第二次课程设计——TDOA 定位法（双曲定位）

黄琅飞　　　201921220305

一、原理分析

假设第 i 个锚节点位置为 (x_i,y_i)，未知节点位置为 (x_0,y_0)，r_i为未知节点到第 i 个锚节点的距离，r_{i1}为未知节点到第 i 个锚节点与第 1 个锚节点的距离差，t_i为未知节点到第 i 个锚节点的时间。

$$r_{i1} = r_i - r_1 = c(t_i - t_1) = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} - \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

$$r_i^{\ 2} = (x_0 - x_i)^2 + (y_0 - y_i)^2 = k_i - 2x_i x_0 - 2y_i y_0 + x_0^{\ 2} + y_0^{\ 2} \qquad k_i = x_i^{\ 2} + y_i^{\ 2}$$

由 TDOA 定义可知

$$r_i^{\ 2} = (r_{i1} + r_1)^2$$

展开

$$r_{i1}^{\ 2} + 2r_{i1}r_1 + r_1^{\ 2} = k_i - 2x_i x_0 - 2y_i y_0 + x_0^{\ 2} + y_0^{\ 2}$$

又

$$r_1^{\ 2} = k_1 - 2x_1 x_0 - 2y_1 y_0 + x_0^{\ 2} + y_0^{\ 2}$$

两式相减，消除 r_1^2

$$r_{i1}^{\ 2} + 2r_{i1}r_1 = k_i - 2x_{i1}x_0 - 2y_{i1}y_0 - k_1$$

用矩阵形式表达

$$\mathbf{GZ} = \mathbf{Y}$$

$$\mathbf{G} = \begin{bmatrix} 2x_{21} & 2y_{21} & 2r_{21} \\ \vdots & \vdots & \vdots \\ 2x_{N1} & 2y_{N1} & 2r_{N1} \end{bmatrix} \mathbf{Z} = \begin{bmatrix} x_0 \\ y_0 \\ r_1 \end{bmatrix} \mathbf{Y} = \begin{bmatrix} k_2 - k_1 - r_{21}^2 \\ \vdots \\ k_N - k_1 - r_{N1}^2 \end{bmatrix}$$

然后可以用二步加权最小二乘法求解(x_0,y_0)，具体如下：

第一步

$$\mathbf{Z} = (\mathbf{G}^T \operatorname{cov}(\mathbf{e})^{-1}\mathbf{G})^{-1}\mathbf{G}^T \operatorname{cov}(\mathbf{e})^{-1}\mathbf{Y}$$

$$\mathbf{e}' = \mathbf{Y}' - \mathbf{G}'\mathbf{Z}'$$

$$\operatorname{cov}(e) = E[\Delta Y \Delta Y^T]$$

$\operatorname{cov}(e)(i,j)=E[4r_{(i+1)1}r_{(j+1)1}(n_{i+1}n_{j+1}-n_{i+1}n_1-n_{j+1}n_1+n_1n_1)]$，$n_i$为未知节点到锚节点的高斯白噪声，所以 $\operatorname{cov}(e)$的对角线元素为 $8r_{(i+1)1}r_{(i+1)1}*\sigma^2$，非对角线元素为 $4r_{(i+1)1}r_{(j+1)1}*\delta^2$。

第二步，为消除 Z 各元素间相关性带来的误差，设第一个锚节点的坐标为(0,0)，则 $r_1^2=x^2+y^2$，令 $Z_1=x+\delta_1$，$Z_2=y+\delta_2$，$Z_3=r_1^2+\delta_3$，δ_1~δ_3为 Z 的估计误差，则误差矩阵可表示为

$$\mathbf{e}' = \mathbf{Y}' - \mathbf{G}'\mathbf{Z}'$$

$$\mathbf{Y'}=\begin{bmatrix}Z_1^2\\ Z_2^2\\ Z_3\end{bmatrix}\quad \mathbf{G'}=\begin{bmatrix}1 & 0\\ 0 & 1\\ 1 & 1\end{bmatrix}\quad \mathbf{Z'}=\begin{bmatrix}Z'_1\\ Z'_2\end{bmatrix}=\begin{bmatrix}x^2\\ y^2\end{bmatrix}\quad \mathbf{e'}=\begin{bmatrix}e'_1\\ e'_2\\ e'_3\end{bmatrix}$$

采用泰勒展开，并忽略二次项

$$\begin{aligned}e'_1&=2x\delta_1+\delta_1^2\approx 2x\delta_1\\ e'_2&=2y\delta_2+\delta_2^2\approx 2y\delta_2\\ e'_3&=\delta_3\end{aligned}$$

$$\operatorname{cov}(\mathbf{e'})=E(\mathbf{e'}\mathbf{e'}^T)=\mathbf{D}\{\operatorname{cov}(\mathbf{Z})\}\mathbf{D}$$

其中

$$\mathbf{D}=diag\left\{\begin{bmatrix}2x & 2y & 1\end{bmatrix}\right\}$$

则

$$\begin{aligned}\mathbf{Z'}&=\arg\min\{(\mathbf{Y'}-\mathbf{G'Z'})^T\operatorname{cov}(\mathbf{e'})^{-1}(\mathbf{Y'}-\mathbf{G'Z'})\}\\ &=(\mathbf{G'}^T\operatorname{cov}(\mathbf{e'})^{-1}\mathbf{G'})^{-1}\mathbf{G'}^T\operatorname{cov}(\mathbf{e'})^{-1}\mathbf{Y'}\end{aligned}$$

$$\mathbf{D}=diag\left\{\begin{bmatrix}2x & 2y & 1\end{bmatrix}\right\}\Rightarrow\mathbf{D}\approx diag\left\{\begin{bmatrix}2Z_1 & 2Z_2 & 1\end{bmatrix}\right\}$$

最后得到目标的位置

$$\mathbf{Zp}=\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}sign(Z_1)\sqrt{Z'_1}\\ sign(Z_2)\sqrt{Z'_2}\end{bmatrix}$$

二、仿真步骤

1、推导基于二步加权最小二乘的 TDOA 定位算法

2、推导 TDOA 的 GDOP

3、完成如下仿真

（1）采用匹配滤波方法完成 TDOA 的估计（产生基带信号的程序已经给出）;

（2）利用估计好的 TDOA，完成不同信噪比情况下(SNR=5:5:25dB)的 LS 和二步加权最小二乘算法的定位精度

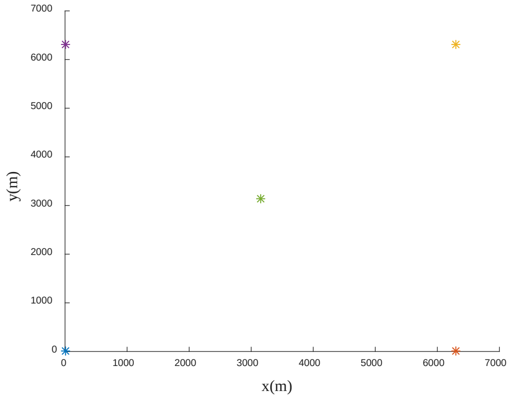
（3）计算各个位置的 GDOP，并采用 matlab 函数 contour 画出来 GDOP

（4）仿真参数如下：

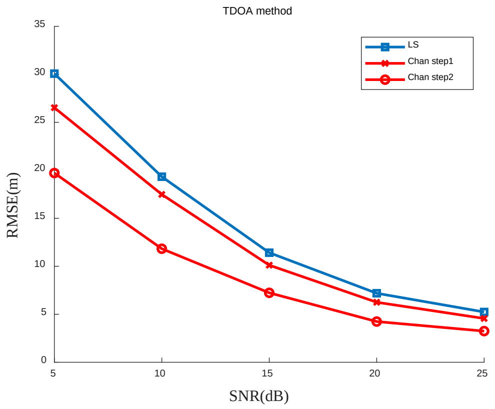
基带采样频率为 fs=30Mhz，采样个数为 Nsample=8192*4，未知节点均匀分布在 x 轴[100,6200]m, y 轴[100,6200]m 的区间内，5 个锚节点位置为：(0,0)、(0,6300)、(6300,6300)、(6300,0)、(3150,3150)。

三、仿真结果及分析

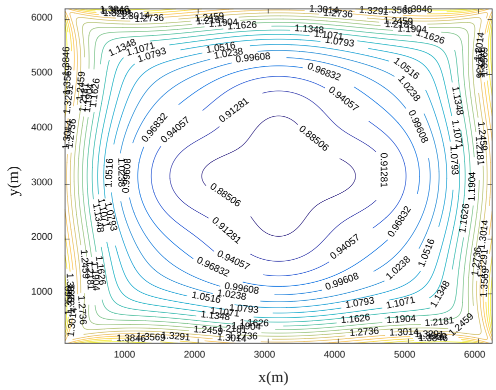
锚节点位置：



TDOA 定位法的均方根误差：



TDOA 定位法的 GDOP：



从结果可以看出随着信噪比的增大，TDOA 的定位误差逐渐减小，并逐渐趋于稳定；而且二步加权后的定位精度也高于仅一步加权后的精度。

四、Matlab 仿真程序

```
clear,clc,close all
wavespeed=3e8;
fs=30e6;
Nsample=8196;
%节点分布区域
xscale=6.3e3;
yscale=6.3e3;
%锚节点位置
anchor_node_position=[0 0;xscale 0;xscale yscale;0 yscale;xscale/2 yscale/2];
%画出锚节点布局
figure
hold on
for i=1:length(anchor_node_position)
    plot(anchor_node_position(i,1),anchor_node_position(i,2),'*');
end
xlabel('x(m)','FontSize',16,'FontName','Times New Roman')
ylabel('y(m)','FontSize',16,'FontName','Times New Roman')
set(gcf,'Color',[1,1,1])
```

```
%发射信号
s_t=Generate_tdoa_base_band_signal(0,100,fs,Nsample);
s_t_fft=fft(s_t);
%共进行 runnum 次仿真，并对其进行统计
runnum=50;
SNR=0:5:25;
for k=1:length(SNR)
    SNR(k)
    for i=1:runnum
%           SNR(k),i
        %随机生成盲节点位置
        blind_node_position(1)=(xscale-20)*rand(1,1)+10;
        blind_node_position(2)=(yscale-20)*rand(1,1)+10;
        %生成测距信息
        for j=1:length(anchor_node_position)
            %真实距离值                d(j)=sqrt((blind_node_position(1)-
anchor_node_position(j,1))^2+(blind_node_position(2)-anchor_node_position(j,2))^2);
            s_r(j,:)=Generate_tdoa_base_band_signal(d(j)/wavespeed,SNR(k),fs,Nsample);
            s_r_fft(j,:)=fft(s_r(j,:));
        end
        %匹配滤波估计距离信息
        for j=1:length(anchor_node_position)-1
            %估计距离差
               h=fftshift(ifft(s_r_fft(j+1,:).*conj(s_r_fft(1,:))));
            [maxvalue maxindex]=max(abs(h));
%               figure
%               plot(abs(h))
            tdoa_es=(maxindex-Nsample/2-1)/fs;
            r_es(j)=tdoa_es*wavespeed;
        end
        %基于 LS 的 TOA 方法对盲节点进行定位
        [Z,Z2]=TDOA_LOCATION_Chan_Method(anchor_node_position,r_es,1);
        %统计每一次定位误差
        chan_tdoa_err1(i)=sqrt((Z(1)-blind_node_position(1))^2+(Z(2)-blind_node_position(2))^2);
        chan_tdoa_err2(i)=sqrt((Z2(1)-blind_node_position(1))^2+(Z2(2)-blind_node_position(2))^2);
    end
    %均方根定位误差
    chan_tdoa_err1_rmse(k)=sqrt(mean(chan_tdoa_err1.^2));
    chan_tdoa_err2_rmse(k)=sqrt(mean(chan_tdoa_err2.^2));
end
figure
hold on
plot(SNR,chan_tdoa_err1_rmse,'-rx','LineWidth',2)
plot(SNR,chan_tdoa_err2_rmse,'-ro','LineWidth',2)
```

```matlab
%legend('LS','Chan step1','Chan step2')
title('TDOA method')
xlabel('SNR(dB)','FontSize',16,'FontName','Times New Roman')
ylabel('RMSE(m)','FontSize',16,'FontName','Times New Roman')
set(gcf,'Color',[1,1,1])
legend('Chan step1','Chan step2')
X=100:13:xscale-100;
Y=100:13:yscale-100;
for i=1:length(X)
    for j=1:length(Y)
        [gdop(i,j)]=TOA_GDOP(anchor_node_position,[X(i) Y(j)]);
    end
end
figure
[C,h] =contour(X,Y,gdop,20);
set(h,'ShowText','on')
xlabel('x(m)','FontSize',16,'FontName','Times New Roman')
ylabel('y(m)','FontSize',16,'FontName','Times New Roman')
set(gcf,'Color',[1,1,1])

function [s_r]=Generate_tdoa_base_band_signal(t_delay,SNR,fs,Nsample)
B=1e6;
Tc=Nsample/fs;
S=B/Tc;
t=0:1/fs:Tc-1/fs;
t=t-Tc/2;
fc=100e6;
s_r=exp(1i*pi*S.*(t-t_delay).^2)*exp(-1i*2*pi*fc*t_delay);
s_r=awgn(s_r,SNR);

function [Z,Z2]=TOA_LOCATION_Chan_Method(bs,r,delta_var)
for i=1:length(bs)-1
    G(i,:)=[2*(bs(1,1)-bs(i+1,1)) 2*(bs(1,2)-bs(i+1,2)) 2*r(i)];
    H(i,1)=bs(i+1,1)^2+bs(i+1,2)^2-r(i)^2;
    B(i,i)=8*(r(i)^2)*delta_var;
    for j=1:length(bs)-1
        if j~=i
            B(i,j)=4*r(i)*r(j)*delta_var;
        end
    end
end
Z=pinv(G.'*pinv(B)*G)*G.'*pinv(B)*H;
covZ=pinv(G.'*pinv(B)*G);
%第二步加权
```

```
G2=[1 0;0 1;1 1];
H2=[Z(1)^2 Z(2)^2 Z(3)^2].';
D=diag([2*Z(1) 2*Z(2) 2*Z(3)]);
cove2=D*covZ*D;
Z2=pinv(G2.'*pinv(cove2)*G2)*G2.'*pinv(cove2)*H2;
Z2=[sign(Z(1))*sqrt(abs(Z2(1))) sign(Z(2))*sqrt(abs(Z2(2)))].';

function [gdop]=TDOA_GDOP(anchor_node_position,blind_node_position)
Nanchor=length(anchor_node_position);
for i=1:length(anchor_node_position)     d(i)=sqrt((blind_node_position(1)-
anchor_node_position(i,1))^2+(blind_node_position(2)-anchor_node_position(i,2))^2);
end
for i=1:Nanchor-1     H(1,i)=[(blind_node_position(1)-anchor_node_position(i+1,1))/d(i+1)-
(blind_node_position(1)-anchor_node_position(1,1))/d(1)];   H(2,i)=[(blind_node_position(2)-
anchor_node_position(i+1,2))/d(i+1)-(blind_node_position(2)-anchor_node_position(1,2))/d(1)];
end
q=eye(Nanchor-1)+ones(Nanchor-1);
gdop=trace(pinv(H*pinv(q)*H.')
```