TDOA Localization (positioning)

Abhinav Appidi(201431218)
Srinivas Sri(201431226)

Introduction :

Time Delay of Arrival (TDOA) is a technique for locating an acoustic source (ie. gunshot, explosion, etc.) near a receiver array. By exploiting the differences in the arrival time of the sound to the receivers, TDOA locates the source of the sound. As a function, it takes a set of receiver signals as input and returns the coordinates of the source relative to the receiver array.

Let $\{(x_m, y_m, z_m)\}^M_{m=1}$ be the coordinates of M receivers. Let (x, y, z) be the unknown coordinates of the source we are locating. Let tm be the time of transit from the source to receiver m. Let v be the speed of sound (340.29 meters per sec in air). Let Rm = vTm be the distance between between the source and the receiver m. Let τm = Tm − t1 be the difference in transit time between receiver m and receiver 1. Note then, that tau1 = 0.

$$\tau_m = T_m - T_1$$

$$V\tau_m = VT_m - VT_1 = R_m - R_1$$

$$R^2_m = (V\tau_m + R_1)^2 = V^2 \cdot \tau^2_m + 2V\tau_m R_1 + R^2_1$$

$$.0 = V\tau_m + 2R_1 + (R^2_1 - R^2_m)/ V\tau_m \qquad m= 3,4,5,.......,M$$

$$0 = V\tau_m - V\tau_2 + (R^2_1 - R^2_m) /V\tau_m - (R^2_1 - R^2_m)/ V\tau_m$$

$R_m = \sqrt{(x_m - x)^2 + (y_m - y)^2 + (z_m - z)^2}$ → substitute this in above equation.

$R^2 m = x^2 m - 2xmx + x^2 + y^2 m - 2yym + y^2 + z^2 m - 2zmz + z^2$

$R^2 1 - R^2 m = x^2 1 + y^2 1 + z^2 1 - x^2 m - y^2 m - z^2 m - 2x1x - xy1y - 2z1z + 2xxm + 2yym + 2zzm$ m=3,4,5,.......M

From here we get.,

$0 = V\tau m - V\tau 2 + (1/V\tau m) (x^2 1 + y^2 1 + z^2 1 - x^2 m - y^2 m - z^2 m - 2x1x - 2y1y - 2z1z + 2xmx + 2ymy + 2zmz) - (1/V\tau m) (x^2 1 + y^2 1 + z^2 1 - x^2 2 - y^2 2 - z^2 2 - 2x1x - 2y1y - 2z1z + 2x2x + 2y2y + 2z2z)]$ m=3,4,5,.........M

We rewrite the above equation as

$0 = Dm + Amx + Bmy + Cmz$

Where.,

$Am = (1/V\tau m) (-2x1 + 2xm) - (1/V\tau 2) (2x2 - 2x1)$

$Bm = (1/V\tau m) (-2y1 + 2ym) - (1/v\tau 2) (2y2 - 2y1)$

$Cm = (1/V\tau m) (-2z1 + 2zm) - 1\ v\tau 2\ (2z2 - 2z1)$

and

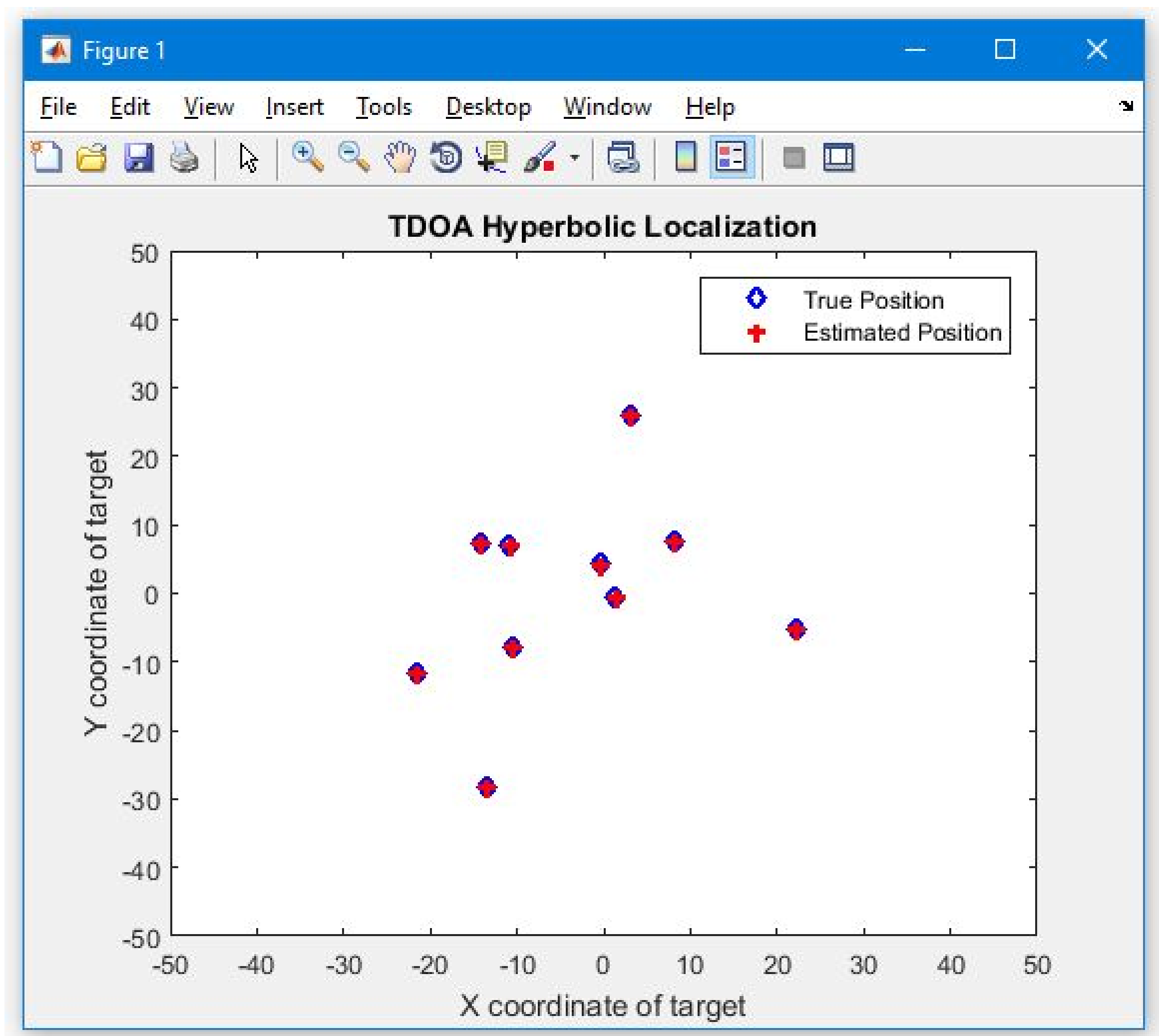
$Dm = V\tau m - V\tau 2 + (1/V\tau m) (x^2 1 + y^2 1 + z^2 1 - x^2 m - y^2 m - z^2 m) - (1/V\tau 2) (x^2 1 + y^2 1 + z^2 1 - x^2 2 - y^2 2 - z^2 2)$

Here m=3,4,5,......M (since we took all as non-coplanar and so we need at least 3 receivers to get 3D location)

We rewrite the above set of M-2 equations into matrix form to get

$$\begin{bmatrix} A_3 & B_3 & C_3 \\ A_4 & B_4 & D_4 \\ \vdots & \vdots & \vdots \\ A_M & B_M & C_M \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -D_3 \\ -D_4 \\ \vdots \\ -D_M \end{bmatrix}$$

From this we solve for x,y,z which gives the location of source(x,y,z)
Matlab code for this is implemented and attached
In matlab code., took many (around 10 signals) and calculated the position of sources for each possibility. Estimated and exact locations are matching and so code is working
Signal is given as wav file.



Matlab code :

```
function TDOAshell
wave = audioread('sample.wav');
wave = wave(:,1);
scale = 0.8/max(wave);
wave = scale*wave;
Trials = 10;
Radius = 50;
N = 8;
Theta = linspace(0,2*pi,N+1);
X = Radius * cos(Theta(1:end-1));
Y = Radius * sin(Theta(1:end-1));
Z = [1:N];
Z = (-1).^Z;
Z = 5*Z+5;
Sen_position = [X.',Y.',Z.'];
Sen_position = [Sen_position];
True_position = zeros(Trials, 3);
Est_position = zeros(Trials,3);

% Generate position of source
for i=1:Trials
   r = rand(1)*50;
   t = rand(1)*2*pi;
   x = r*cos(t);
   y = r*sin(t);
   z = rand(1)*20;
   True_position(i,1) = x;
   True_position(i,2) = y;
   True_position(i,3) = z;
```

```
end

% Generate distances
Distances = zeros(Trials,8);
for i=1:Trials
    for j=1:8
        x1 = True_position(i,1);
        y1 = True_position(i,2);
        z1 = True_position(i,3);
        x2 = Sen_position(j,1);
        y2 = Sen_position(j,2);
        z2 = Sen_position(j,3);
        Distances(i,j) = sqrt((x1-x2)^2 + (y1-y2)^2 + (z1-z2)^2);
    end
end

Distances;
TimeDelay = Distances./340.29;
Padding = TimeDelay*44100;

% Generate the signals
for i=1:Trials
   x = True_position(i,1);
   y = True_position(i,2);
   z = True_position(i,3);
   xstr = num2str(round(x));
   ystr = num2str(round(y));
   zstr = num2str(round(z));
   istr = num2str(i);
   name = strcat( 'Trial_', istr, '_', xstr, '_', ystr, '_', zstr,
'_mdove.wav');
   mic1 = [zeros(round(Padding(i,1)),1) ; wave];
   mic2 = [zeros(round(Padding(i,2)),1) ; wave];
```

```
mic3 = [zeros(round(Padding(i,3)),1) ; wave];
mic4 = [zeros(round(Padding(i,4)),1) ; wave];
mic5 = [zeros(round(Padding(i,5)),1) ; wave];
mic6 = [zeros(round(Padding(i,6)),1) ; wave];
mic7 = [zeros(round(Padding(i,7)),1) ; wave];
mic8 = [zeros(round(Padding(i,8)),1) ; wave];
l1 = length(mic1);
l2 = length(mic2);
l3 = length(mic3);
l4 = length(mic4);
l5 = length(mic5);
l6 = length(mic6);
l7 = length(mic7);
l8 = length(mic8);
lenvec = [l1 l2 l3 l4 l5 l6 l7 l8];
m = max(lenvec);
c = [m-l1, m-l2, m-l3, m-l4, m-l5,m-l6, m-l7, m-l8];
mic1 = [mic1; zeros(c(1),1)];
mic2 = [mic2; zeros(c(2),1)];
mic3 = [mic3; zeros(c(3),1)];
mic4 = [mic4; zeros(c(4),1)];
mic5 = [mic5; zeros(c(5),1)];
mic6 = [mic6; zeros(c(6),1)];
mic7 = [mic7; zeros(c(7),1)];
mic8 = [mic8; zeros(c(8),1)];


mic1 = mic1./Distances(i,1);
mic2 = mic2./Distances(i,2);
mic3 = mic3./Distances(i,3);
mic4 = mic4./Distances(i,4);
mic5 = mic5./Distances(i,5);
mic6 = mic6./Distances(i,6);
```

```
  mic7 = mic7./Distances(i,7);
  mic8 = mic8./Distances(i,8);

multitrack = [mic1, mic2, mic3, mic4, mic5, mic6, mic7, mic8];
%   wavwrite(multitrack, 44100, name);

[x y z] = Locate(Sen_position, multitrack);
Est_position(i,1) = x;
Est_position(i,2) = y;
Est_position(i,3) = z;

end

plot(True_position(:,1),True_position(:,2),'bd',Est_position(:,1),E
st_position(:,2),'r+','LineWidth',2);
legend('True Position','Estimated Position');
xlabel('X coordinate of target');
ylabel('Y coordinate of target');
title('TDOA Hyperbolic Localization');
axis([-50 50 -50 50]);


end



function [x y z] = Locate(Sen_position, multitrack)
% sensor index shift of 1 occurrs here

s = size(Sen_position);
len = s(1);
timedelayvec = zeros(len,1);
for i=1:len
  timedelayvec(i) = timedelayfunc(multitrack(:,1),multitrack(:,i));
```

```
end

timedelayvec;

Amat = zeros(len,1);
Bmat = zeros(len,1);
Cmat = zeros(len,1);
Dmat = zeros(len,1);
for i=3:len
    x1 = Sen_position(1,1);
    y1 = Sen_position(1,2);
    z1 = Sen_position(1,3);
    x2 = Sen_position(2,1);
    y2 = Sen_position(2,2);
    z2 = Sen_position(2,3);
    xi = Sen_position(i,1);
    yi = Sen_position(i,2);
    zi = Sen_position(i,3);
   Amat(i) = (1/(340.29*timedelayvec(i)))*(-2*x1+2*xi) -
(1/(340.29*timedelayvec(2)))*(-2*x1+2*x2);
   Bmat(i) = (1/(340.29*timedelayvec(i)))*(-2*y1+2*yi) -
(1/(340.29*timedelayvec(2)))*(-2*y1+2*y2);
   Cmat(i) = (1/(340.29*timedelayvec(i)))*(-2*z1+2*zi) -
(1/(340.29*timedelayvec(2)))*(-2*z1+2*z2);
   Sum1 = (x1^2)+(y1^2)+(z1^2)-(xi^2)-(yi^2)-(zi^2);
   Sum2 = (x1^2)+(y1^2)+(z1^2)-(x2^2)-(y2^2)-(z2^2);
   Dmat(i) = 340.29*(timedelayvec(i) - timedelayvec(2)) +
(1/(340.29*timedelayvec(i)))*Sum1 -
(1/(340.29*timedelayvec(2)))*Sum2;
end


M = zeros(len,3);
```

```
D = zeros(len,1);
for i=1:len
   M(i,1) = Amat(i);
   M(i,2) = Bmat(i);
   M(i,3) = Cmat(i);
   D(i) = Dmat(i);
end

M = M(3:len,:);
D = D(3:len);


D = D.*-1;

Minv = pinv(M);
T = Minv*(D);
x = T(1);
y = T(2);
z = T(3);

end

function out = timedelayfunc(x,y)
% suppose sampling rate is 44100
% Let Tx be transit time for x
% Let Ty be transit time for y
% out is Ty - Tx

c = xcorr(x, y);
[C I] = max(c);
out = ((length(c)+1)/2 - I)/44100;

end
```