# SP第一次大作业

无研223 刘子源 2022310709

## 一、背景介绍

为了获取传感器节点的位置信息，我们使用基于到达时间（TOA）的定位方法，一个节点通过测量另一个节点所发射电磁波信号的传输延时来推算节点间的距离，进而求解节点位置坐标。

需要注意的有：两个节点的驱动时钟可能不同步；网络中一般会有锚点；定位方式有单点定位和协作定位等。

## 二、问题建模

网络中的节点分为锚点和待定位的节点，为方便书写，重新定义节点坐标为$p_n=(x_n,y_n)$，其中当$1\leq n\leq M$时表示锚点的已知坐标，当$M+1\leq N$时表示待定位节点的未知坐标。

锚点的时钟是同步的，未知节点相对于锚点有时钟偏差，记$t_u^{(n)}$表示节点n相对于锚点的坐标，其中$M+1\leq n<N$，可认为当$1\leq n\leq M$时$t_u^{(n)}=0$。

传感器网络中任意两个节点n与k间的TOA观测量记为$\rho_{k_1,k_2}$，每个观测量包含方差为$\sigma_{k_1,k_2}$的白噪声，这里我们认为是高斯白噪声，不同观测量间的噪声相互独立。记$x=[x_1,\dots,x_M,x_{M+1},\dots,x_N]^T$，$y=[y_1,\dots,y_M,y_{M+1},\dots,y_N]^T$为待定节点的横纵坐标，则有

$$\begin{gathered}\rho_{n,k}=\frac{\sqrt{(x_n-x_k)^2+(y_n-y_k)^2}}{c}-t_u^{(n)}+w_{n,k}=\frac{d_{k,l}}{c}-t_u^{(n)}+w_{n,k}=t_{n,k}+\omega_{n,k}\\w_{n,k}\sim\mathbb{N}(0,\sigma_{n,k}^2),\ w_{n,k}\ iid\end{gathered}\tag{1}$$

这里我们认为是一个对称的场景，即$\rho_{n,k}=\rho_{k,n},\sigma_{n,k}=\sigma_{k,n}$。由于锚点的位置已知，令$\sigma_{n,k}^2=\infty,1\leq n,k\leq M$；节点无需测量自己的位置，令$\sigma_{n,k}^2=\infty,n=k$。

现在我们需要根据观测到的$\rho_{n,k}$来估计节点的位置坐标$(x,y)$。

## 三、CRLB推导

值得注意的是，在Reference1中规定当$k>l$时$\sigma_{k,l}^2=\infty$，这是因为如果$\rho_{k,l}$和$\rho_{l,k}$均可获得的话，时间偏差可通过平均的方法消去：

$$\begin{aligned}&\frac{1}{2}\left(\hat{\rho}_{k,l}+\hat{\rho}_{l,k}\right)\\=&\frac{1}{2}\left(\frac{d_{k,l}}{c}+t_u^{(k)}-t_u^{(l)}+\omega_{k,l}+\frac{d_{k,l}}{c}-t_u^{(k)}+t_u^{(l)}+\omega_{l,k}\right)\\=&\frac{d_{k,l}}{c}+\frac{\omega_{k,l}+\omega_{l,k}}{2}\end{aligned}\tag{2}$$

由于本题中没有强调TOA的获取顺序问题，因此我们认为$\rho_{k,l}$和$\rho_{l,k}$是对称的，所以在下述分析中不将时间偏差作为待估计参数，所以Fisher信息矩阵可由（3）化简为（4）：

$$I_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{\tau}}=\begin{bmatrix}\boldsymbol{I}_{x,x}&\boldsymbol{I}_{x,y}&\boldsymbol{I}_{x,\tau}\\\boldsymbol{I}_{x,y}^T&\boldsymbol{I}_{y,y}&\boldsymbol{I}_{y,\tau}\\\boldsymbol{I}_{x,\tau}^T&\boldsymbol{I}_{y,\tau}^T&\boldsymbol{I}_{\tau,\tau}\end{bmatrix}\tag{3}$$

$$I_{\boldsymbol{x},\boldsymbol{y}}=\begin{bmatrix}I_{x,x}&I_{x,y}\\I_{x,y}^T&I_{y,y}\end{bmatrix}\tag{4}$$

由Reference1公式（5）可知，对于$[x,y]$的任意无偏估计，有

$$\operatorname{conv}\begin{bmatrix}\hat{\boldsymbol{x}}\\\hat{\boldsymbol{y}}\end{bmatrix}\geq\begin{bmatrix}I_{x,x}&I_{x,y}\\I_{x,y}^T&I_{y,y}\end{bmatrix}^{-1}=\boldsymbol{I}^{-1}\tag{5}$$

由Reference1公式（6）可知，对于坐标$p_n=(x_n,y_n)$的估计有

$$\operatorname{conv}\begin{bmatrix}\hat{x_n}\\\hat{y_n}\end{bmatrix}\geq\begin{bmatrix}[I^{-1}]_{n,n}&[I^{-1}]_{n,n+N}\\[I^{-1}]_{n+N,n}&[I^{-1}]_{n+N,n+N}\end{bmatrix}^{-1}\tag{6}$$

由Reference1公式（7）可知，当假设噪声是高斯的，Fisher信息矩阵可通过Slepian–Bang公式即公式（6）求解

$$[\boldsymbol{I}_{\mu,\nu}]_{m,n}=\sum_{k=1}^{N}\sum_{l=1}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial t_{k,l}}{\partial\mu_n}\cdot\frac{\partial t_{k,l}}{\partial\nu_m}\tag{7}$$

其中$\mu,v$为带估计矢量参数，在本问题中为$x,y$，代入（1）中可得

$$\begin{cases}\frac{1}{\hat{c}}\frac{x_n-x_l}{\sqrt{(x_n-x_k)^2+(y_n-y_k)^2}},k=n\end{cases}$$

$$\frac{\partial \rho_{k,l}}{\partial x_n}=\begin{cases}\frac{1}{c}\frac{x_n-x_l}{\sqrt{(x_n-x_l)^2+(y_n-y_l)^2}}, l=n\\ \frac{1}{c}\frac{x_n-x_k}{\sqrt{(x_n-x_k)^2+(y_n-y_k)^2}}, l=n\\ 0,\ \text{else}\end{cases} \tag{8}$$

$$\frac{\partial \rho_{k,l}}{\partial y_n}=\begin{cases}\frac{1}{c}\frac{y_n-y_l}{\sqrt{(x_n-x_l)^2+(y_n-y_l)^2}}, k=n\\ \frac{1}{c}\frac{y_n-y_k}{\sqrt{(x_n-x_k)^2+(y_n-y_k)^2}}, l=n\\ 0,\ \text{else}\end{cases} \tag{9}$$

#### 3.1单点定位

单点定位只根据待定节点与锚点之间的TOA数据来进行估计，即当$M+1\le n,k\le N$时$\sigma_{n,k}=\infty$，所以（3）中的$I_{xx},Iyy,I_{xy}$可化简为

$$\begin{aligned}[I_{x,x}]_{m,n}&=\sum_{k=1}^{M}\sum_{l=M+1}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial x_m}+\sum_{k=M+1}^{N}\sum_{l=1}^{M}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial x_m}\\ [I_{x,y}]_{m,n}&=\sum_{k=1}^{M}\sum_{l=M+1}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial y_m}+\sum_{k=M+1}^{N}\sum_{l=1}^{M}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial y_m}\\ [I_{y,y}]_{m,n}&=\sum_{k=1}^{M}\sum_{l=M+1}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial y_n}\frac{\partial\rho_{k,l}}{\partial y_m}+\sum_{k=M+1}^{N}\sum_{l=1}^{M}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial y_n}\frac{\partial\rho_{k,l}}{\partial y_m}\end{aligned} \tag{10}$$

由（5）（6）可知，只有当$m=n=k$或$m=n=l$时，$[I_{\mu,v}]_{m,n}\neq 0$，即（10）式可化简为

$$\begin{aligned}[I_{x,x}]_{n,n}&=\frac{1}{c^2}\sum_{l=M+1}^{N}\frac{2}{\sigma_{n,l}^2}\frac{(x_n-x_l)^2}{(x_n-x_l)^2+(y_n-y_l)^2}\\ [I_{x,y}]_{n,n}&=\frac{1}{c^2}\sum_{l=M+1}^{N}\frac{2}{\sigma_{n,l}^2}\frac{(x_n-x_l)(y_n-y_l)}{(x_n-x_l)^2+(y_n-y_l)^2}\\ [I_{y,y}]_{n,n}&=\frac{1}{c^2}\sum_{l=M+1}^{N}\frac{2}{\sigma_{n,l}^2}\frac{(y_n-y_l)^2}{(x_n-x_l)^2+(y_n-y_l)^2}\end{aligned} \tag{11}$$

将（11）代入到（6）式中，即可得到待定位的节点坐标的CRLB。

#### 3.2协作定位

对于协作定位，有

$$\begin{aligned}[I_{x,x}]_{m,n}&=\sum_{k=1}^{M}\sum_{l=M+1}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial x_m}+\sum_{k=M+1}^{N}\sum_{l=1}^{M}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial x_m}+\sum_{k=M+1l=M+1}^{N}\sum_{k,l}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial x_m}\\ [I_{x,y}]_{m,n}&=\sum_{k=1}^{M}\sum_{l=M+1}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial y_m}+\sum_{k=M+1}^{N}\sum_{l=1}^{M}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial y_m}+\sum_{k=M+1l=M+1}^{N}\sum_{k,l}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial x_n}\frac{\partial\rho_{k,l}}{\partial y_m}\\ [I_{y,y}]_{m,n}&=\sum_{k=1}^{M}\sum_{l=M+1}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial y_n}\frac{\partial\rho_{k,l}}{\partial y_m}+\sum_{k=M+1}^{N}\sum_{l=1}^{M}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial y_n}\frac{\partial\rho_{k,l}}{\partial y_m}+\sum_{k=M+1l=M+1}^{N}\sum_{k,l}^{N}\frac{1}{\sigma_{k,l}^2}\frac{\partial\rho_{k,l}}{\partial y_n}\frac{\partial\rho_{k,l}}{\partial y_m}\end{aligned} \tag{12}$$

进一步化简可得

$$\begin{aligned}[I_{x,x}]_{m,n}&=\begin{cases}\frac{1}{c^2}\sum_{l=1}^{N}\frac{2}{\sigma_{n,l}^2}\frac{(x_n-x_l)^2}{(x_n-x_l)^2+(y_n-y_l)^2}, m=n\\ \frac{1}{c^2}\frac{2}{\sigma_{m,n}^2}\frac{(x_m-x_n)^2}{(x_m-x_n)^2+(y_m-y_n)^2}, m\neq n\end{cases}\\ [I_{x,y}]_{m,n}&=\begin{cases}\frac{1}{c^2}\sum_{l=1}^{N}\frac{2}{\sigma_{n,l}^2}\frac{(x_n-x_l)(y_n-y_l)}{(x_n-x_l)^2+(y_n-y_l)^2}, m=n\\ \frac{1}{c^2}\frac{2}{\sigma_{m,n}^2}\frac{(x_n-x_m)(y_n-y_m)}{(x_m-x_n)^2+(y_m-y_n)^2}, m\neq n\end{cases}\\ [I_{y,y}]_{m,n}&=\begin{cases}\frac{1}{c^2}\sum_{l=1}^{N}\frac{2}{\sigma_{n,l}^2}\frac{(y_n-y_l)^2}{(x_n-x_l)^2+(y_n-y_l)^2}, m=n\\ \frac{1}{c^2}\frac{2}{\sigma_{m,n}^2}\frac{(y_n-y_m)^2}{(x_m-x_n)^2+(y_m-y_n)^2}, m\neq n\end{cases}\end{aligned} \tag{13}$$

将（13）代入到（6）式中，即可得到待定位的节点坐标的CRLB。

## 四、MLE求解

首先写出TOA观测量的累计概率密度函数，这里我们假设观测噪声是高斯白噪声且彼此独立：

$$f(\mathbf{p};\mathbf{x},\mathbf{y})=\frac{1}{\prod_{m=1}^{N}\prod_{n=1}^{N}(\sqrt{2\pi}\sigma_{m,n})}\exp\left(-\sum_{m=1}^{N}\sum_{n=1}^{N}\frac{1}{2\sigma_{m,n}^2}\left(\rho_{m,n}-\frac{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}}{c}+t_n^{(n)}\right)^2\right) \tag{14}$$

#### 4.1单点定位

对于单点定位，上式可改写为：

$$f(\rho;\mathbf{x},\mathbf{y})=\frac{1}{\prod_{m=1}^{M}\prod_{n=M+1}^{N}\sqrt{2\pi}\sigma_{m,n}}\exp\left(-\sum_{m=1}^{M}\sum_{n=M+1}^{N}\frac{1}{2\sigma_{m,n}^2}\left(\rho_{m,n}-\frac{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}}{c}+t_n^{(n)}\right)^2\right) \tag{15}$$

对参数$\theta$求导并令导数为0就可求解MLE估计量，即令下式为0，求解得到的$\hat{x}_n,\hat{y}_n$就是待定位坐标的MLE估计。

$$
\begin{aligned}
\frac{\partial \ln f(\mathbf{p};\mathbf{x},\mathbf{y})}{\partial x_n} &= \left(\sum_{m=1}^{M}\frac{1}{\sigma_{m,n}^2}(\rho_{m,n}-t_{m,n})^*\frac{\partial t_{m,n}}{\partial x_n}\right) \\
&= \frac{1}{c}\sum_{m=1}^{M}\frac{1}{\sigma_{m,n}^2}\left(\rho_{m,n}-\frac{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}}{c}+t_n^{(n)}\right)*\frac{(x_n-x_m)}{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}} \\
\frac{\partial \ln f(\mathbf{p};\mathbf{x},\mathbf{y})}{\partial y_n} &= \left(\sum_{m=1}^{M}\frac{1}{\sigma_{m,n}^2}(\rho_{m,n}-t_{m,n})^*\frac{\partial t_{m,n}}{\partial y_n}\right) \\
&= \frac{1}{c}\sum_{m=1}^{M}\frac{1}{\sigma_{m,n}^2}\left(\rho_{m,n}-\frac{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}}{c}+t_u^{(n)}\right)*\frac{(y_n-y_m)}{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}}
\end{aligned}
\tag{16}
$$

**4.2协作定位**

对于协作定位，累计概率密度函数可改写为：

$$
f(\rho;\mathbf{x},\mathbf{y})=\frac{1}{\prod_{m=1}^{N}\prod_{n=M+1}^{N}(\sqrt{2\pi}\sigma_{m,n})}\exp\left(-\sum_{m=1}^{N}\sum_{n=M+1}^{N}\frac{1}{2\sigma_{m,n}^2}\left(\rho_{m,n}-\frac{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}}{c}+t_n^{(n)}\right)^2\right) \tag{17}
$$

其对$x_n$和$y_n$的导数为

$$
\begin{aligned}
\frac{\partial \ln f(\mathbf{p};\mathbf{x},\mathbf{y})}{\partial x_n} &= \left(\sum_{m=1}^{N}\frac{1}{\sigma_{m,n}^2}(\rho_{m,n}-t_{m,n})*\frac{\partial t_{m,n}}{\partial x_n}\right)=\frac{1}{c}\sum_{m=1}^{N}\frac{1}{\sigma_{m,n}^2}\left(\rho_{m,n}-\frac{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}}{c}+t_n^{(n)}\right)*\frac{(x_n-x_m)}{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}} \\
\frac{\partial \ln f(\rho;\mathbf{x},\mathbf{y})}{\partial y_n} &= \left(\sum_{m=1}^{N}\frac{1}{\sigma_{m,n}^2}(\rho_{m,n}-t_{m,n})*\frac{\partial t_{m,n}}{\partial y_n}\right)=\frac{1}{c}\sum_{m=1}^{N}\frac{1}{\sigma_{m,n}^2}\left(\rho_{m,n}-\frac{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}}{c}+t_n^{(n)}\right)*\frac{(y_n-y_m)}{\sqrt{(x_m-x_n)^2+(y_m-y_n)^2}}
\end{aligned}
\tag{18}
$$

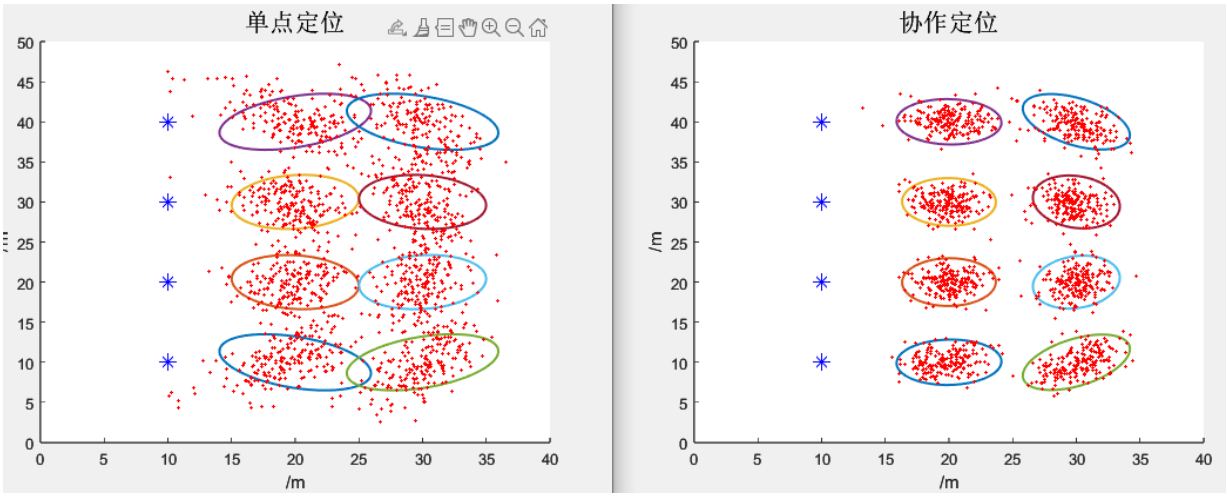
令导数为0，求解即可得到待定位坐标的MLE估计。

## 五、仿真验证

本次实验采用梯度下降法迭代求解，迭代方程为式（16）的指数部分（其余部分不含未知参数无需加入计算）。

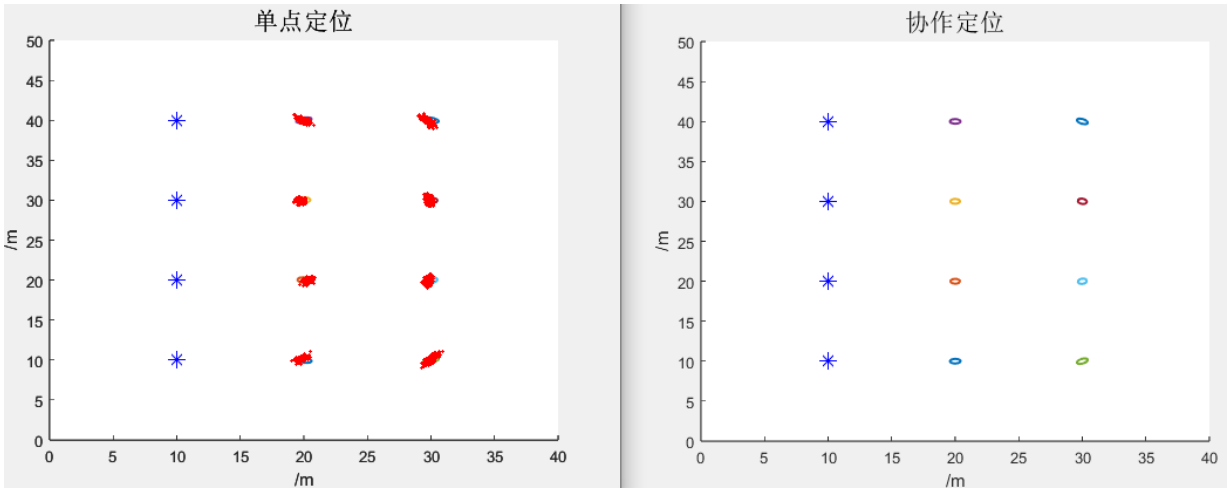
设置噪声标准差为$10^{-8}$；CRLB的椭圆置信区间为$99\%$；时钟偏差为$10^{-8}\sim10^{-9}$量级，否则算法无法收敛。

仿真结果如下，显然协作定位的效果要优于单点定位：




其中，单点定位迭代次数为357次，协作定位迭代次数为206次，协作定位的收敛速度更快。

为与理论推导的结果对比，设置TOA观测量噪声标准差为$10^{-9}$，得到结果如下：




## 六、总结

1.单点定位与协作定位均可以通过TOA观测量估计待定位节点的位置，且CRLB可以很好地描述两个算法的性能；

2.由于协作定位利用了待定位节点间的TOA观测量，其性能要优于单点定位。协作定位的收敛速度更快，CRLB的置信区间更小，仿真结果更集中在真值附近。

## 七、源代码

主文件

```
clear all,close all,clc;

N = 200;
noise_var = 1e-8;%噪声方差
clk_bias = 1e-9;%时钟偏差
bais_max = 10;
alpha = 0.05;
i_thr = 0.05;
c = 3e8;

ax = zeros(12,2);
for i = 1:12
    ax(i,1) = 10*(ceil(i/4));
    ax(i,2) = 10*(i-4*(ceil(i/4)-1));
end

ut = zeros(12,1);
ut(5:12) = clk_bias*(1-2*rand([8,1]));

Rho = zeros(12,12);
for i = 1:12
    for j = i+1:12
       Rho(i,j) = sqrt((ax(i,1)-ax(j,1))^2+(ax(i,2)-ax(j,2))^2)/c - ut(j);
       Rho(j,i) = Rho(i,j);
    end
end

ax_est_new_1 = zeros(N,12,2);
ite_num_1 = zeros(N,1);
ax_est_new_2 = zeros(N,12,2);
ite_num_2 = zeros(N,1);
for i = 1:N
    noise = normrnd(0,noise_var,[12,12]);
    for j = 1:12
        noise(j,j) = 0;
        for k = j+1:12
           noise(k,j) = noise(j,k);
        end
    end
    TOA1 = Rho + noise;

    [ax_est_new_1(i,:,:),ite_num_1(i)] = GradientDescent(TOA1,ax,bais_max,1,alpha,i_thr,ut,noise_var);
    [ax_est_new_2(i,:,:),ite_num_2(i)] = GradientDescent(TOA1,ax,bais_max,2,alpha,i_thr,ut,noise_var);
end
```

```
fprintf('单点定位迭代次数为：%d\n',mean(ite_num_1));
fprintf('协作定位迭代次数为：%d\n',mean(ite_num_2));

Is_xx = zeros(8,8);
Is_xy = zeros(8,8);
Is_yy = zeros(8,8);
for i = 1:8
    for j = 1:8
        if (j ~= i)
            Is_xx(i,i) = Is_xx(i,i) + (ax(i+4,1)-ax(j+4,1))*(ax(i+4,1)-ax(j+4,1)) / ((ax(i+4,1)-ax(j+4,1))^2+(ax(i+4,2)-ax(j+4,2))^2) / c^2 / noise_var^2;
            Is_xy(i,i) = Is_xy(i,i) + (ax(i+4,1)-ax(j+4,1))*(ax(i+4,2)-ax(j+4,2)) / ((ax(i+4,1)-ax(j+4,1))^2+(ax(i+4,2)-ax(j+4,2))^2) / c^2 / noise_var^2;
            Is_yy(i,i) = Is_yy(i,i) + (ax(i+4,2)-ax(j+4,2))*(ax(i+4,2)-ax(j+4,2)) / ((ax(i+4,1)-ax(j+4,1))^2+(ax(i+4,2)-ax(j+4,2))^2) / c^2 / noise_var^2;
        end
    end
end
Is = [Is_xx,Is_xy;Is_xy',Is_yy];
IsO = inv(Is);

figure,hold on;
Is_point = zeros(8,2,2);
for i = 1:8
    Is_point(i,:,:) = [IsO(i,i),IsO(i,i+8);IsO(i+8,i),IsO(i+8,i+8)];
    Is_point1 = inv([IsO(i,i),IsO(i,i+8);IsO(i+8,i),IsO(i+8,i+8)]);

    [V,D] = eig(Is_point1);
    a = sqrt(5.991/D(1));
    b = sqrt(5.991/D(4));
    t = linspace(0,2*pi,1e2);

    XY = V*[a*cos(t);b*sin(t)];
    X = XY(1,:)+ax(i+4,1);
    Y = XY(2,:)+ax(i+4,2);
    plot(X,Y,'LineWidth',1.5);
end
for i = 1:12
    if (i <= 4)
        plot(ax(i,1),ax(i,2),'b*','MarkerSize',10);
    else
        for j = 1:N
            plot(ax_est_new_1(j,i,1),ax_est_new_1(j,i,2),'r*','MarkerSize',1);
        end
    end
end
xlabel('/m');
ylabel('/m');
title('单点定位','FontSize',16);
xlim([0,40]);
ylim([0 50]);

Im_xx = zeros(8,8);
Im_xy = zeros(8,8);
Im_yy = zeros(8,8);
for i = 1:8
    for j = 1:8
        if (i == j)
            for k = 1:12
                if (k ~= i+4)
                    Im_xx(i,i) = Im_xx(i,i) + (ax(i+4,1)-ax(k,1))*(ax(i+4,1)-ax(k,1)) / ((ax(i+4,1)-ax(k,1))^2+(ax(i+4,2)-ax(k,2))^2) / c^2 / noise_var^2;
                    Im_xy(i,i) = Im_xy(i,i) + (ax(i+4,1)-ax(k,1))*(ax(i+4,2)-ax(k,2)) / ((ax(i+4,1)-ax(k,1))^2+(ax(i+4,2)-ax(k,2))^2) / c^2 / noise_var^2;
                    Im_yy(i,i) = Im_yy(i,i) + (ax(i+4,2)-ax(k,2))*(ax(i+4,2)-ax(k,2)) / ((ax(i+4,1)-ax(k,1))^2+(ax(i+4,2)-ax(k,2))^2) / c^2 / noise_var^2;
                end
            end
        else
            Im_xx(i,j) = (ax(i,1)-ax(j,1))*(ax(i,1)-ax(j,1)) / ((ax(i,1)-ax(j,1))^2+(ax(i,2)-ax(j,2))^2) / c^2 / noise_var^2;
            Im_xy(i,j) = (ax(i,1)-ax(j,1))*(ax(i,2)-ax(j,2)) / ((ax(i,1)-ax(j,1))^2+(ax(i,2)-ax(j,2))^2) / c^2 / noise_var^2;
```

```
            Im_yy(i,j) = (ax(i,2)-ax(j,2))*(ax(i,2)-ax(j,2)) / ((ax(i,1)-ax(j,1))^2+(ax(i,2)-ax(j,2))^2) / c^2 / noise_var^2;
        end
    end
end
Im = [Im_xx,Im_xy;Im_xy',Im_yy];
ImO = inv(Im);

figure,hold on;
Im_point = zeros(8,2,2);
for i = 1:8
    Im_point(i,:,:) = [ImO(i,i),ImO(i,i+8);ImO(i+8,i),ImO(i+8,i+8)];
    Im_point1 = inv([ImO(i,i),ImO(i,i+8);ImO(i+8,i),ImO(i+8,i+8)]);

    [V,D] = eig(Im_point1);
    a = sqrt(5.991/D(1));
    b = sqrt(5.991/D(4));
    t = linspace(0,2*pi,1e2);

    XY = V*[a*cos(t);b*sin(t)];
    X = XY(1,:)+ax(i+4,1);
    Y = XY(2,:)+ax(i+4,2);
    plot(X,Y,'LineWidth',1.5);
end
for i = 1:12
    if (i <= 4)
        plot(ax(i,1),ax(i,2),'b*','MarkerSize',10);
    else
        for j = 1:N
            plot(ax_est_new_2(j,i,1),ax_est_new_2(j,i,2),'r*','MarkerSize',1);
        end
    end
end
xlabel('/m');
ylabel('/m');
title('协作定位','FontSize',16);
xlim([0,40]);
ylim([0 50]);
```

梯度下降函数

```
function [ax_est_new,ite_num] = GradientDescent(TOA,ax,bais_max,TOA_mode,alpha,i_thr,ut,sigma)
c = 3e8;
if (TOA_mode == 1)
    TOA_ite = 4;
elseif (TOA_mode == 2)
    TOA_ite = 12;
end

ax_est_old = zeros(12,2);
ax_est_old(1:4,:) = ax(1:4,:);
ax_bias = bais_max*(1-2*rand([12,2]));
ax_est_new = ax - ax_bias;
ax_est_new(1:4,:) = ax(1:4,:);

ite_num = 0;
while sum(sum(abs(ax_est_new-ax_est_old))) >= i_thr
    ax_est_old = ax_est_new;

    for i = 5:12
        grad_x = 0;
        grad_y = 0;
        for j = 1:TOA_ite
            if (j ~= i)
                grad_x = grad_x - (TOA(i,j) - sqrt((ax_est_old(i,1)-ax_est_old(j,1))^2+(ax_est_old(i,2)-ax_est_old(j,2))^2)/c + ut(j)) * (ax_est_old(i,1)-ax_est_old(j,1))/sqrt((ax_est_old(i,1)-ax_est_old(j,1))^2+(ax_est_old(i,2)-ax_est_old(j,2))^2)/c/sigma^2;
                grad_y = grad_y - (TOA(i,j) - sqrt((ax_est_old(i,1)-ax_est_old(j,1))^2+(ax_est_old(i,2)-ax_est_old(j,2))^2)/c + ut(j)) * (ax_est_old(i,2)-ax_est_old(j,2))/sqrt((ax_est_old(i,1)-ax_est_old(j,1))^2+(ax_est_old(i,2)-ax_est_old(j,2))^2)/c/sigma^2;
            end
        end
```

```
28
29          ax_est_new(i,1) = ax_est_old(i,1) - alpha*grad_x;
30          ax_est_new(i,2) = ax_est_old(i,2) - alpha*grad_y;
31          if(ite_num >= 1e4)
32              break;
33          end
34      end
35      ite_num = ite_num + 1;
36  end
37  end
```