Report of “A Simple and Efficient Estimator for Hyperbolic Location

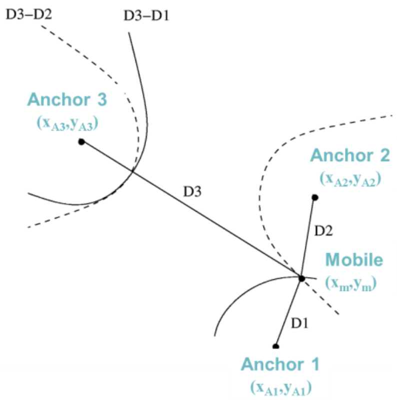
16308073 刘渤　1 班

Abstract--The paper introduce an efficient technique in locating a source based on intersections of hyperbolic curves defined by the time differences, which we call TDOA, of arrival of a signal received at a number of sensors is proposed. The solutions it mentions include the Cramer-Rao lower bound (CRLB), and the Taylor-series, the proposed technique which is seen to perform significantly better than SI (mention in other paper), and has a higher noise threshold than DAC before performance breaks away from the CRLB. It also provides an explicit solution form that is not available in the Taylor-series methods. Computational complexity is comparable to SI but less than Taylor-series.

I. INTRODUCTION

- TDOA

  To explain the TDOA, we can see the basic measurements made by Loran-C receivers to determine the difference in the time-of-arrival (TD) between the master signal and the signals from each of the secondary stations of a chain. The principle of time difference measurements in hyperbolic mode is illustrated in the adjacent figure. In general, you could say that when the Master signal is received, it is the "Start" of the Stopwatch. When a secondary station is received it is the "Stop" for one TD.



$$d_{32} = \sqrt{\left(x_{A_3}-x_M\right)^2+\left(y_{A_3}-y_M\right)^2}-\sqrt{\left(x_{A_2}-x_M\right)^2+\left(y_{A_2}-y_M\right)^2}$$
$$d_{31} = \sqrt{\left(x_{A_3}-x_M\right)^2+\left(y_{A_3}-y_M\right)^2}-\sqrt{\left(x_{A_1}-x_M\right)^2+\left(y_{A_1}-y_M\right)^2}$$

- In the paper, we use a 2-D plane for ease of illustration. And the paper gives us several locations of some sensors. We can sort them into two kinds: the arbitrary and the linear. And we know that the linear array is just included in the arbitrary array but its solution will be different for the specialty while it may cause some problems using normal solutions, such as the singular value. But we can see the special way has a simple algorithm.
- Besides the sensors positions, we should consider the sources while it will be distant or near and as we know, the distant sources will have more noise and its error may be strengthened so we should give the different deal which I will mention next.
- So by now, we can give a conclusion that the paper put forward some solutions

based on TDOA to solve the source positions and compares them by estimating their MSE. It tells us the MSE of SI (not mention in the paper), Taylor-series and proposed methods in the arbitrary array and near source, the MSE of Taylor-series and proposed methods in the linear array and near source, and their distant situation. And simulate them in the data sheets.

- I will realize the Taylor-series, CRLB and 2 proposed ways in the simulation. With the simulation, the paper also gives us a comparison of the threshold effect between DAC and our method. I will draw the Taylor, 2 proposed ways and the CRLB which are more realistic for us to observe the thresholds of them.

Ⅱ.SOLUTION

- Taylor-series

Based on the TDOA:

$$r_i^2 = (x_i - x)^2 + (y_i - y)^2 = K_i - 2x_i x - 2y_i y + x^2 + y^2,\ i = 1, 2, \cdots, M \qquad r_{i,1} = cd_{i,1} = r_i - r_1$$

We give the matrix ht and Gt and the deviation is:

$$\mathbf{h}_t = \begin{bmatrix} r_{2,1} - (r_2 - r_1) \\ r_{3,1} - (r_3 - r_1) \\ \\ r_{M,1} - (r_M - r_1) \end{bmatrix}$$

$$\mathbf{G}_t = \begin{bmatrix} (x_1 - x)/r_1 - (x_2 - x)/r_2 & (y_1 - y)/r_1 - (y_2 - y)/r_2 \\ (x_1 - x)/r_1 - (x_3 - x)/r_3 & (y_1 - y)/r_1 - (y_3 - y)/r_3 \\ \\ (x_1 - x)/r_1 - (x_M - x)/r_M & (y_1 - y)/r_1 - (y_M - y)/r_M \end{bmatrix} \qquad \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \left(\mathbf{G}_t^T \mathbf{Q}^{-1} \mathbf{G}_t\right)^{-1} \mathbf{G}_t^T \mathbf{Q}^{-1} \mathbf{h}_t$$

While Taylor-series use the deviation to modify the sources and continually iterate the process to the position where deviation is small enough and we will give the x and y location. We can see the partial code below.

```
Preudo_Code:
Q=sigma_d_square.*T;          %the Q matrix has been solved
R(ii)=sqrt((x(ii)-x_).^2+(y(ii)-y_).^2);     %solve the ri
r0_i1(ii)=R(ii)-R(1);       %generate the r0_i1
noise=3*10.^8*normrnd(0,sqrt(sigma_d_square./2),1,M);
r_i1=r0_i1+noise;                          %make the noise and r_i1
while(norm(deta)>exp(-15)&&nn<10000)
  %when the norm is small enough or the times is large enough
ht(ii,1)=r_i1(ii+1)-(r(ii+1)-r(1));    %solve the ht
Gt(ii,1)=(x(1)-x_)./r(1)-(x(ii+1)-x_)./r(ii+1);
Gt(ii,2)=(y(1)-y_)./r(1)-(y(ii+1)-y_)./r(ii+1);    %get the Gt
deta=(((Gt'*(Q^(-1))*Gt)^(-1))*Gt'*(Q^(-1))*ht);    %solve the deta
x_=x_+deta(1);
y_=y_+deta(2);       %iterate the x, y
EE=(x_-x0)^2+(y_-y0)^2+EE;     %to calculate the MSE
```

- Proposed method

We first give the Za' this is what we want to solve the Zp

$$\mathbf{z}_a' \approx (\mathbf{G}_a'^T\mathbf{B}'^{-1}\mathbf{G}_a\mathbf{Q}^{-1}\mathbf{G}_a\mathbf{B}'^{-1}\mathbf{G}_a')^{-1}(\mathbf{G}_a'^T\mathbf{B}'^{-1}\mathbf{G}_a\mathbf{Q}^{-1}\mathbf{G}_a\mathbf{B}'^{-1})\mathbf{h}'.$$

Where we get the Ga', B', Ga, h' Za below

$$\mathbf{h}' = \begin{bmatrix}(z_{a,1}-x_1)^2\\(z_{a,2}-y_1)^2\\z_{a,3}^2\end{bmatrix},\qquad \mathbf{G}_a' = \begin{bmatrix}1&0\\0&1\\1&1\end{bmatrix}\qquad \mathbf{h} = \frac{1}{2}\begin{bmatrix}r_{2,1}^2-K_2+K_1\\r_{3,1}^2-K_3+K_1\\ \cdot \\ r_{M,1}^2-K_M+K_1\end{bmatrix}$$

$$\mathbf{z}_a' = \begin{bmatrix}(x-x_1)^2\\(y-y_1)^2\end{bmatrix}\qquad \mathbf{G}_a = -\begin{bmatrix}x_{2,1}&y_{2,1}&r_{2,1}\\x_{3,1}&y_{3,1}&r_{3,1}\\ \cdot&\cdot&\cdot\\x_{M,1}&y_{M,1}&r_{M,1}\end{bmatrix}$$

$$\mathbf{z}_a \approx (\mathbf{G}_a^T\mathbf{Q}^{-1}\mathbf{G}_a)^{-1}\mathbf{G}_a^T\mathbf{Q}^{-1}\mathbf{h}.$$

$$\mathbf{B}' = \mathrm{diag}\{x^0-x_1, y^0-y_1, r_1^0\}.$$

So now we can get the Zp which is a vector consists of the x, y by the formula

$$\mathbf{z}_p = \sqrt{\mathbf{z}_a'} + \begin{bmatrix}x_1\\y_1\end{bmatrix}$$

All the parameters are given so we can deal with the position. We can see the partial code below. The solution have a similar thought with other ways, that is to iterate the x and y in the process of the algorithm when we have a new signal data to strengthen the real source position to modify the result so that we can get a more accurate results. For example in the proposed method we can see when the Za is get, we will replace it to the x and y in the next calculation such as the B'. We can have a look of the codes:

```
Preudo_Code:
h(ii,1)=(1./2).*((r_i1(ii+1)).^2-K(ii+1)+K(1));    %make the h
Ga(ii,1)=-(x(ii+1)-x(1));
Ga(ii,2)=-(y(ii+1)-y(1));
Ga(ii,3)=-(r_i1(ii+1));                    %make the Ga
Za=((Ga'*(Q^-1)*Ga)^-1)*Ga'*(Q^-1)*h;    %make the Za
x1=Za(1);
y1=Za(2);                      %iterate the x and y
Ga_=[1,0;0,1;1,1];             %make the Ga_
B_=diag([x1-x(1),y1-y(1),Za(3)]);    %make the B_
h_=[(Za(1)-x(1))^2;(Za(2)-y(1)).^2;(Za(3)).^2];    %the h_
Za_=((Ga_'*(B_^-1)*Ga'*(Q^-1)*Ga*(B_^-1)*Ga_)^-1)*(Ga_'*(B_^-1)*Ga'
    *(Q^-1)*Ga*(B_^-1))*h_;          %make the Za_
Zp=sqrt(Za_)+[x(1);y(1)];      %make the Zp
EE=(Zp(1)-x0).^2+(Zp(2)-y0).^2+EE;
```

- Proposed method_2

  In this way, we can see some same way we use in the proposed way because it is calculated with the interate of the former method.

  We know the purpose for us to solve the position by this way is to get the Zp like the former one, but what differs is the Za', and Ga, so we can first get the Za' and its parameters below. While the Ga', h' and B' have been given.

$$\mathbf{z}_a' = (\mathbf{G}_a'^T\Psi'^{-1}\mathbf{G}_a')^{-1}\mathbf{G}_a'^T\Psi'^{-1}\mathbf{h}'.\qquad \Psi' = [\psi'\psi'^T] = 4\mathbf{B}'\mathrm{cov}(\mathbf{z}_a)\mathbf{B}'$$

$$\mathbf{cov}(\mathbf{z}_a) = E[\Delta\mathbf{z}_a \Delta\mathbf{z}_a^T] = (\mathbf{G}_a^{0T}\Psi^{-1}\mathbf{G}_a^0)^{-1}$$

$$\begin{aligned}\mathbf{z}_a &= \arg\ \min\{(\mathbf{h} - \mathbf{G}_a\mathbf{z}_a)^T\Psi^{-1}(\mathbf{h} - \mathbf{G}_a\mathbf{z}_a)\}\\ &= (\mathbf{G}_a^T\Psi^{-1}\mathbf{G}_a)^{-1}\mathbf{G}_a^T\Psi^{-1}\mathbf{h}\end{aligned}$$

$$\Psi = \mathbf{E}[\psi\psi^T] = c^2\mathbf{BQB}.$$

$$\mathbf{B} = \mathbf{diag}\{r_2^0, r_3^0, \cdots, r_M^0\}.$$

By the formula we set we can get the Zp so that we can have the position, too. What we should focus on is that I explain the solution in an inverted sequence so when you want to use the solution you should write the algorithm in an inverted direction. And we can analysis the algorithm of the proposed_2 that it use the iteration based on the proposed_1, and when we get the new x and y we should replace them to the old ones. In the solution we know when Za is generated we will get a new ones. In the process we have two iterations to the x and y, so it will get a more satisfying result. We can see the partial codes below.

```
Preudo_Code:
Za=((Ga'*(Q^(-1))*Ga)^(-1))*Ga'*(Q^(-1))*h;    %make the Za
x1=Za(1);
y1=Za(2);      %iterate the x and y
r0(i-1)=sqrt((x(1)-x1)^2+(y(1)-y1)^2);
B(i-1,i-1)=r0(i-1);
Y=9.0*10.^16.*B*Q*B;      %Y
Zaa=((Ga'*(Y^-1)*Ga)^-1)*Ga'*(Y^-1)*h;
x1=Zaa(1);
y1=Zaa(2);        %iterate the x and y again
Ga_=[1,0;0,1;1,1];              %make the Ga_
B_=diag([x1-x(1),y1-y(1),Za(3)]);    %make the B_
h_=[(Zaa(1)-x(1))^2;(Zaa(2)-y(1)).^2;(Zaa(3)).^2];    %h_
cov_Za=(Ga'*(Y^(-1))*Ga)^(-1);        %conv_Za
Y_=4.*B_*cov_Za*B_;                           % Y_
Za_=((Ga_'*(Y_^(-1))*Ga_)^(-1))*Ga_'*(Y_^(-1))*h_; %Za_
Zp=sqrt(Za_)+[x(1);y(1)];      %make the Zp
EE=(Zp(1)-x0).^2+(Zp(2)-y0).^2+EE;
```

- CRLB

  The solution has a more simple process while we will have the theory result.

$$\begin{aligned}\Phi = \mathrm{cov}(\mathbf{z}_p) &= \frac{1}{4}\mathbf{B}''^{-1}\mathrm{cov}(\mathbf{z}'_a)\mathbf{B}''^{-1}\\ &= c^2\mathbf{B}''\mathbf{G}_a'^T\mathbf{B}'^{-1}\mathbf{G}_a^{0T}\mathbf{B}^{-1}\mathbf{Q}^{-1}\mathbf{B}^{-1}\mathbf{G}_a^0\mathbf{B}'^{-1}\mathbf{G}'_a\mathbf{B}'')^{-1}\end{aligned}$$

$$\mathbf{B}'' = \begin{bmatrix}(x^0 - x_1) & 0\\ 0 & (y^0 - y_1)\end{bmatrix}.$$

And the result we want is to add all the diagonal line elements. We can see the code below.

```
Preudo_Code:
B(ii,ii)=R(ii+1);      %the B
```

B__=[(x0-x(1)), 0;0,(y0-y(1))];   %the B__
B_=diag([x0-x(1),y0-y(1),R(1)]);   %the B_
YY=(3.0*10^8)^2.*((B__*Ga_'*(B_^-1)*Ga'*(B^-1)*(Q^-1)*(B^-1)*Ga*(B_^-1)*Ga_*B__)^-1);

- Three sensors

  When we encounter some situation that the sensors are in number of 3, we will have other solution to modify the x and y. We should use the formula below to solve the problems.

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\begin{bmatrix} x_{2,1} & y_{2,1} \\ x_{3,1} & y_{3,1} \end{bmatrix}^{-1} \times \left\{ \begin{bmatrix} r_{2,1} \\ r_{3,1} \end{bmatrix} r_1 + \frac{1}{2} \begin{bmatrix} r_{2,1}^2 - K_2 + K_1 \\ r_{3,1}^2 - K_3 + K_1 \end{bmatrix} \right\}.$$

  Also when we encounter the three linear sensors, we will other solution to do it.

$$r = \frac{L_1\left[1-(\frac{r_{2,1}}{L_1})^2\right] + L_2\left[1-(\frac{r_{3,1}}{L_2})^2\right]}{2(\frac{r_{3,1}}{L_2} + \frac{r_{2,1}}{L_1})} \quad x = \frac{r_{2,1}L_2^2 - r_{3,1}L_1^2 - r_{2,1}r_{3,1}r_{3,2}}{2\{r_{2,1}L_2 + r_{3,1}L_1\}}$$

y is obtained from $\sqrt{r^2 - x^2}$.

- Q matrix and sigma

  The paper gives us a solution of Q below

$$\mathbf{Q} = \left\{ \frac{2T}{2\pi} \int_0^{\Omega} \omega^2 \frac{S(\omega)^2}{1 + S(\omega)\mathrm{tr}(\mathbf{N}(\omega)^{-1})} \times \left[\mathrm{tr}\left(\mathbf{N}(\omega)^{-1}\right)\mathbf{N}_p(\omega)^{-1} - \mathbf{N}_p(\omega)^{-1}\mathbf{1}\mathbf{1}^T\mathbf{N}_P(\omega)^{-1}\right] d\omega \right\}^{-1}$$

  Actually we can use the sigma to get it. Q=sigma^2*M; the M matrix can be described as $\begin{bmatrix} 1 & 0.5 & 0.5 & \cdots & 0.5 \\ 0.5 & 1 & 0.5 & \cdots & 0.5 \\ . & . & . & \cdots & . \\ 0.5 & 0.5 & 0.5 & \cdots & 1 \end{bmatrix}$, and the sigma can be adjust with the situation changing to arbitrary, linear or near and distant. It can be d./(c^2), while the d can be 0.001, 0.0001,0.00001 when it is arbitrary, linear and distant.

Ⅲ.SIMULATION

- Arbitrary Array

  While the paper gives the arbitrary array which is consist of 10 group data and the source data is x=8, y=10, so we can see the MSE the paper simulates is in the chart below.

TABLE I
COMPARISON OF MSE FOR THE SI, TAYLOR-SERIES AND PROPOSED METHODS; ARBITRARY ARRAY AND NEAR SOURCE

MSE	M = 3	M = 4	M = 5	M = 6	M = 7	M = 8	M = 9	M = 10
A	no sol.	1.5768	0.1597	0.1480	0.1229	0.1164	0.1148	0.1103
B	2.1726	0.7061	0.1460	0.1341	0.1144	0.1057	0.1034	0.09462
C	2.1726	0.7282	0.1456	0.1359	0.1159	0.1077	0.1055	0.09697
D	2.1726	0.6986	0.1451	0.1337	0.1141	0.1055	0.1034	0.09480
E	1.9794	0.6884	0.1451	0.1334	0.1143	0.1054	0.1032	0.09432

A: SI method. B: Taylor series method. C: proposed method, { (14*b*), (22*b*), (24) }. D: proposed method, { (14*b*), (14*a*), (22*a*), (24) }. E: theoretical MSE of the new method = CRLB.

We can see in the chart that the paper compare the MSE of SI (no mention in the paper), and Taylor-series, proposed_1, proposed_2 and CRLB. So I use the Taylor, proposed and CRLB to simulate the MSE, the result is similar.

Table 1

MSE	M=3	M=4	M=5	M=6	M=7	M=8	M=9	M=10
B	2.51247	0.67408	0.12023	0.10776	0.09558	0.09580	0.09507	0.08809
C	2.55841	0.74229	0.11825	0.10341	0.09768	0.09671	0.09882	0.08813
D	2.53063	0.71126	0.11648	0.10457	0.09740	0.09848	0.09625	0.08798
E	1.97937	0.68835	0.14508	0.13343	0.11427	0.10538	0.10322	0.09432

- Linear Array

  The difference of the array is that the paper give the linear array so it gets the MSE, and the source data is still x=8, y=10.

TABLE II
COMPARISON OF MSE FOR THE PROPOSED AND TAYLOR-SERIES METHODS; LINEAR ARRAY AND NEAR SOURCE

MSE	M = 3	M = 4	M = 5	M = 6	M = 7	M = 8	M = 9	M = 10
B	8.2574	1.2836	0.3566	0.1222	0.06155	0.02854	0.01746	0.009540
C	8.2574	1.1131	0.3556	0.1225	0.06199	0.02884	0.01772	0.009726
D	8.2574	1.1117	0.3545	0.1219	0.06148	0.02852	0.01746	0.009541
E	7.2819	1.1000	0.3548	0.1219	0.06123	0.02840	0.01750	0.009599

B: Taylor series method. C: proposed method, {(28) with $\Psi = Q$, (29)}. D: proposed method, {(28) with $\Psi = Q$, (28), (29)}. E: theoretical MSE of the new method = CRLB.

And my result is similar, too.

Table 2

MSE	M=3	M=4	M=5	M=6	M=7	M=8	M=9	M=10
B	2.49104	0.86361	0.24959	0.10094	0.04916	0.02498	0.01488	0.00865
C	2.50821	0.82232	0.25265	0.09891	0.04864	0.02409	0.01470	0.00835
D	2.52143	0.83038	0.24806	0.09678	0.04843	0.02405	0.01484	0.00829
E	7.28191	1.10001	0.35481	0.12192	0.06123	0.02840	0.01750	0.00960

- Arbitrary Array with distant sources

  In this situation it sets the sources in x=-50, y=250, the simulation is

TABLE III
COMPARISON OF MSE FOR THE SI, TAYLOR-SERIES AND PROPOSED METHODS; ARBITRARY ARRAY AND DISTANT SOURCE

MSE	M = 4	M = 5	M = 6	M = 7	M = 8	M = 9	M = 10
A	101881	213.93	48.12	40.40	41.87	40.11	37.23
B	346.86	147.57	44.38	38.64	38.63	36.55	33.80
C	348.74	144.84	44.06	38.41	38.47	36.50	33.87
E	328.82	143.94	44.06	38.54	38.53	36.47	33.73

A: SI method. B: Taylor series method. C: proposed method, {(14*b*), (22*b*), (24)}. E: theoretical MSE of the new method = CRLB.

And my result is below

Table 3

MSE	M=4	M=5	M=6	M=7	M=8	M=9	M=10
B	219.12	108.47	34.72	32.48	32.71	32.01	29.75
C	374.77	109.06	33.68	32.29	32.13	32.25	29.50
E	328.82	143.94	44.06	38.54	38.53	36.47	33.73

- Linear Array with distant sources

We can see actually the change is from the linear array with a source of (8,22) to (-50,250) so we can change it and get the simulation, too.

TABLE IV
COMPARISON OF MSE FOR THE PROPOSED AND TAYLOR-SERIES METHODS; LINEAR ARRAY AND DISTANT SOURCE

MSE	M = 4	M = 5	M = 6	M = 7	M = 8	M = 9	M = 10
B	1797.13	432.68	159.50	69.16	34.56	18.56	10.85
C	1588.31	409.66	153.92	68.41	34.44	18.56	10.86
E	1437.25	408.17	154.05	68.06	34.25	18.57	10.90

B: Taylor series method. C: proposed method, {(28) with $\Psi = Q$, (29)}. E: theoretical MSE of the new method = CRLB.

And my result is

Table 4

MSE	M=4	M=5	M=6	M=7	M=8	M=9	M=10
B	230163.58	299.12	127.21	55.68	28.55	15.64	9.33
C	1066.91	285.03	122.29	56.07	28.98	15.90	9.65
E	1437.25	408.17	154.05	68.06	34.25	18.57	10.90

All the chart simulations are completed and we can see with the difference or the fluctuations of the noise the result and the paper gives may have some deviations but in fact they are very similar. And the deviations may be caused by the noise or the times we iterate thinking the computer velocity.

- Thresholds effect

To compare the thresholds effect we will adjust the Q while we will set the sensors to 9 (M=9) and the sources is x=-50, y=250. It is a distant situation. And as we can see in the paper the simulation is pointed to the CRLB and proposed, DAC, BHB2 but the last two solutions haven’t been explained, so I simulate the

Taylor-series, the proposed_1 and the proposed_2 and the CRLB. We can see the paper gives us a fig

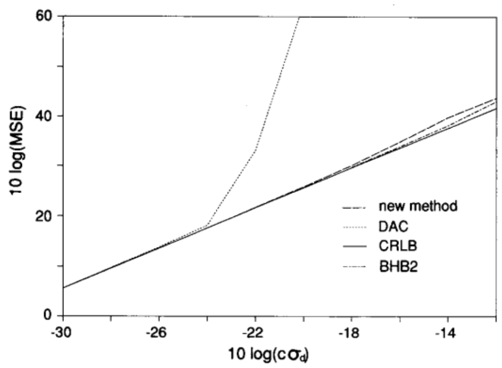
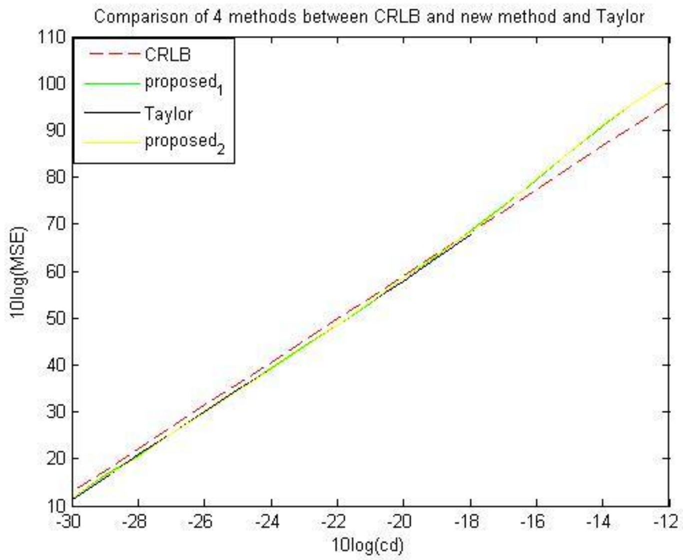
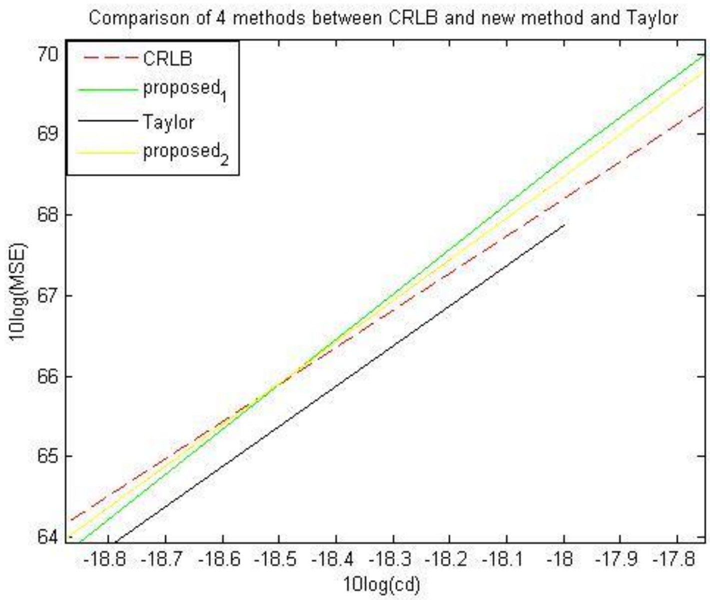


Fig. 4. Comparison of threshold effect between the new method and DAC.

And my simulations is





The first one is the figure as a whole, and the second one is a partial one we amplify it. We can see it is similar as a whole but when the black line reaches some data, it will have singular value so the calculation may have some errors that cannot be ignored, so the line stop at there while the paper has a line which has a sudden change may have some relationship. In all the simulation can be sorted into the chart and figure and as we can see, the deviation actually can be changed with the method we use.

Ⅳ.Conclusions

- With the comparison of the MSE among these methods, we can see that as a whole, when the number of the sensors increases, the MSE will decrease so that we can get the position more accurately. What is more, arbitrary array will have larger error than the linear, and the near sources will have less error than the distant.
- While we get to the methods, we can conclude that the proposed methods in the paper actually have a great advantage in locating the position and decrease the MSE. The SI is the worst methods among all, and our new methods perform better slightly than the Taylor-series. What we should note is that the Taylor-series in fact gives a most favorable initial guess. And the proposed_1 is still a good way than the SI. The CRLB is an identical value which is seen as the theoretical MSE of the solution. So we can get to know the new methods are great as they nearly approach the result of the CRLB.
- In some case, such as linear array of the sensors, the Taylor will have a similar result with the proposed way. But when it comes to the distant, while the M is 4 or 5, Taylor will have large error than the proposed for the linearization error. When the M increases, the error is reduced so it will become normal.
- In the figure, we can see four methods have a rough similarity as a trend. While we make the CRLB as the identical standard, we can get that they have some small deviations and the Taylor will have some situation in the singular error so that it will have a sudden change as we can see in the figure.

Ⅴ.REFERENCES

[R1] Y. T. Chan and K. C. Ho, “A simple and efficient estimator for hyperbolic location”, IEEE Transactions on Signal Processing, vol. 42. no. 8, pp. 1905–1915, Aug. 1994.