Relative position in 2D space

April 6, 2022

0.1 Python implementation

```
import numpy as np

# receive points
x1, y1 = (10, 10)
x2, y2 = (2, 5)
x3, y3 = (7, 1)

# to compute radius, simply use
# r1 = time_of_signal_to_reach_receiver_1 * (speed_of_signal)
# r2 = time_of_signal_to_reach_receiver_2 * (speed_of_signal)
# r3 = time_of_signal_to_reach_receiver_3 * (speed_of_signal)

def computePosition2DSpace(r1, r2, r3):
    A_matr = 2 * np.array([[x3 - x1, y3 - y1],
                           [x3 - x2, y3 - y2]])

    b_vect = np.array([[(r1**2 - r3**2) - (x1**2 - x3**2) - (y1**2 - y3**2)],
                       [(r2**2 - r3**2) - (x2**3 - x3**2) - (y1**2 - y3**2)]])

    # or use np
    x_vect = np.linalg.inv(A_matr) @ b_vect

    # Position in 2d space
    (x, y) = x_vect[0], x_vect[1]

    return (x, y)


(x,y) = computePosition2DSpace(4,5,1)
print(x,y)
```