *IEEE Journal of Oceanic Engineering*, 2020.