EECS 16A Designing Information Devices and Systems I
Fall 2023 Lecture Notes Note 25

25.1 Trilateration

Recall that GPS works by receiving signals from many satellites, computing the time delay t of each signal, and using these time delays to determine the receiver's location.

In the previous note, we saw how to use correlation to compute these time delays. Now, we will see how to use these time delays to determine the actual position of our receiver, in a process known as *trilateration*.

25.2 Computing distances from time delays

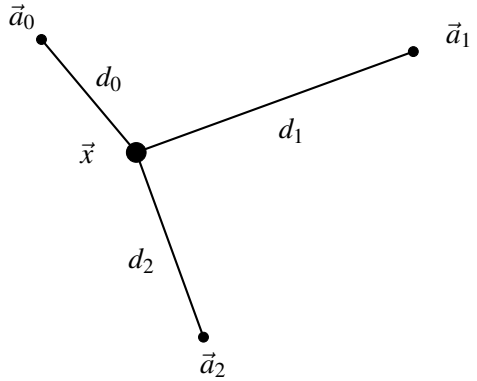
Imagine that each of the beacons start broadcasting their signal at time $t = 0$. Assume also that our receiver starts recording at this same time $t = 0$. Since our signals travel at a finite velocity v (such as the speed of light for radio waves, or the speed of sound for audio signals), it will take some time for each signal to reach our receiver. Specifically, if beacon i is at some distance d_i from our receiver, then it will take time d_i/v to reach the receiver, so the time delay becomes

$$t_i = \frac{d_i}{v}.$$

Thus, we can rearrange the above to compute all the distances as $d_i = vt_i$.

25.3 Determining our location

Let's imagine we have a situation like the one below. We know the locations of the beacons $\vec{a}_0, \vec{a}_1, \vec{a}_2$, but don't know the location of the point at $\vec{x}$. Our goal will be to solve for $\vec{x}$, given the distances d_0, d_1, d_2.



By the Pythagorean theorem, we have that

$$\begin{aligned}\|\vec{a}_0 - \vec{x}\|^2 &= d_0^2 \\ \|\vec{a}_1 - \vec{x}\|^2 &= d_1^2 \\ \|\vec{a}_2 - \vec{x}\|^2 &= d_2^2.\end{aligned}$$

We wish to solve for $\vec{x}$. It is natural to rewrite the squared norms using inner products and expand out the squared binomial terms, in order to separate out the terms involving $\vec{x}$ from the terms that are known.

Doing so, we obtain

$$\begin{aligned}\vec{a}_0^T\vec{a}_0 - 2\vec{a}_0^T\vec{x} + \vec{x}^T\vec{x} &= d_0^2 \\ \vec{a}_1^T\vec{a}_1 - 2\vec{a}_1^T\vec{x} + \vec{x}^T\vec{x} &= d_1^2 \\ \vec{a}_2^T\vec{a}_2 - 2\vec{a}_2^T\vec{x} + \vec{x}^T\vec{x} &= d_2^2.\end{aligned}$$

We're interested in solving for $\vec{x}$. If the above system were a linear system, then we could do so easily, using Gaussian elimination. Unfortunately, the $\vec{x}^T\vec{x}$ quadratic term prevents us from doing so. To eliminate that terms, we use a "trick" - we will subtract the first of our equations from each of the other two. In doing so, the $\vec{x}^T\vec{x}$ term will subtract out, leaving us with a linear system!

Doing so, we obtain the two equations:

$$\begin{aligned}\vec{a}_1^T\vec{a}_1 - \vec{a}_0^T\vec{a}_0 - 2\vec{a}_1^T\vec{x} + 2\vec{a}_0^T\vec{x} &= d_1^2 - d_0^2 \\ \vec{a}_2^T\vec{a}_2 - \vec{a}_0^T\vec{a}_0 - 2\vec{a}_2^T\vec{x} + 2\vec{a}_0^T\vec{x} &= d_2^2 - d_0^2.\end{aligned}$$

Simplifying and collecting terms within each equation, we can rewrite the above system as

$$\begin{bmatrix}2(\vec{a}_0^T - \vec{a}_1^T) \\ 2(\vec{a}_0^T - \vec{a}_2^T)\end{bmatrix}\vec{x} = \begin{bmatrix}\vec{a}_1^T\vec{a}_1 - \vec{a}_0^T\vec{a}_0 + d_0^2 - d_1^2 \\ \vec{a}_2^T\vec{a}_2 - \vec{a}_0^T\vec{a}_0 + d_0^2 - d_2^2\end{bmatrix}.$$

Now, we have two linear equations. So if we are working in two dimensions, we can solve for the components of $\vec{x}$ and recover our position. If we are working in three dimensions, however, then we do not have enough equations - we would need to add one more beacon $\vec{a}_3$ in order to get our third linear equation. More generally, if we are working in n-dimensional space, we need at least $n+1$ beacons in order to determine our position, since we "lose" one beacon in our subtraction trick in order to linearize the system.

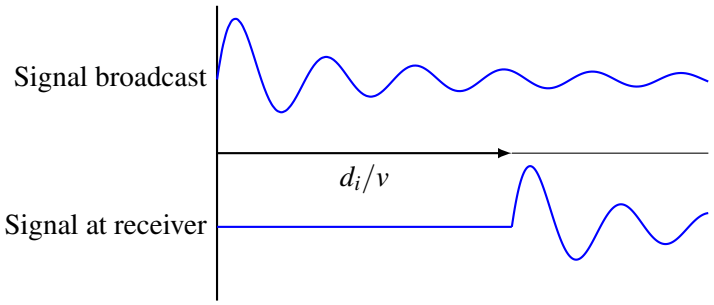
25.4 Unsynchronized Clocks

However, there is one critical assumption baked into the above approach. How do we find the t_i? We assume that our receiver starts recording at the same time $t = 0$ that the beacons start broadcasting. Then we use cross-correlation to compute each t_i, and then solve for d_i and do trilateration as described above.

But how does our receiver know to start recording at exactly that time $t = 0$? It must have some clock that is synchronized with the beacons, in order to know when to begin recording. In reality, however, such a clock does not exist - while the beacons can all be synchronized to start broadcasting at the same time, the

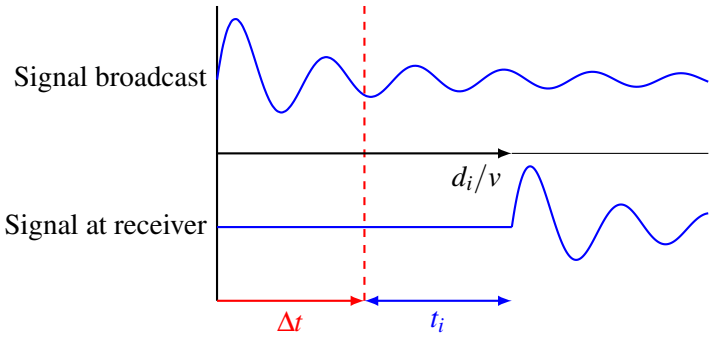
receiver cannot guarantee that it starts recording at exactly that same time. Instead, it begins recording at some time Δt, representing the offset from $t = 0$ when we start recording. Note that Δt may be either positive or negative, but is unknown to the receiver.

Consider the following figure, representing the signals broadcast by the beacons and received by the receiver:



Notice that the received signal is delayed by some amount, due to its travel time. As we discussed earlier, letting d_i be the distance between the receiver and the beacon, this delay will equal d_i/v.

If the receiver were able to synchronize its recording with the time at which the signal was broadcast, then we could use cross correlation to compute the above delay, recover d_i, and perform trilateration in manner described above. However, what if we were unable to start our recording at exactly $t = 0$, but instead started at some unknown time $t = \Delta t$?



As shown above, the recording begins only after some time Δt. When we perform cross-correlation, we determine that the signal has been delayed by some time t_i after we began recording. But as shown in the above diagram, the signal has in fact been traveling for a time $t_i + \Delta t$, not simply t_i.

So we may write our observations t_i

$$\begin{aligned} t_0 &= \frac{d_0}{v} - \Delta t \\ t_1 &= \frac{d_1}{v} - \Delta t \\ &\vdots \\ t_{n-1} &= \frac{d_{n-1}}{v} - \Delta t. \end{aligned}$$

So what do we do now? Since Δt is unknown, we can't recover the d_i directly and use our previous approach to solve for our position. Instead, we can rearrange and write the d_i in terms of both t_i (which are known) and Δt (which is not!):

$$\begin{aligned} d_0 &= v(\Delta t + t_0) \\ d_1 &= v(\Delta t + t_1) \\ &\vdots \\ d_{n-1} &= v(\Delta t + t_{n-1}). \end{aligned}$$

Observe that this system of equations looks very similar to the system in the previous section, except in the previous system of equations $\Delta t = 0$. This makes sense, since in our previous setup we were told that the receiver began recording at time 0, whereas now all we know is that it starts recording at some unknown time Δt. Thus, we see from these equations that the time taken for the ith signal to travel from the beacon to the receiver, known as the "time of flight," is $\Delta t + t_i$.

Note that since all the *beacons* are still synchronized, they all broadcast together at the same time, so we only have one unknown Δt, not n of them.

Unfortunately, since we don't know Δt, we can't use the same approach that we did earlier. Ideally, we'd like to eliminate Δt from our equations, so that the distances are the only remaining unknowns. One natural way of doing this is to subtract the first equation from all the rest, to obtain:

$$\begin{aligned} d_1 - d_0 &= v(t_1 - t_0) \\ d_2 - d_0 &= v(t_2 - t_0) \\ &\vdots \\ d_{n-1} - d_0 &= v(t_{n-1} - t_0). \end{aligned}$$

For convenience, we will define the "time difference of arrival", or TDOA, of each beacon i to be $\tau_i = t_i - t_0$. Observe that while we cannot directly measure the time of flight' $t_i - \Delta t$ since Δt is unknown, we *can* measure these time differences of arrival, as all the t_i *are* known.

Observe that d_0 occurs many times in the above system of equations. To make future calculations simpler, we will choose our origin to be the position of the 0th beacon and place the receiver at the position $\vec{x}$ in this coordinate system, so

$$d_0^2 = \vec{x}^T \vec{x}.$$

As before, we let the coordinates of the ith beacon be $\vec{a}_i$, with $\vec{a}_0 = \vec{0}$.

Substituting in the distances computed using the Pythagorean theorem, we obtain the equations

$$\begin{aligned} \sqrt{(\vec{x} - \vec{a}_1)^T (\vec{x} - \vec{a}_1)} - \sqrt{\vec{x}^T \vec{x}} &= v\tau_1 \\ \sqrt{(\vec{x} - \vec{a}_2)^T (\vec{x} - \vec{a}_2)} - \sqrt{\vec{x}^T \vec{x}} &= v\tau_2 \\ &\vdots \\ \sqrt{(\vec{x} - \vec{a}_{n-1})^T (\vec{x} - \vec{a}_{n-1})} - \sqrt{\vec{x}^T \vec{x}} &= v\tau_{n-1}. \end{aligned}$$

How can we solve these equations? The square roots certainly aren't helpful, so let's see if we can get rid of

them. Squaring each equation would remove some square roots, but then we'd get a cross term caused by the product of the two terms on the left hand side, which looks difficult to deal with. Instead, let's pull one square root to the other side, so that when we then square, we won't get as ugly a cross term.

Rearranging in such a manner, we obtain

$$\begin{aligned}\sqrt{(\vec{x}-\vec{a}_1)^T(\vec{x}-\vec{a}_1)} &= v\tau_1+\sqrt{\vec{x}^T\vec{x}}\\ \sqrt{(\vec{x}-\vec{a}_2)^T(\vec{x}-\vec{a}_2)} &= v\tau_2+\sqrt{\vec{x}^T\vec{x}}\\ &\vdots\\ \sqrt{(\vec{x}-\vec{a}_{n-1})^T(\vec{x}-\vec{a}_{n-1})^2} &= v\tau_{n-1}+\sqrt{\vec{x}^T\vec{x}}.\end{aligned}$$

And now squaring, we obtain

$$\begin{aligned}(\vec{x}-\vec{a}_1)^T(\vec{x}-\vec{a}_1) &= (v\tau_1)^2+\vec{x}^T\vec{x}+2v\tau_1\sqrt{\vec{x}^T\vec{x}}\\ (\vec{x}-\vec{a}_2)^T(\vec{x}-\vec{a}_2) &= (v\tau_2)^2+\vec{x}^T\vec{x}+2v\tau_2\sqrt{\vec{x}^T\vec{x}}\\ &\vdots\\ (\vec{x}-\vec{a}_{n-1})^T(\vec{x}-\vec{a}_{n-1}) &= (v\tau_{n-1})^2+\vec{x}^T\vec{x}+2v\tau_{n-1}\sqrt{\vec{x}^T\vec{x}}\end{aligned}$$

Does this look any better? Ideally, we'd get a linear equation, which we could then solve using Gaussian elimination or least squares, depending on the number of beacons. But there are still terms of the form $\vec{x}^T\vec{x}$ and $2\sqrt{\vec{x}^T\vec{x}}$ that keep this system nonlinear. Can we get rid of these terms?

The first thing to observe is that, by expanding out the left-hand-side, we get a $\vec{x}^T\vec{x}$ term, which cancels with the right hand side. Awesome! Performing this expansion, we obtain

$$\begin{aligned}\vec{x}^T\vec{x}-2\vec{a}_1^T\vec{x}+\vec{a}_1^T\vec{a}_1 &= (v\tau_1)^2+\vec{x}^T\vec{x}+2v\tau_1\sqrt{\vec{x}^T\vec{x}}\\ \vec{x}^T\vec{x}-2\vec{a}_2^T\vec{x}+\vec{a}_2^T\vec{a}_2 &= (v\tau_2)^2+\vec{x}^T\vec{x}+2v\tau_2\sqrt{\vec{x}^T\vec{x}}\\ &\vdots\\ \vec{x}^T\vec{x}-2\vec{a}_{n-1}^T\vec{x}+\vec{a}_{n-1}^T\vec{a}_{n-1} &= (v\tau_{n-1})^2+\vec{x}^T\vec{x}+2v\tau_{n-1}\sqrt{\vec{x}^T\vec{x}}.\end{aligned}$$

And canceling, we then obtain

$$\begin{aligned}-2\vec{a}_1^T\vec{x}+\vec{a}_1^T\vec{a}_1 &= (v\tau_1)^2+2v\tau_1\sqrt{\vec{x}^T\vec{x}}\\ -2\vec{a}_2^T\vec{x}+\vec{a}_2^T\vec{a}_2 &= (v\tau_2)^2+2v\tau_2\sqrt{\vec{x}^T\vec{x}}\\ &\vdots\\ -2\vec{a}_{n-1}^T\vec{x}+\vec{a}_{n-1}^T\vec{a}_{n-1} &= (v\tau_{n-1})^2+2v\tau_{n-1}\sqrt{\vec{x}^T\vec{x}}.\end{aligned}$$

Is this system of equations linear? Recall that the $\vec{a}_i$ are constants (since their values are known), so the presence of $\vec{a}_i^T\vec{a}_i$ terms *does not* make the system nonlinear, since they are also just constant terms.

However, the $2\sqrt{\vec{x}^T\vec{x}}$ term is a nonlinear term, since $\vec{x}$ is an unknown. So our system isn't linear just yet. To get rid of this term, we could try to use the same "trick" we used in the previous section, where we

subtracted equation 1 from all the other equations to get rid of the $\vec{x}^T\vec{x}$ term. Unfortunately, here there are multiplicative coefficients in front of the $2\sqrt{\vec{x}^T\vec{x}}$ term, so we can't directly subtract it out. Instead, we need to first scale each equation in order to get rid of the constant, and then do the subtraction trick.

Dividing each equation by $v\tau_i$ to get rid of the constant in front of $2\sqrt{\vec{x}^T\vec{x}}$, we obtain the system

$$\begin{aligned}
\frac{\vec{a}_1^T\vec{a}_1 - 2\vec{a}_1^T\vec{x}}{v\tau_1} &= v\tau_1 + 2\sqrt{\vec{x}^T\vec{x}} \\
\frac{\vec{a}_2^T\vec{a}_2 - 2\vec{a}_2^T\vec{x}}{v\tau_2} &= v\tau_2 + 2\sqrt{\vec{x}^T\vec{x}} \\
&\vdots \\
\frac{\vec{a}_{n-1}^T\vec{a}_{n-1} - 2\vec{a}_{n-1}^T\vec{x}}{v\tau_{n-1}} &= v\tau_{n-1} + 2\sqrt{\vec{x}^T\vec{x}}.
\end{aligned}$$

Now, we can subtract the first equation from each of the others individually, to obtain

$$\begin{aligned}
\frac{\vec{a}_2^T\vec{a}_2 - 2\vec{a}_2^T\vec{x}}{v\tau_2} - \frac{\vec{a}_1^T\vec{a}_1 - 2\vec{a}_1^T\vec{x}}{v\tau_1} &= v\tau_2 - v\tau_1 \\
\frac{\vec{a}_3^T\vec{a}_3 - 2\vec{a}_3^T\vec{x}}{v\tau_3} - \frac{\vec{a}_1^T\vec{a}_1 - 2\vec{a}_1^T\vec{x}}{v\tau_1} &= v\tau_3 - v\tau_1 \\
&\vdots \\
\frac{\vec{a}_{n-1}^T\vec{a}_{n-1} - 2\vec{a}_{n-1}^T\vec{x}}{v\tau_{n-1}} - \frac{\vec{a}_1^T\vec{a}_1 - 2\vec{a}_1^T\vec{x}}{v\tau_1} &= v\tau_{n-1} - v\tau_1.
\end{aligned}$$

Whew! That was a lot of messy algebra there at the end. But have we ended up with a linear system of equations? Well, notice that the only "quadratic" terms remaining are of the form $\vec{a}_i^T\vec{a}_i$. But since all the $\vec{a}_i$ are known to us, these are just constant terms. The only terms involving the unknown $\vec{x}$ are linear terms, so we have indeed obtained a linear system of equations!

Let's rewrite our equations in a more familiar form, collecting the constant terms on the right hand side, and stacking everything into a matrix-vector system:

$$\begin{bmatrix}
2\vec{a}_1^T/(v\tau_1) - 2\vec{a}_2^T/(v\tau_2) \\
2\vec{a}_1^T/(v\tau_1) - 2\vec{a}_3^T/(v\tau_3) \\
\vdots \\
2\vec{a}_1^T/(v\tau_1) - 2\vec{a}_{n-1}^T/(v\tau_{n-1})
\end{bmatrix}\vec{x} = \begin{bmatrix}
v\tau_2 - v\tau_1 + \frac{\vec{a}_1^T\vec{a}_1}{v\tau_1} - \frac{\vec{a}_2^T\vec{a}_2}{v\tau_2} \\
v\tau_3 - v\tau_1 + \frac{\vec{a}_1^T\vec{a}_1}{v\tau_1} - \frac{\vec{a}_3^T\vec{a}_3}{v\tau_3} \\
\vdots \\
v\tau_{n-1} - v\tau_1 + \frac{\vec{a}_1^T\vec{a}_1}{v\tau_1} - \frac{\vec{a}_{n-1}^T\vec{a}_{n-1}}{v\tau_{n-1}}
\end{bmatrix}.$$

Now, we have made the linear structure of our system of equations entirely explicit. Observe that we end up with $n-2$ equations and a single unknown vector $\vec{x}$. Let x be in d-dimensional space. Thus, we need at least $n \geq d+2$ beacons to determine our location when our clocks are not synchronized with those of the beacons - one more than we did in the case of synchronized clocks.

Intuitively, this can be justified by recognizing that we really have one more unknown - the offset Δt - so we clearly need at least $d+1$ beacons. The fourth and last equation is needed to "linearize" our system by

subtracting out the quadratic terms. Of course, having more equations doesn't hurt. If we have more than four equations, then we end up with an overdetermined system, which can still be solved using least squares.