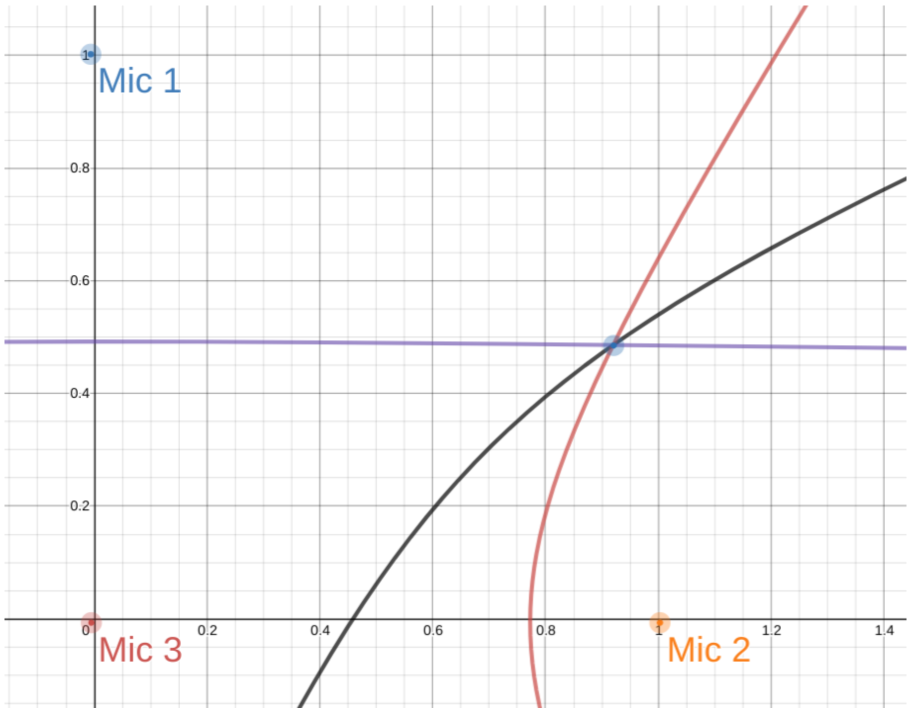
Parametric Based 2D Multilateration

Michael Gorbunov



Overview

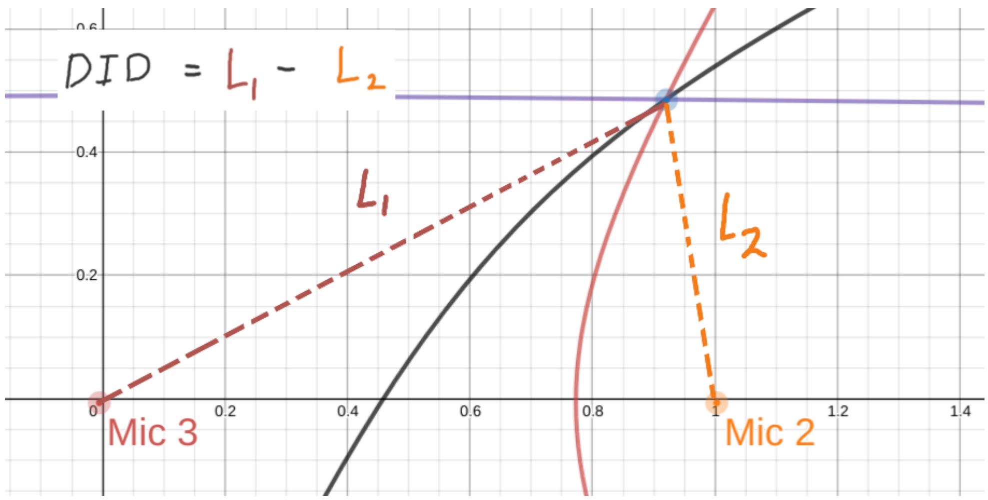
The following work was done for my end of year capstone project, and I worked alongside a partner. Still, the work presented here is entirely my own. The motivation for this project was mostly that it would be interesting to try and do sound-based positioning, and my teacher allowed it as the final project was supposed to be very general.

Presented is an approach for position solving. The final project involved a lot of audio processing, but what is presented is the most mathematical part. Sadly, the final position solver was a boring gradient descent on a simple error function, so this work did not actually make it into the final result. Still, I quite enjoyed deriving it.

Problem

Given some microphones in known positions and a sound source at an unknown position, the goal is to reconstruct the position of the sound source.

Because we do not know when the sound was created, we cannot determine the distance from each microphone and use triangulation. Instead, we must examine pairs of microphones. The sound source will hit each microphone at a different time. Using a time difference of arrival, we can calculate a **Difference In Distance (DID)**, i.e. how much further or closer the sound is to one microphone compared to another. Below is an illustration showing this simple idea, with the DID between mic 1 and mic 2.



The question then becomes: Given many DIDs (determined via audio processing on recordings), how to reconstruct the original position.

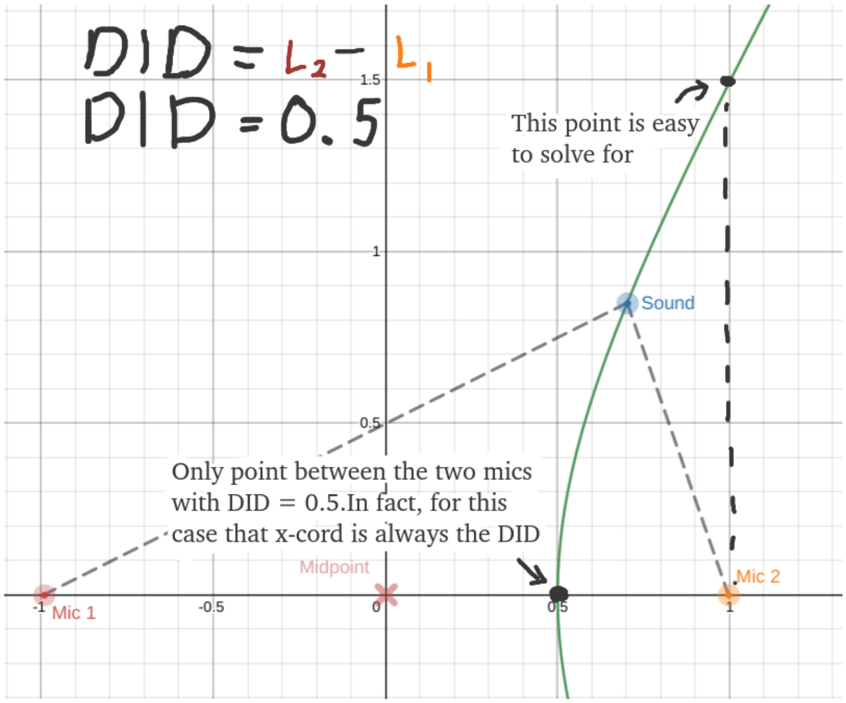
Solution

Given a pair of microphones and a DID, we are able to restrict the possible locations of the sound to a hyperbola. This should make sense because a common construction of a hyperbola is a difference in distance.

However, hyperbolas in general can be stretched vertically or horizontally and so are not always strictly a difference in distance. Also, hyperbolas in standard form contain 2 disjoint segments, but we can eliminate one curve from to the sign of the DID.

So, instead we construct a hyperbola parametrically. Given 2 points (microphones) and a DID, there are 3 trivial points to find which will lie on the hyperbola. One directly between the two microphones, and two directly above or below the microphones. The one between the two microphones will be a distance DID from the microphone. If two microphones are 2 units apart, and the DID is 1/2, that point can only be at (½, 0).

The other two points are easy to analytically produce, but do not have the same nice intuition for being trivial. Below is a diagram of these 3 points, which in reality is only 2 points of information due to the symmetry of the hyperbola.



These points are enough to restrict a hyperbola.

Solving this problem with general microphone placement can be tricky. So, first it is solved with microphone positions (-1, 0) and (1, 0), and then the points from the parametric are transformed via a matrix (scaling & rotation) and vector addition (translation) to properly fit arbitrary positions.

When solving at nicer positions, it is necessary to scale the DID. If the original microphone positions are (5, 1), (5, 5), then the scaled DID would be half the original DID, because the distance from (-1, 0) to (1, 0) is half that of (5, 1) to (5, 5).

Then, all that's needed to do is construct a hyperbola for each DID (i.e. each pair of microphones), and the search space for possible sound locations is greatly reduced.

Demos

Hopefully this has been interesting. If you want to try it out, linked here are 2 desmos demos which let you drag points around.

2 Microphones (1 hyperbola): https://www.desmos.com/calculator/pamkqaqiht
3 microphones and error curves: https://www.desmos.com/calculator/uib4cmahog

Applicability

This approach only narrows the search space, and a regression step would still be required to determine where the parametrics intersect. However, it does reduce the search space by a dimension. In 3D, each DID would produce a surface thus allowing sampling along some UV plane.

Mathematically, the exact collision of the curves could be determined, perhaps even in closed form, but in real life the curves will never perfectly intersect so I did not pursue it.

Further Thoughts

Because I had a project to do, I wasn't able to dwell further to generalize dimensions, etc. There is one non-obvious idea I remember considering, described below.

Excess Constraints
With 3 microphones, there are 3 pairs, and so 3 lines. However, a point only needs the intersection of 2 lines, so in an ideal world this third line is extraneous. Also, not all sets of 3 DIDs are valid. This means in real life, all pairs of DIDs produced will be a little off and not perfectly intersecting. But, it would be possible to somehow regress the 3 DIDs to an ideal set of 3 DIDs before creating parametrics.

This also may make regression along the curves unnecessary, and gives some merit to the idea of solving for the position purely analytically, which may cut down a lot of regression.

Also, it would be a very interesting visual to see the volume or surface of all valid DIDs. Of course with more pairs this space quickly becomes very large in dimension, but in 3-space it may give some insight as to how to find ideal DID sets from some non optimal set.

Work Log

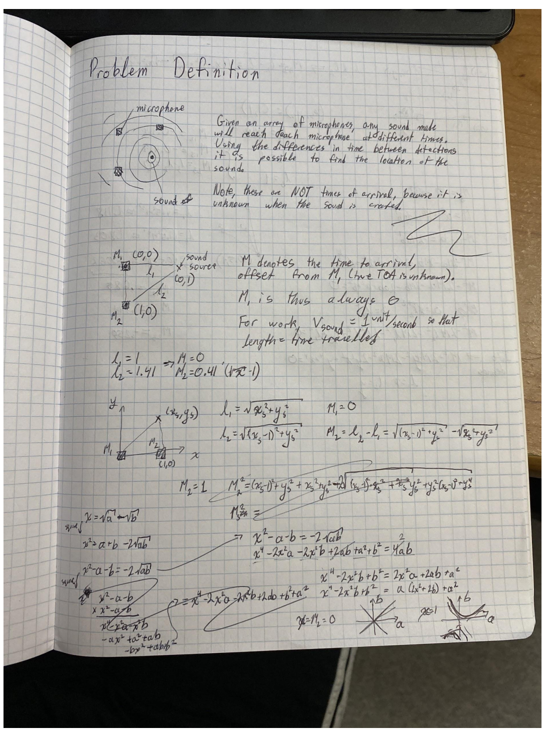
Iff. you are interested, below is my worklog documenting the development of this math. This was for a class, so the record is a proof of work, and not really an explanation of the result. Still, it may be interesting to see my thought process. All dates are 2021.

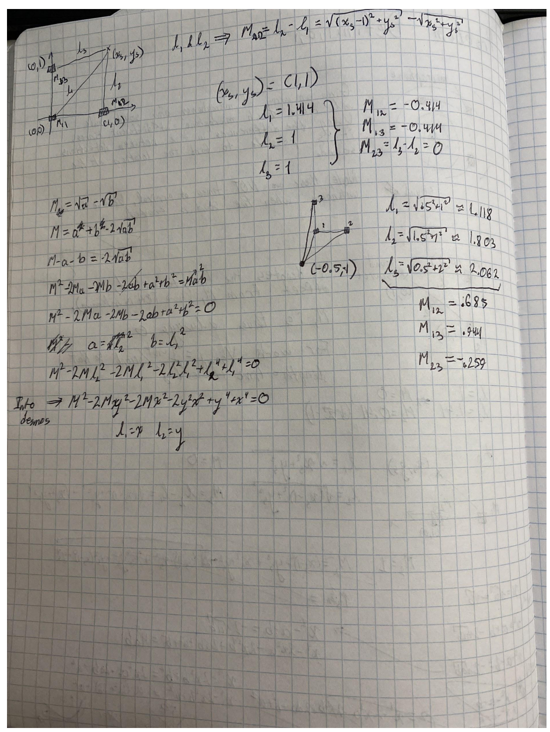
Apr 29, 2H Home

Today I worked out an approach for multilateration that I understand. Many papers I found on the topic were over my head.

Initially, I tried working out algebraically solutions but very soon arrived at implicit equations that were difficult if not impossible to make explicit. I did not consider polar coordinate systems or parametrizing, but overall this was not fruitful.

See below some pictures from a notebook.

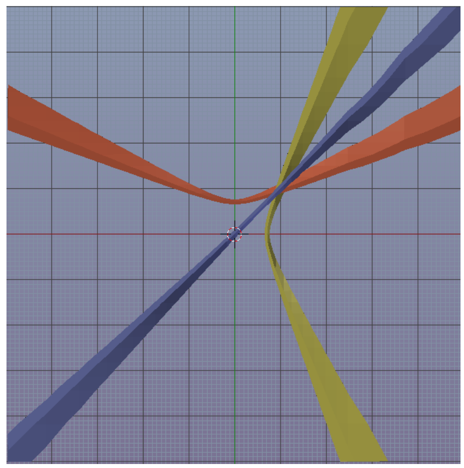




After this another idea came to mind, trying to work graphically. A description of my approach is in the design document, but essentially the approach works backwards.

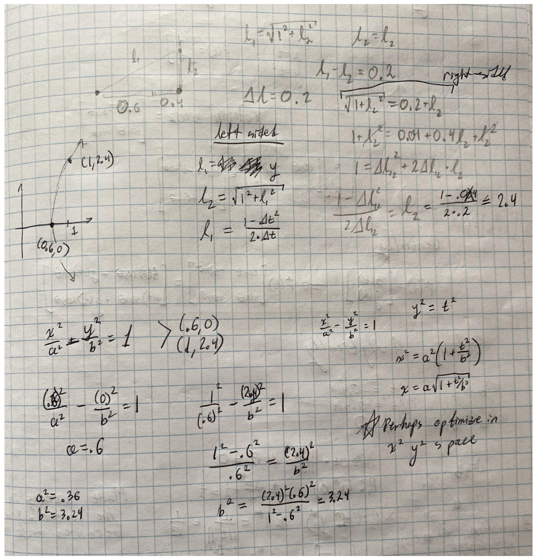
Instead of starting with a time difference between two microphones and finding an analytic curve to restrict the domain of possible sound sources, it starts with a function of two variables (a point in the xy plane) and returns the resulting time difference. Then, to constrain the xy plane, a slice is taken at a specific value by sampling the plane, returning an approximate curve of solutions. Do this with multiple equations and you have multiple curves converging on a single point.

Below is an example of 3 curves converging on the point (1.0, 1.0), created in Blender.



Apr 30, 1H Class + 1H Home

Continuing to work on the math of multilateration, I found a way to generate parametric curves from a difference in distance. I will need to clean everything up, but here are some images of my notebook:



The idea is actually fairly simple. Given a difference in distance, there are 2 trivial points to generate, one at the midpoint of the two microphones and one directly above/below one of the microphones. From there, it's easy to solve for a^2 and b^2 of a hyperbola in the form:

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

This was the key idea that allowed everything else to work. Moving forward, I now have parametric equations for the hyperbola, and so to make it work for any arbitrary microphone configuration simply requires some transformations (matricies!).

May 1 & 2, 6H Home

To start, I tried plugging my equations into Desmos to see if they work. This actually revealed an error but also verified my results. In the demo below, you can adjust the 'd' value, or difference in distance between the two microphones, and 'E', or the amount of error. The error hyperbolas are drawn by just adding both ways (+ & -) to the 'd' value.
https://www.desmos.com/calculator/sursvqwcfg

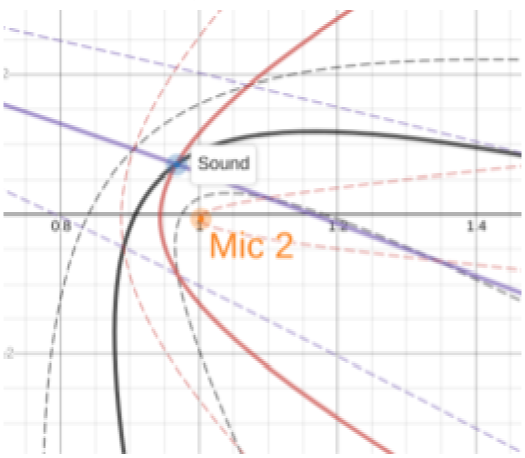
The previous demo works but only for microphones located at +-1 on the x-axis. The next step is to generalize for arbitrary microphone positions.

The math behind this is basically just a matrix transformation, which does make the parametrics in Desmos a little convoluted, but the final result works. You can drag around the positions of the microphones and the sound source, and a correct hyperbola is always drawn.

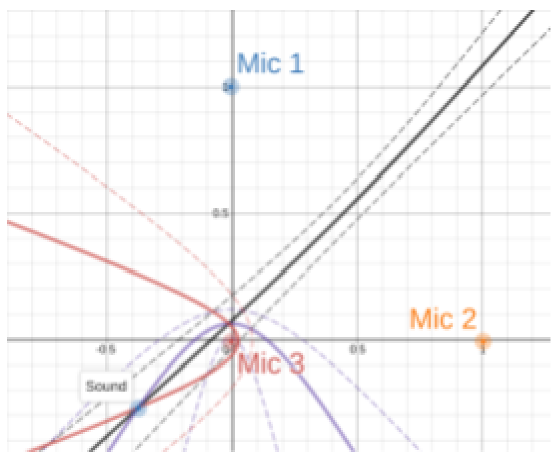
Note, technically within Desmos I have perfect information about the point. However, the hyperbola is only drawn based on the difference in distance from the sound to the two microphones. In real life, this will be determined by multiplying the time difference in arrival by the speed of sound.
https://www.desmos.com/calculator/pamkqaqiht

The final step to seeing this work is introducing one more microphone for a total of 3. This was just some more work, and you can view the demo below. It also draws error lines, and reveals that the error shape to be somewhat hexagonal.
https://www.desmos.com/calculator/uib4cmahog

The above demo also helps reveal the importance of microphone placement. As the sound source gets closer and closer to the microphones, the error shape becomes increasingly distorted. Also, when outside of the surrounding square, the error shape becomes extremely distorted.



The error lines (dotted) get warped when near the microphones.



The error lines (dotted) become incomprehensible outside the region.