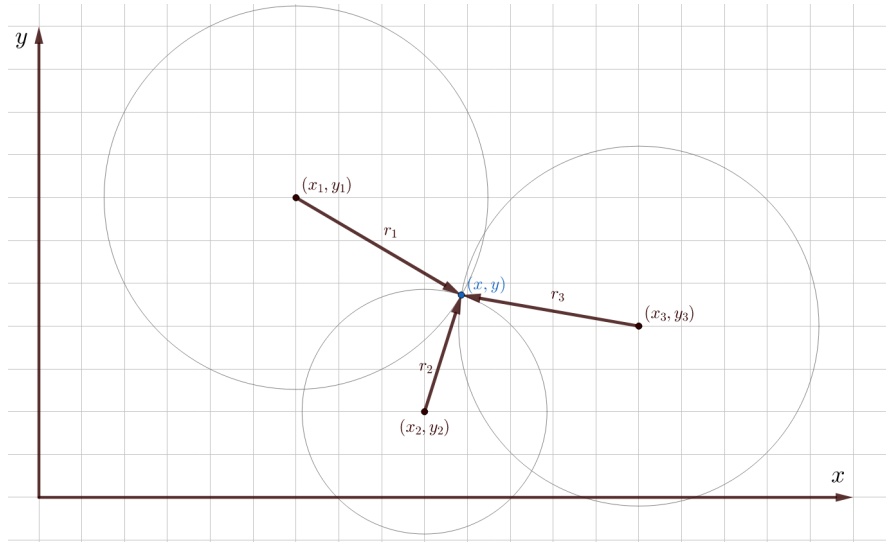


Error Propagation

1 Trilateration

Premise of trilateration is that there are three beacons from which the distance can be measured. Need at least three beacons for unique intersection.



Will consider all pods within range. Pod range is roughly 60 meters. For n pods, need to find point (x, y) which satisfies all of these equations:

$$\begin{aligned}
 (x - x_1)^2 + (y - y_1)^2 &= r_1^2 \\
 (x - x_2)^2 + (y - y_2)^2 &= r_2^2 \\
 (x - x_3)^2 + (y - y_3)^2 &= r_3^2 \\
 &\vdots \\
 (x - x_n)^2 + (y - y_n)^2 &= r_n^2
 \end{aligned}$$

Subtract equation i from $i - 1$ for i greater than 1. Forms system of equations which can be solved.

$$\begin{aligned}
 -2x(x_1 - x_2) - 2y(y_1 - y_2) &= r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \\
 -2x(x_2 - x_3) - 2y(y_2 - y_3) &= r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \\
 &\vdots \\
 -2x(x_{n-1} - x_n) - 2y(y_{n-1} - y_n) &= r_{n-1}^2 - r_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2
 \end{aligned}$$

For $n < 2$ this obviously fails since there will be only one equation for two unknowns. For $n = 3$ there is a unique solution, for $n > 3$ we can solve the least squares problem:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = (A^T A)^{-1} A^T b$$

$$A = \begin{bmatrix} -2x(x_1 - x_2) & -2y(y_1 - y_2) \\ -2x(x_2 - x_3) & -2y(y_2 - y_3) \\ \vdots & \vdots \\ -2x(x_{n-1} - x_n) & -2y(y_{n-1} - y_n) \end{bmatrix} \quad b = \begin{bmatrix} r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2 \\ r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2 \\ \vdots \\ r_{n-1}^2 - r_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix}$$

```

def trilateration_eqn_npods(n):
    if n < 3:
        return "You can't TRIlaterate with less than 3, dummy"

    x, y = var('x', 'y')

    x_loc = list(var('x_{}'.format(i)) for i in range(1, n + 1))
    y_loc = list(var('y_{}'.format(i)) for i in range(1, n + 1))
    r = list(var('r_{}'.format(i)) for i in range(1, n + 1))

    if n == 3:
        distance_equations = []

        for i in range(1, n):
            distance_equations.append(-2 * x * (x_loc[i - 1] - x_loc[i]) \
                                      -2 * y * (y_loc[i - 1] - y_loc[i]) == \
                                      r[i - 1] ** 2 - r[i] ** 2 \
                                      - x_loc[i - 1] ** 2 + x_loc[i] ** 2 \
                                      - y_loc[i - 1] ** 2 + y_loc[i] ** 2)

        res = solve(distance_equations, x, y)
        x_func = res[0][0].right()
        y_func = res[0][1].right()

        return x_func.simplify_full(), y_func.simplify_full()

    if n > 3:
        A = matrix(SR, n - 1, 2)
        b = matrix(SR, n - 1, 1)

        for i in range(1, n):
            A[i-1] = (-2 * (x_loc[i - 1] - x_loc[i]), -2 * (y_loc[i - 1] - y_loc[i]))
            b[i-1] = r[i - 1] ** 2 - r[i] ** 2 - x_loc[i - 1] ** 2 \
                    + x_loc[i] ** 2 - y_loc[i - 1] ** 2 + y_loc[i] ** 2

        x_hat = (A.T * A).inverse() * A.T * b

        return x_hat[0].simplify_full(), x_hat[1].simplify_full()

```

For $n = 3$:

$$x = \frac{y_1 y_2^2 - (y_1 - y_2) y_3^2 - (r_2^2 - r_3^2 - x_2^2 + x_3^2) y_1 + (r_1^2 - r_3^2 - x_1^2 + x_3^2 - y_1^2) y_2 - (r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2) y_3}{2((x_2 - x_3) y_1 - (x_1 - x_3) y_2 + (x_1 - x_2) y_3)}$$

$$y = -\frac{x_1 x_2^2 - (x_1 - x_2) x_3^2 - (x_2 - x_3) y_1^2 + (x_1 - x_3) y_2^2 - (x_1 - x_2) y_3^2 - (r_2^2 - r_3^2) x_1 + (r_1^2 - r_3^2 - x_1^2) x_2 - (r_1^2 - r_2^2 - x_1^2 + x_2^2) x_3}{2((x_2 - x_3) y_1 - (x_1 - x_3) y_2 + (x_1 - x_2) y_3)}$$

For $n = 4$: (This shows less than half– it's quite the expression)

$$x = -\frac{(x_1 - x_3) y_1 y_2^3 + (2(x_1 - x_2) y_1 - (2x_1 - 3x_2 + x_4) y_2) y_3^3 + ((x_1 - x_2) y_1 - (x_1 - 2x_2 + x_3) y_2 - (x_2 - x_3) y_3) y_4^3 - (x_2^3 - x_2 x_3^2 + 2x_3^3 - x_3 x_4^2)}{2((x_2 - x_3) y_1 - (x_1 - x_3) y_2 + (x_1 - x_2) y_3)}$$

$$y = \frac{(x_2^2 - 2x_2 x_3 + 2x_3^2 - 2x_3 x_4 + x_4^2) y_1^3 - (x_1 x_2 - (x_1 + x_2) x_3 + 2x_3^2 - 2x_3 x_4 + x_4^2) y_1 y_2^2 + (x_1^2 - 2x_1 x_3 + 3x_3^2 - 4x_3 x_4 + 2x_4^2) y_2^3 + (2x_1^2 - 4x_1 x_2 + 2x_2^2 - 2x_2 x_3 + 2x_3^2 - 2x_3 x_4 + x_4^2) y_2 y_3^2 - (x_1^2 - 2x_1 x_2 + x_2^2 - 2x_2 x_3 + 2x_3^2 - 2x_3 x_4 + x_4^2) y_3^3 - (x_1^2 - 2x_1 x_2 + x_2^2 - 2x_2 x_3 + 2x_3^2 - 2x_3 x_4 + x_4^2) y_3 y_4^2 - (x_1^2 - 2x_1 x_2 + x_2^2 - 2x_2 x_3 + 2x_3^2 - 2x_3 x_4 + x_4^2) y_4^3}{2((x_2 - x_3) y_1 - (x_1 - x_3) y_2 + (x_1 - x_2) y_3)}$$

The trilateration depends on both the range measured from each pod r_i and the location of the pod (x_i, y_i) .

Recall uncertainty in a function of many variables is written as:

$$\sigma_f = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 \sigma_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 \sigma_y^2 + \dots}$$

We'll call the functions giving x and y components of the trilateration output g and h :

$$\begin{aligned} x &= g(\{(x_i, y_i)\}_{i=1}^n, \{r_i\}_{i=1}^n) \\ y &= h(\{(x_i, y_i)\}_{i=1}^n, \{r_i\}_{i=1}^n) \end{aligned}$$

The uncertainty introduced by range errors σ_{r_i} and position errors $(\sigma_{x_i}, \sigma_{y_i})$ is given below, considering just three pods.

$$\begin{aligned} \sigma_{x, \text{rover}} &= \sqrt{\left(\frac{\partial g}{\partial r_1}\right)^2 \sigma_{r_1}^2 + \left(\frac{\partial g}{\partial r_2}\right)^2 \sigma_{r_2}^2 + \left(\frac{\partial g}{\partial r_3}\right)^2 \sigma_{r_3}^2 + \left(\left(\frac{\partial g}{\partial x_1}\right)^2 \sigma_{x_1}^2 + \left(\frac{\partial g}{\partial y_1}\right)^2 \sigma_{y_1}^2\right) + \left(\left(\frac{\partial g}{\partial x_2}\right)^2 \sigma_{x_2}^2 + \left(\frac{\partial g}{\partial y_2}\right)^2 \sigma_{y_2}^2\right) + \left(\left(\frac{\partial g}{\partial x_3}\right)^2 \sigma_{x_3}^2 + \left(\frac{\partial g}{\partial y_3}\right)^2 \sigma_{y_3}^2\right)} \\ \sigma_{y, \text{rover}} &= \sqrt{\left(\frac{\partial h}{\partial r_1}\right)^2 \sigma_{r_1}^2 + \left(\frac{\partial h}{\partial r_2}\right)^2 \sigma_{r_2}^2 + \left(\frac{\partial h}{\partial r_3}\right)^2 \sigma_{r_3}^2 + \left(\left(\frac{\partial h}{\partial x_1}\right)^2 \sigma_{x_1}^2 + \left(\frac{\partial h}{\partial y_1}\right)^2 \sigma_{y_1}^2\right) + \left(\left(\frac{\partial h}{\partial x_2}\right)^2 \sigma_{x_2}^2 + \left(\frac{\partial h}{\partial y_2}\right)^2 \sigma_{y_2}^2\right) + \left(\left(\frac{\partial h}{\partial x_3}\right)^2 \sigma_{x_3}^2 + \left(\frac{\partial h}{\partial y_3}\right)^2 \sigma_{y_3}^2\right)} \end{aligned}$$

2 Deployment

The deployment direction will be specified by an azimuth (α) actuator, and the distance will be determined by the elevation angle. It's possible to estimate the distance travelled using the elevation angle and the kinematics equation, but it's much more likely that the RF ranging will be more accurate.

For measuring deployment location relative to the rover we will then use the azimuth angle α offset from the rover orientation θ and the RF range r . (α, θ measured CCW from x axis, arbitrarily).

The x and y uncertainty in the pod's location will be written as σ_{xi}, σ_{yi} , where i is the pod number. This value depends on the uncertainty in the rover location while it's being deployed, $\sigma_{x,rover}, \sigma_{y,rover}$ and the uncertainty in the RF measurement σ_{ri} .

The deployment distances are given as:

$$\begin{aligned}x_i &= r \cos(\theta + \alpha) \\y_i &= r \sin(\theta + \alpha)\end{aligned}$$

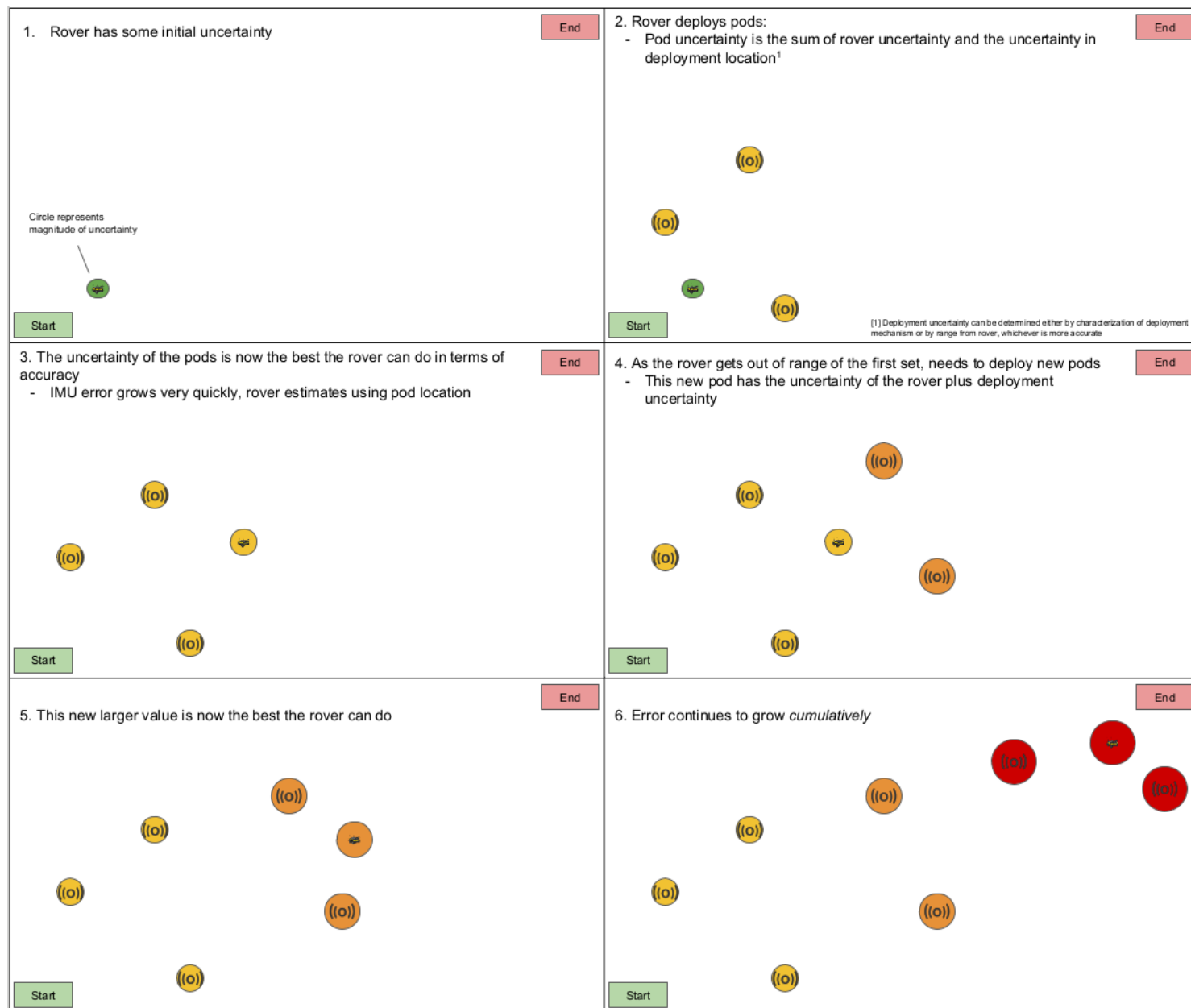
$$\begin{aligned}\sigma_{xi} &= \sigma_{x,rover} + \sigma_{ri} \cos(\theta + \alpha) \\\sigma_{yi} &= \sigma_{y,rover} + \sigma_{ri} \sin(\theta + \alpha)\end{aligned}$$

Using the RF measurement has the advantage of being agnostic to initial velocity and friction in the deployment; the azimuth angle accuracy is the major factor from the deployment system.

Note, however, that this assumes flat ground (otherwise pod could roll in an angular direction) and that the wind effect is negligible.

3 Cumulative Deployment Error

One major problem from the previous section is that the uncertainty for each deployed pod depends on the uncertainty of the rover position. This is problematic because the rover then needs to use this pod's location for trilateration; this error gets added on and gets contributes to the uncertainty in the next pod's location.



The magnitude of the problem depends on the range of each of the pods— error accumulates as the rover moves out of range of each set of pods, and as the rover gets further from each pod.

4 Example Mission
