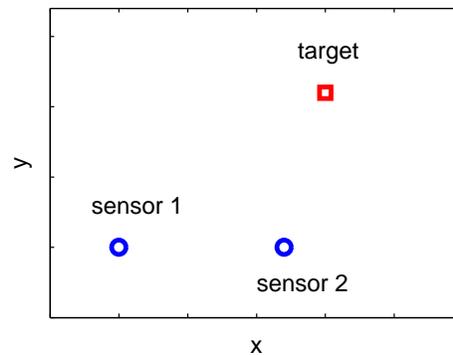


33 Positioning Using Ranging: 2D Case

- Two sensors, denoted 1 and 2 and located at different locations in the $x - y$ plane, make a range measurements to a target t . Let the range from Sensor 1 to the target be r_1 and let the range from Sensor 2 to the target be r_2 ; we call the sensor locations $[x_1, y_1]$ and $[x_2, y_2]$, and the target location $[x_t, y_t]$.



What are the two equations describing the target's $[x_t, y_t]$ position in the horizontal plane, based on the range measurements from the two devices?

- Sketch these constraints for the fixed sensor locations $x_1 = 0, y_1 = 0, x_2 = 1, y_2 = 0$, for two different target locations a) $x_t = 2, y_t = 0$, and b) $x_t = 0.5, y_t = 0.5$. This is a total of four circles to draw.
- Consider your two constraints for a given target location; there are two unknowns $[x_t, y_t]$, so we ought to be able to solve for them. First, solve for x_t by subtracting the two constraint equations, such that the x_t^2 and y_t^2 terms go away; be sure you take advantage of the fact that $x_1 = y_1 = y_2 = 0$! This will let you derive a clean expression for x_t . Then, put this value for x_t into the constraint equation for Sensor 1, and solve for y_t . There will be two solutions for y_t , because this array can't distinguish which side the target is on.
- Now we model the whole thing in a program; consider the first target location. You are to calculate the two ranges that perfect sensors would give, perturb each of the perfect ranges by $0.1 \cdot \text{randn}$ (`randn` is a Gaussian random number with zero mean and unity variance) to simulate some sensor noise in a measurement, and then solve the real-world location problem, i.e., estimate x_t and y_t , using the measured ranges.

The idea is to understand how the noisy range measurements affect the estimate of where the target is. To see this, make a hundred noisy measurements, and plot all the target estimates on an $x - y$ plot that also shows the true target location. *Note that when the noisy measurement makes the square root negative in the equation for y_t , you have to explicitly set $y_t = 0$, because we don't want imaginary results.*

- Repeat this procedure for the second target location.

6. I find that the second configuration is about four times better than the first, in terms of scatter in the estimated location. Explain in words what is happening with the geometry of your constraint circles and the sensor noise.

1. *The range constraints are equations of circles centered at the sensors:*

$$\begin{aligned}(x_t - x_1)^2 + (y_t - y_1)^2 &= r_1^2 \\ (x_t - x_2)^2 + (y_t - y_2)^2 &= r_2^2\end{aligned}$$

2. *The figure below shows all four circles for the constraints.*

3. *Making the suggested manipulations, and with $x_1 = y_1 = y_2 = 0$, we get*

$$\begin{aligned}x_t &= \frac{r_1^2 + x_2^2 - r_2^2}{2x_2} \\ y_t &= \pm \sqrt{r_1^2 - x_t^2}\end{aligned}$$

4. *See the figure.*

5. *Ditto.*

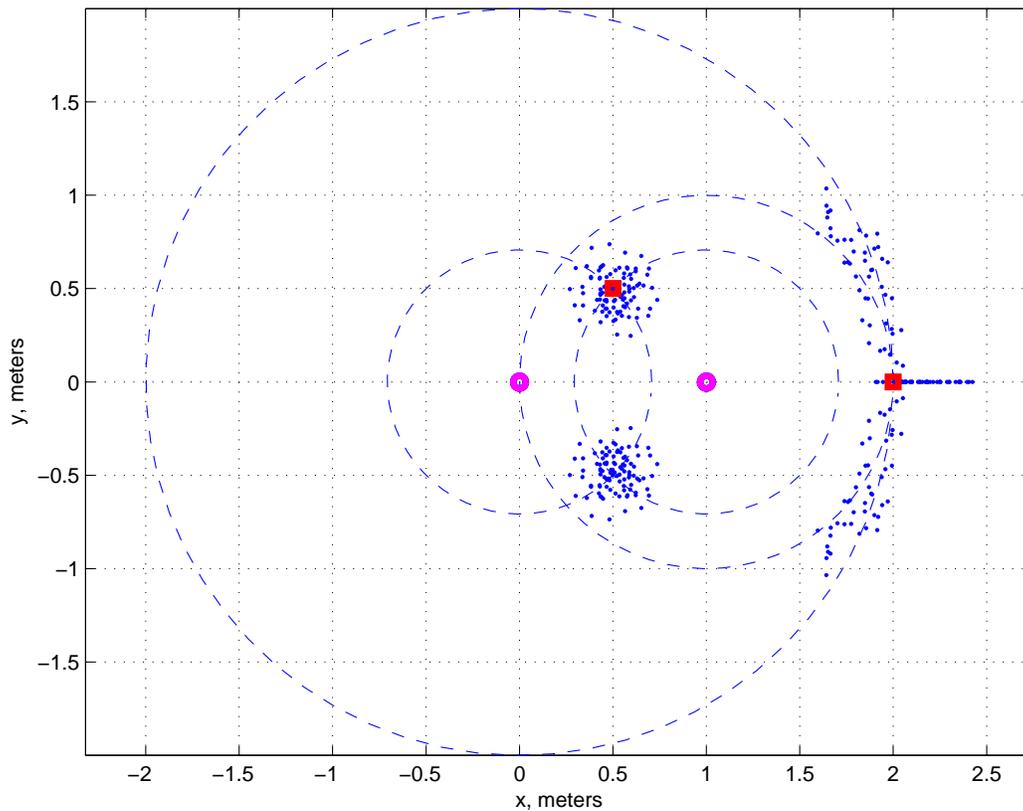
6. *The effect of sensor noise is reduced when the range constraint circles are closer to perpendicular than to tangent. In the former case, the constraints span two directions, which are sufficient to locate the point in the plane; in the latter case, the tangent circles only constrain one direction, namely the radial. The lessons: don't operate near the straight lines that pass through several sensors - the baselines; if you have to, then additional sensors may be needed to get good geometry and a good tight solution. In the figure shown, the standard deviation of error is 0.30 meters for the poor configuration, and 0.064 meters for the good one.*

```
%-----
% Analysis of a two-dimension range positioning system

% FSH MIT Mechanical Engineering April 2008

clear all; close all;

% set the positions of the receivers
x1 = 0 ;
y1 = 0 ;
x2 = 1 ;
```



```

y2 = 0 ;

% set the position of the target (both will be done in turn)
xt = [2 0.5] ;
yt = [0 0.5] ;

sigma = .1 ;    % standard deviation of noise to be applied
                % to range measurements

figure(1);clf;hold off;
for j = 1:length(xt),

    plot(x1,y1,'bo',x2,y2,'bo',xt(j),yt(j),'rs','LineWidth',3);
    axis('equal');
    grid on ;
    xlabel('x, meters');
    ylabel('y, meters');

% make the calculation of ranges from each receiver to target
r1 = sqrt((xt(j)-x1)^2 + (yt(j)-y1)^2) ;

```

```

r2 = sqrt((xt(j)-x2)^2 + (yt(j)-y2)^2) ;

% plot the range circles
figure(1);hold on;
cang = cos(0:.1:2*pi) ;
sang = sin(0:.1:2*pi);
plot(r1*cang+x1,r1*sang+y1,'--');
plot(r2*cang+x2,r2*sang+y2,'--');

% enter the loop to study sensitivity to noise and geometry
for i = 1:100,

    % add random noise to make range measurements
    r1m = r1 + sigma*randn ;
    r2m = r2 + sigma*randn ;

    % do the estimations of xt and yt from these ranges
    xte = (r1m^2 + x2^2 - r2m^2) / 2 / x2 ;
    % (only do the square root if the argument is positive...
    if (r1m^2 - xte^2) > 0,
        yte(1) = sqrt(r1m^2 - xte^2) ;
        yte(2) = -sqrt(r1m^2 - xte^2) ;
    % ... otherwise we have to just guess
    else,
        yte(1) = 0 ;
        yte(2) = 0 ;
    end;

    figure(1); hold on;
    plot([xte xte],yte,'.');

    % calculate an error - the distance between true and estimated
    err(i) = sqrt((xte-xt(j))^2 + (yte(1)-yt(j))^2);
end;

% (make this plot again in case the scatter points cover up)
plot(x1,y1,'mo',x2,y2,'mo',xt(j),yt(j),'rs','LineWidth',3);

disp(sprintf('For %d samples, RMS error is: %g meters.', i, ...
    std(err)));

end;

print -depsc acousticRangingIn2DSolution.eps

```


MIT OpenCourseWare
<http://ocw.mit.edu>

2.017J Design of Electromechanical Robotic Systems
Fall 2009

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.