

Multilateration: Background Physics and Implementation

What is Multilateration?

Multilateration is a positioning technique that determines the location of an unknown point by using distance measurements from multiple known reference points. The fundamental principle involves creating geometric constraints (circles in 2D, spheres in 3D) centered at each known location with radii equal to the measured distances. The unknown position is found at the intersection of these geometric constraints.

In the context of an agent swarm, such as drones assisting a GPS-denied rover conducting land surveys, multilateration enables precise positioning by having multiple drones with known locations measure their distances to the rover and calculate its position through geometric intersection.

Multilateration in Two Dimensions

Constraints for 2D Multilateration

- **Minimum Reference Points:** A minimum of **three known locations** is required to uniquely determine an unknown location in two dimensions
- **Distance Separation:** Ideally, no known location should be the same distance from the unknown location as any other known location, as this creates ambiguous intersections
- **Angular Separation:** Known locations should not be positioned at the same angle from the unknown location to avoid geometric degeneracy
- **Non-collinear Positioning:** The three reference points should not be collinear to ensure a unique solution

In a drone swarm scenario, this means at least three drones with known positions are needed to calculate the rover's location effectively.

Mathematical Overview for 2D

The mathematical foundation of 2D multilateration involves creating circles centered at each known position with radii equal to the measured distances:

For each known position (x_i, y_i) and measured distance d_i to the unknown position (x, y) :

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2$$

With three or more known positions, we generate multiple circles. The unknown position is located at the point where the maximum number of circle intersections occur.

The system of equations can be solved by:

1. **Circle Generation:** Create circles with origins at known locations and radii equal to measured distances
2. **Intersection Finding:** Calculate all pairwise circle intersections
3. **Convergence Point:** Identify the point with maximum intersections (ideally where all circles meet)

For three circles, the mathematical solution involves solving the system:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 = d_3^2 \end{cases}$$

Circle-Circle Intersection Calculations

To find intersection points between two circles with centers (x_1, y_1) and (x_2, y_2) and radii r_1 and r_2 :

Step 1: Calculate the distance between centers:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Step 2: Check for intersection conditions:

- If $d > r_1 + r_2$: Circles are too far apart (no intersection)
- If $d < |r_1 - r_2|$: One circle is contained within the other (no intersection)
- If $d = 0$ and $r_1 = r_2$: Circles are identical (infinite intersections)

Step 3: Calculate intersection points when they exist:

$$a = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

$$h = \sqrt{r_1^2 - a^2}$$

Step 4: Find the midpoint between intersections:

$$x_m = x_1 + a \cdot \frac{x_2 - x_1}{d}$$

$$y_m = y_1 + a \cdot \frac{y_2 - y_1}{d}$$

Step 5: Calculate the two intersection points:

$$x_{int1} = x_m + h \cdot \frac{y_1 - y_2}{d}$$

$$y_{int1} = y_m + h \cdot \frac{x_2 - x_1}{d}$$

$$x_{int2} = x_m - h \cdot \frac{y_1 - y_2}{d}$$

$$y_{int2} = y_m - h \cdot \frac{x_2 - x_1}{d}$$

Finding the Point of Maximum Intersections

Step 1: Generate all pairwise circle intersections

For n circles, compute intersections for all $\binom{n}{2}$ circle pairs, creating a set of candidate points.

Step 2: Count intersections for each candidate point

For each candidate point (x_c, y_c) , count how many circles it lies on within a tolerance ϵ :

$$\text{count} = \sum_{i=1}^n \begin{cases} 1 & \text{if } |\sqrt{(x_c - x_i)^2 + (y_c - y_i)^2} - d_i| \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

Step 3: Select optimal point

Choose the candidate point with the highest intersection count. If multiple points have the same maximum count, select based on additional criteria (e.g., geometric center, minimum total error).

Multilateration in Three Dimensions

Constraints for 3D Multilateration

- **Minimum Reference Points:** A minimum of **four known locations** is required to uniquely determine an unknown location in three dimensions
- **Distance Separation:** No known location should be the same distance from the unknown location as any other known location
- **Angular Separation:** Known locations should not be positioned at the same angle from the unknown location
- **Non-coplanar Positioning:** The four reference points should not be coplanar to ensure a unique 3D solution

Additional Constraints for Drone Swarm Applications

- **Altitude Constraints:** Maintaining all drones within the same altitude creates an imaginary XY plane constraint
- **Advantages:** Easier 2D visualization on screens, reduced computational power for position rendering
- **Disadvantages:** Added constraint makes optimal drone positioning more challenging for effective multilateration

Mathematical Overview for 3D

Three-dimensional multilateration uses spheres instead of circles, with the unknown position located at the intersection of multiple spheres.

For each known position (x_i, y_i, z_i) and measured distance d_i :

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = d_i^2$$

The solution process involves:

1. **Sphere Generation:** Create spheres centered at known locations with radii equal to measured distances
2. **Sphere-Sphere Intersections:** When two spheres intersect, they form a circle (intersection plane)
3. **Circle Intersections:** Multiple intersection circles from sphere pairs converge at the unknown point
4. **Convergence Point:** Find the point of maximum circle intersections

Sphere Intersection Mathematics

When two spheres intersect, they create a circular intersection. For spheres centered at (x_1, y_1, z_1) and (x_2, y_2, z_2) with radii r_1 and r_2 :

The intersection circle lies on a plane perpendicular to the line connecting the sphere centers. The circle's center is located at:

$$\mathbf{c} = \mathbf{p}_1 + a \cdot \frac{\mathbf{p}_2 - \mathbf{p}_1}{|\mathbf{p}_2 - \mathbf{p}_1|}$$

where:

$$a = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

and d is the distance between sphere centers.

The intersection circle's radius is:

$$h = \sqrt{r_1^2 - a^2}$$

With four or more spheres, multiple intersection circles are generated, and their convergence point represents the unknown position. This is typically solved using least-squares optimization to find the point that minimizes the sum of squared distances to all intersection planes.

Sphere-Sphere Intersection Calculations

To find the circular intersection between two spheres:

Step 1: Define sphere parameters

- Sphere 1: center $\mathbf{c}_1 = (x_1, y_1, z_1)$, radius r_1
- Sphere 2: center $\mathbf{c}_2 = (x_2, y_2, z_2)$, radius r_2

Step 2: Calculate distance between centers:

$$d = |\mathbf{c}_2 - \mathbf{c}_1| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Step 3: Check intersection conditions:

- If $d > r_1 + r_2$: Spheres don't intersect
- If $d < |r_1 - r_2|$: One sphere contains the other
- If $d = 0$: Spheres are concentric

Step 4: Calculate intersection circle parameters:

$$a = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

$$h = \sqrt{r_1^2 - a^2}$$

Step 5: Find intersection circle center:

$$\mathbf{p} = \mathbf{c}_1 + a \cdot \frac{\mathbf{c}_2 - \mathbf{c}_1}{d}$$

Step 6: Define intersection plane

The intersection circle lies on a plane with:

- **Center:** \mathbf{p}
- **Normal vector:** $\mathbf{n} = \frac{\mathbf{c}_2 - \mathbf{c}_1}{d}$
- **Radius:** h

Finding the Point of Maximum Circle Intersections

Step 1: Generate all sphere-pair intersection circles

For n spheres, compute intersection circles for all $\binom{n}{2}$ sphere pairs.

Step 2: Convert circles to plane equations

Each intersection circle defines a plane equation:

$$n_x(x - p_x) + n_y(y - p_y) + n_z(z - p_z) = 0$$

Step 3: Solve system of plane equations using least squares

Construct the overdetermined system $\mathbf{A}\mathbf{x} = \mathbf{b}$:

$$\mathbf{A} = \begin{bmatrix} n_{1x} & n_{1y} & n_{1z} \\ n_{2x} & n_{2y} & n_{2z} \\ \vdots & \vdots & \vdots \\ n_{mx} & n_{my} & n_{mz} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{n}_1 \cdot \mathbf{p}_1 \\ \mathbf{n}_2 \cdot \mathbf{p}_2 \\ \vdots \\ \mathbf{n}_m \cdot \mathbf{p}_m \end{bmatrix}$$

Step 4: Solve for unknown position:

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

Alternative Method: Point-Circle Distance Minimization

Step 1: Define objective function

For each intersection circle i with center \mathbf{p}_i , normal \mathbf{n}_i , and radius h_i :

$$f_i(\mathbf{x}) = ||\mathbf{x} - \mathbf{p}_i - (\mathbf{n}_i \cdot (\mathbf{x} - \mathbf{p}_i))\mathbf{n}_i| - h_i|^2$$

Step 2: Minimize total error:

$$\mathbf{x}_{\text{optimal}} = \arg \min_{\mathbf{x}} \sum_{i=1}^m f_i(\mathbf{x})$$

This minimization finds the point that lies closest to all intersection circles simultaneously.

Applications in GPS-Denied Land Surveying

Multilateration provides a robust solution for rovers operating in GPS-denied environments by leveraging drone swarms as mobile reference beacons. The technique enables:

- **Continuous Positioning:** Real-time location updates as the rover moves

- **Scalable Accuracy:** More reference drones generally improve positioning precision
- **Flexible Deployment:** Drones can be repositioned to optimize geometric configuration
- **Redundancy:** Additional reference points beyond the minimum provide error checking and improved reliability

The mathematical foundations ensure that with proper geometric distribution of reference points and accurate distance measurements, multilateration can achieve positioning accuracy suitable for precision land surveying applications.