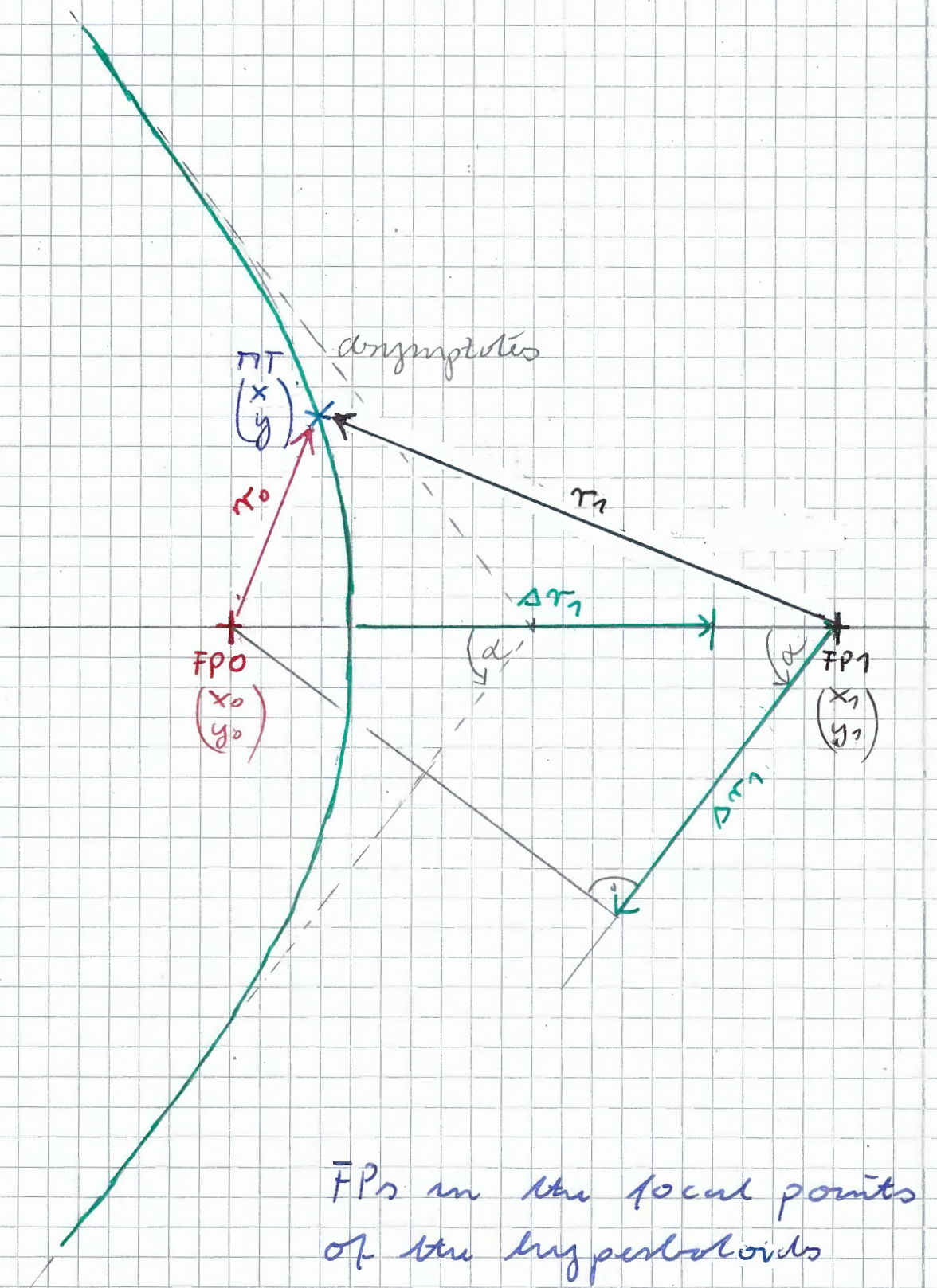
Time Difference of Arrival (TDOA)

Scenario



FPs in the focal points of the hyperboloids

Assumptions

- for simplicity 2D-scenario, extension to 3D-scenarios straightforward
- choose coordinates such that FP0 is at the origin
- typical situation: synchronized clocks at the FPs (transmitters) but no time reference at the MT (receiver), e.g., GPS uses synchronized atomic clocks at the satellites
- choose FP0 as the reference and measure K-1 range differences

$$\Delta r_k = r_k - r_0 \quad | \; k = 1 \ldots K-1$$

(equivalent to delay differences, "Time Difference of Arrival", TDOA)

Time Difference of Arrival (TDOA)

- each measured range difference

$$\Delta r_n = r_n - r_0$$

$$= \sqrt{(x-x_n)^2 + (y-y_n)^2} - \sqrt{x^2+y^2}$$

defines a hyperboloid of possible MT positions

⇒ find the intersection point of the hyperboloids, various algorithms to solve this nonlinear (in general overdetermined) system of equations

- in 2D scenario 2 unknowns x and y
  - 3 FPs ⇒ 2 equations but in general two intersection points, ambiguity might be resolved by some side information
  - ≥ 4 FPs ⇒ one unique intersection point (if no measurement errors)
- as compared to time of arrival methods one more FP is required (the time at the MT is one additional unknown which needs to be determined implicitely)

Analytical Method

- requires $K = 3$ FPs in 2D scenarios
- idea: introduce r_0 as an additional unknown
- first step: compute x and y as linear functions of r_0

$$r_n^2 = (\Delta r_n + r_0)^2$$

$$(x - x_n)^2 + (y - y_n)^2 = \Delta r_n^2 + 2\Delta r_n r_0 + r_0^2$$

$$\underbrace{x^2 + y^2}_{r_0^2} - 2xx_n - 2yy_n + x_n^2 + y_n^2 = \Delta r_n^2 + 2\Delta r_n r_0 + r_0^2$$

$\Rightarrow$ linear system of equations

$$2\begin{pmatrix} x_1 & y_1 \\ x_2 & y_2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 - 2\Delta r_1 r_0 - \Delta r_1^2 \\ x_2^2 + y_2^2 - 2\Delta r_2 r_0 - \Delta r_2^2 \end{pmatrix}$$

from this x and y can be obtained as linear functions of r_0

- second step: substituting these x and y being linear functions of r_0 into

$$r_0^2 = x^2 + y^2$$

one obtains a quadratic equation for r_0

$\Rightarrow$ can be easily solved and with the so obtained r_0 the final results for x and y can be obtained

Least Squares Method

- requires $K \geq 4$ FPs in 2D scenarios
- $K-1$ linear equations for the three variables x, y and r_0 can be obtained:

$$r_k^2 = (\Delta r_k + r_0)^2$$

$$(x - x_k)^2 + (y - y_k)^2 = \Delta r_k^2 + 2\,\Delta r_k\, r_0 + r_0^2$$

$$\underbrace{x^2 + y^2}_{r_0^2} - 2x x_k - 2y y_k + x_k^2 + y_k^2 = \Delta r_k^2 + 2\,\Delta r_k\, r_0 + r_0^2$$

$$\Rightarrow \; +2x x_k + 2y y_k + 2\Delta r_k\, r_0 = x_k^2 + y_k^2 - \Delta r_k^2$$

- set up an (over-) determined linear system of equations

$$2\underbrace{\begin{pmatrix} x_1 & y_1 & \Delta r_1 \\ \vdots & \vdots & \vdots \\ x_{K-1} & y_{K-1} & \Delta r_{K-1} \end{pmatrix}}_{A} \cdot \underbrace{\begin{pmatrix} x \\ y \\ r_0 \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} x_1^2 + y_1^2 - \Delta r_1^2 \\ \vdots \\ x_{K-1}^2 + y_{K-1}^2 - \Delta r_{K-1}^2 \end{pmatrix}}_{b}$$

$$A \cdot x = b$$

- least squares pseudosolution

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

Gauss-Newton Method

- idea: linearize the range difference equations around an initial estimate $\begin{pmatrix} x(n) \\ y(n) \end{pmatrix}$

$$\Delta r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} - \sqrt{x^2 + y^2}$$

$$\approx \underbrace{\sqrt{(x(n) - x_i)^2 + (y(n) - y_i)^2} - \sqrt{x^2(n) + y^2(n)}}_{\Delta r_i(n)}$$

$$+ \frac{\partial \Delta r_i}{\partial x} \underbrace{(x - x(n))}_{\Delta x(n)}$$

$$+ \frac{\partial \Delta r_i}{\partial y} \underbrace{(y - y(n))}_{\Delta y(n)}$$

$$= \Delta r_i(n)$$

$$+ \left(\frac{x(n) - x_i}{r_i(n)} - \frac{x(n)}{r_0(n)} \right) \Delta x(n)$$

$$+ \left(\frac{y(n) - y_i}{r_i(n)} - \frac{y(n)}{r_0(n)} \right) \Delta y(n)$$

where $r_i(n) = \sqrt{(x(n) - x_i)^2 + (y(n) - y_i)^2}$

$$r_0(n) = \sqrt{x^2(n) + y^2(n)}$$

- set up an (over-) determined linear system of equations for computing the correction $\begin{pmatrix}\Delta x\\ \Delta y\end{pmatrix}$, requires $K \geq 3$ FPs in 2D scenarios

$$\underbrace{\begin{pmatrix} \frac{x(n)-x_1}{r_1(n)} - \frac{x(n)}{r_0(n)} & \frac{y(n)-y_1}{r_1(n)} - \frac{y(n)}{r_0(n)} \\ \vdots & \vdots \\ \frac{x(n)-x_{K-1}}{r_{K-1}(n)} - \frac{x(n)}{r_0(n)} & \frac{y(n)-y_{K-1}}{r_{K-1}(n)} - \frac{y(n)}{r_0(n)} \end{pmatrix}}_{A(n)} \cdot \underbrace{\begin{pmatrix}\Delta x(n)\\ \Delta y(n)\end{pmatrix}}_{\Delta(n)} = \underbrace{\begin{pmatrix} \Delta r_1 - \Delta r_1(n) \\ \vdots \\ \Delta r_{K-1} - \Delta r_{K-1}(n) \end{pmatrix}}_{b(n)}$$

- least squares pseudosolution

$$\Delta(n) = \left(A^T(n) \cdot A(n)\right)^{-1} \cdot A^T(n) \cdot b(n)$$

- obtain updated estimates

$$x(n+1) = x(n) + \Delta x(n)$$
$$y(n+1) = y(n) + \Delta y(n)$$

- repeat this procedure iteratively until desired accuracy / stop criterion is reached, typically good convergence