LINMA1731 Stochastic Processes: Project

Wen Huang

March 15, 2016

1 Bearings-only Tracking Problem

The bearings-only tracking problem considers finding the location and velocity of a moving target given only the noisy bearing measurements. Specifically, a target α is moving in a space. An observer β measures the bearing θ_t of α related to β. Suppose the position and the velocity of the observer β are given, which is true in practice. The goal is to measure the position and velocity of the target α. Figure 1 shows an example of bearings-only tracking problem.

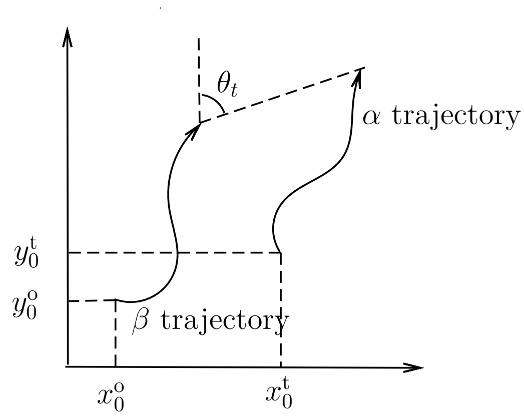

Figure 1: Illustration for a two-dimensional target α and observer β (θ is the bearing). {f1}

Bearings-only tracking problems arise in many important practical applications, e.g., submarine tracking and aircraft surveillance. Many models are built to solve this problem based on different assumptions. In this project, we suppose both the observer and the target are on a plane and the velocity of the target is nearly constant. The sequential Monte Carlo method [Sri09, Page 161 to 163], also called particle filter, is used to model this problem and estimate the target trajectory.

1.1 Notation and preliminaries

The target α, located at coordinates $(x^{\mathrm{t}}, y^{\mathrm{t}})$, moves with a nearly constant velocity vector $(\dot{x}^{\mathrm{t}}, \dot{y}^{\mathrm{t}})$, see Figure 2.

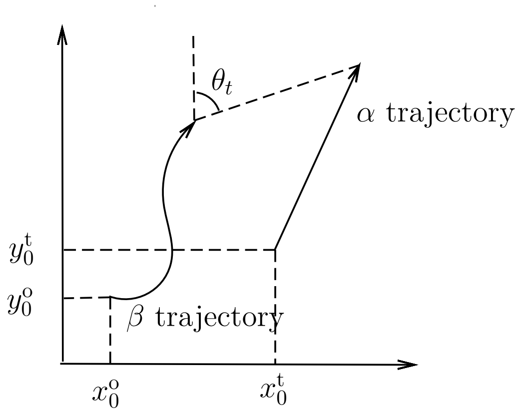

Figure 2: A target α and observer β. {f2}

The state vector of the target α is defined by $\mathbf{x}^{\mathrm{t}} = \begin{bmatrix} x^{\mathrm{t}} & y^{\mathrm{t}} & \dot{x}^{\mathrm{t}} & \dot{y}^{\mathrm{t}} \end{bmatrix}^T$. Similarly, the state vector of the observer β is $\begin{bmatrix} x^{\mathrm{o}} & y^{\mathrm{o}} & \dot{x}^{\mathrm{o}} & \dot{y}^{\mathrm{o}} \end{bmatrix}^T$. The relative state vector is defined by $\mathbf{x} := \mathbf{x}^{\mathrm{t}} - \mathbf{x}^{\mathrm{o}} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T$. The discrete-time state equation is

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k - \mathbf{U}_{k,k+1} + \epsilon_k,$$

where $\mathbf{x}_k$ is the state vector at time t_k, the time interval is constant, i.e., $t_{k+1} - t_k = T, \forall k$, $\mathbf{F} \in \mathbb{R}^{4\times 4}$, and ϵ_k is noise.

The observer β measures the angle θ_t from β to α, referenced to the y-axis, see Figure 2. The measurement z_k at t_k is given by

$$z_k = h(\mathbf{x}_k) + w_k,$$

where $h(\mathbf{x}_k)$ is the true bearing angle defined by x and y, and w_k is a zero mean independent Gaussian noise with variance σ_θ^2.

1.2 Questions

Question 1: Derive the matrix $\mathbf{F}$, $\mathbf{U}_{k,k+1}$, and $h(\mathbf{x}_k)$. (Hint: $\mathbf{F}$ is independent of k and $\mathbf{U}_{k,k+1}$ is independent of $\mathbf{x}^{\mathrm{t}}$.)

Question 2: Suppose $\epsilon_k = \mathbf{\Gamma}\mathbf{v}_k$ is noise on the acceleration, where

$$\mathbf{\Gamma} = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix},$$

$\mathbf{v}_k$ is a 2×1 i.i.d process noise vector with $\mathbf{v}_k \sim \mathcal{N}(0, \sigma_a^2 I_2)$, and σ_a is a scalar. Let $T = 0.5$, $\mathbf{x}_0 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T$, and $\mathbf{U}_{k,k+1} = 0$ for all k. Choose various values of σ_a and σ_θ and write a Matlab program to simulate the system and the observation process for $k = 1, 2, \ldots 200$. Plot the trajectory of relative positions (x, y). Plot the measure z/π versus the time t. Comment on your Matlab code and your observation.

Question 3: Use the same assumptions as in Question 2 with $\sigma_a^2 = 0.01$ and $\sigma_\theta^2 = 0.01$. Write a Matlab program to implement a sequential Monte Carlo algorithm [Sri09, Page 161 to 163]. Run the simulation 5000 times, i.e., $n = 5000$. Plot the histograms of x and y at $k = 1, 50, 100, 200$. Use the samples at each k to estimate the trajectory and plot the trajectory. Comment on your Matlab code and your observation.

Question 4: Let us consider an artificial submarine tracking problem shown in Figure 3. The blue and red dots in the left figure show the trajectories of an observer submarine and a target submarine respectively. The right figure shows the relative trajectory. Please find the attached "data.mat" file for the trajectories.

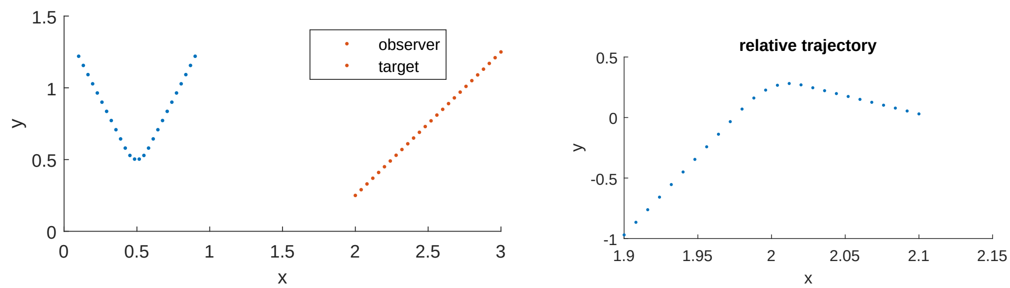

Figure 3: Left: trajectories of an observer and a target; Right: relative trajectory, i.e., the trajectory of the target minus the trajectory of the observer. {f7}

Suppose we know the noisy bearing measurements z and the state vector $\mathbf{x}^{\mathrm{o}}$ of the observer. The goal is to recover the target state vector $\mathbf{x}^{\mathrm{t}}$ or equivalently the relative state vector $\mathbf{x}$. Let the process noise ϵ_k have the form in Question 2. In practice, we have the following initialization: i), the initial relative distance follows $\bar{r} \sim \mathcal{N}(r, \sigma_r^2)$; ii), the initial bearing follows $\bar{\theta} \sim \mathcal{N}(\theta, \sigma_\theta^2)$; iii), the initial speed of the target follows $\bar{s} \sim \mathcal{N}(s, \sigma_s^2)$, and iv), the initial course of the target follows $\bar{c} \sim \mathcal{N}(c, \sigma_c^2)$. Let $n = 5000$, $T = 1$, $\sigma_a^2 = 10^{-6}$, $\sigma_r^2 = 0.1$, $\sigma_\theta^2 = 10^{-4}$, $\sigma_s^2 = 0.1$, and $\sigma_c^2 = 0.1$. The values r, θ, s, c, and the noisy measurements z are given in the attached "data.mat" file. Write a Matlab program that uses the sequential Monte Carlo algorithm [Sri09, Page 161 to 163] to track the state vector of $\mathbf{x}^{\mathrm{t}}$ or $\mathbf{x}$ and plot the following in a same figure with two subplots for a given k. The first subplot includes: i), the trajectory of β; ii), the true trajectory of α; iii), the estimated trajectory of α from the initial iteration to the k-th iteration, iv), the positions $(x_k^{\mathrm{t}}, y_k^{\mathrm{t}})$ of the n particles before resampling, and v), the positions $(x_k^{\mathrm{t}}, y_k^{\mathrm{t}})$ of the n particles after resampling. The second subplot includes i), the velocities $(\dot{x}_k^{\mathrm{t}}, \dot{y}_k^{\mathrm{t}})$ of the n particles before resampling, ii) after resampling, and iii), the true velocity. Give five figures corresponding to $k = 1, 2, 3, 15$, and 26. Comment on your Matlab code and your observation.

Question 5: It is known that when the process noise ϵ_k is small, the resampling step may yield a serious problem of loss of diversity among the particles. In order to observe this phenomenon, run the bearings-only tracking problem in Question 4 with identical parameters except $\sigma_a^2 = 0$. Plot i) a semi-logarithmic figure of the time versus the number of different particles and ii) a figure including the trajectory of β, the true trajectory of α and the estimated trajectory of α. One approach to improve the diversity among the particles is to use a regularized particle filter when $N_{\mathrm{eff}} < N_{\mathrm{th}}$, where $N_{\text{eff}} = \frac{1}{\sum_{i=1}^{n}(\tilde{w}_k^i)^2}$, $\tilde{w}_k^i$ denotes the weight of the i-th particle at the k-th iteration and the threshold N_{th} is set to be $n/3$. Write a Matlab program which uses the post-regularized particle filter [MOF01, Algorithm on Page 254] (or called the regularised sequential Monte Carlo method) with Gaussian kernel and h_{opt} given by [MOF01, (12.2.7)] for the bearings only tracking problem and plot the same figures described above. Comment on your Matlab code and your observations.

Question 6: The Carmér-Rao lower bound (CRLB) provides a lower bound for second-order (mean-squared) error for nonlinear filtering, i.e.,

$$P_k := \mathbb{E}\left\{(\mathbf{x}_k - \mathbb{E}(\mathbf{x}_k))(\mathbf{x}_k - \mathbb{E}(\mathbf{x}_k))^T\right\} \geq J_k^{-1},$$

where the inequality means that $P_k - J_k^{-1}$ is a positive semidefinite matrix and J_k is the information matrix. A realistic amount of process noise in the target trajectory makes a very small impact on the Carmér-Rao lower bound for bearing-only tracking. Therefore, we suppose $\epsilon_k = 0$, which yields the following recursive formula

$$J_{k+1} = \mathbf{F}^{-T} J_k \mathbf{F}^{-1} + \frac{1}{\sigma_\theta^2} H_{k+1} H_{k+1},$$

where $H_{k+1} = \nabla_{x_{k+1}} h(\mathbf{x}_{k+1})$. The recursion is initialized by $J_1 = P_1^{-1}$, where

$$P_1 = \begin{bmatrix} P_{xx} & P_{xy} & 0 & 0 \\ P_{yx} & P_{yy} & 0 & 0 \\ 0 & 0 & P_{\dot{x}\dot{x}} & P_{\dot{x}\dot{y}} \\ 0 & 0 & P_{\dot{y}\dot{x}} & P_{\dot{y}\dot{y}} \end{bmatrix},$$

$\begin{bmatrix} P_{xx} & P_{xy} \\ P_{yx} & P_{yy} \end{bmatrix}$ is the covariance matrix of $\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = \begin{bmatrix} \bar{r}\sin\bar{\theta} \\ \bar{r}\cos\bar{\theta} \end{bmatrix}$, and $\begin{bmatrix} P_{\dot{x}\dot{x}} & P_{\dot{x}\dot{y}} \\ P_{\dot{y}\dot{x}} & P_{\dot{y}\dot{y}} \end{bmatrix}$ is the covariance matrix of $\begin{bmatrix} \bar{\dot{x}} \\ \bar{\dot{y}} \end{bmatrix} = \begin{bmatrix} \bar{s}\sin\bar{c} - \dot{x}_0^{\text{o}} \\ \bar{s}\cos\bar{c} - \dot{y}_0^{\text{o}} \end{bmatrix}$. It can be shown that $P_{xx} = r^2\sigma_\theta^2\cos^2\theta + \sigma_r^2\sin^2\theta$, $P_{yy} = r^2\sigma_\theta^2\sin^2\theta + \sigma_r^2\cos^2\theta$, $P_{xy} = P_{yx} = (\sigma_r^2 - r^2\sigma_\theta^2)\sin\theta\cos\theta$, $P_{\dot{x}\dot{x}} = s^2\sigma_c^2\cos^2 c + \sigma_s^2\sin^2 c$, $P_{\dot{y}\dot{y}} = s^2\sigma_c^2\sin^2 c + \sigma_s^2\cos^2 c$, and $P_{\dot{x}\dot{y}} = P_{\dot{y}\dot{x}} = (\sigma_s^2 - s^2\sigma_c^2)\sin c\cos c$. Write a Matlab program to compute the CRLB for the problem in Question 4 with the same parameters. Plot the following two plots in a figure i), the CRLB of RMS

$$\text{CRLB}(\text{RMS}_k) = \sqrt{J_k^{-1}(1,1) + J_k^{-1}(2,2)} \qquad (1.1) \quad \{\texttt{e3}\}$$

versus time, where $J_k(i,j)$ denote the i-th row and j-th column entry of J_k; ii), the root-mean square (RMS) position error

$$\text{RMS}_k = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{x}_k^i - x_k)^2 + (\hat{y}_k^i - y_k)^2} \qquad (1.2) \quad \{\texttt{e4}\}$$

versus time, where (x_k^i, y_k^i) denote the true relative position at k and $(\hat{x}_k^i, \hat{y}_k^i)$ denote the estimated relative position at time k at the ith simulation. Comment on your Matlab code and your observation.

References

[MOF01] C. Musso, N. Oudjane, and LeGland F. *Improving regularised particle filters.* Springer, 2001.

[Sri09] A. Srivastava. *Computational methods in statistics.* Florida State University, 2009. `http://stat.fsu.edu/~anuj/pdf/classes/CompStatII10/BOOK.pdf`.