2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Assessing the capacity of man-portable UAVs for network access point localization, using RSSI link data

Sérgio Ferreira[1], Guilherme Carvalho[1], Filipe Ferreira[1] and João Sousa[1]

Abstract— **Field operations in remote locations or within hazardous scenarios benefit greatly from the ability to deploy general purpose networks that can be accessed not only by operation vehicles but also human participants. The existence of these networks allows the transmission of data vital to operation success that otherwise might not have the means to flow. Knowing how the network is distributed and its coverage is therefore invaluable to determine what are the operational limitations for data dissemination. The use of unmanned aerial vehicles (UAVs) to detect and locate said access points, and then relay that information back to the ground station, allows for the characterization of the network while maintaining the human element in a supervisory and tactical position. In this paper we assess the feasibility of using a UAV to locate network access points, through Received Signal Strength Indication (RSSI) data, relying solely on the on-board communication modem. We explore and present different localization methods, in order to determine a robust approach which can deal with GPS position error and range estimation errors. Moreover we articulate key challenges and lessons learned for real world application.**

I. Introduction

More and more we are witness to unprecedented technological developments in computation, control, navigation and communications. Said developments have empowered us with new and evermore efficient ways to tackle a plethora of operation scenarios from *Search and Rescue* and *Forest Fire Monitoring* to *Area Surveillance*. Moreover we can now make use of autonomous vehicles, such as unmanned aerial vehicles (UAV), to take the strain away from the human element in these scenarios, allowing them to position themselves way from danger and in a more tactical and coordinative position [1][2].

Nevertheless as operational teams become more complex and their tasks become more ambitions the initial logistics cost, whether it be time or funding, becomes greater. This means that an accurate prediction of vehicle and tool effectiveness becomes invaluable in order to make these endeavours as feasible as possible. One particular need for the applicability of said scenarios is the availability and coverage of wireless networks that might be present in the area. In order to exchange operational data, whether it is simple telemetry data or a live video feed, knowing the limitations of these networks can be a tremendous asset when determining and planning resource deployment and predict their effectiveness.

However the collection of this information isn't trivial if the access points have no way to announce their position or if their placing wasn't previously catalogued. In these situations the ability to use a UAV, already present in the operation and without increasing its load-out, to locate the network access points and relay that information back to the operation team could prove extremely desirable. By doing this we refrain from allocating human resources that are needed elsewhere while, at the same time, make use of the vehicles already present and deployed on the field. A practical example of the relevance and utility of network coverage predictability can be analysed in [3] "(...)UAVs are also assumed to have limited communication range, which means they cannot upload data to the base station unless they are within range of the station(...)". Having the ability of knowing the effective coverage greatly increases objective planning and execution.

[1]S. Ferreira, G. Carvalho, F. Ferreira and J. Sousa are with the Faculty of Engineering, University of Porto, Portugal {asbf,ee06156,filipef,jtasso}@fe.up.pt

II. Contextualization

Studies regarding UAV based localization systems aren't new, but the interest surrounding this topic has increased with use of multiple vehicles as a feasible approach [4]. Moreover RSSI centred solution still maintain interest even though RSSI measurements suffer from a great vulnerability to localization error, aggravated from noise input from UAV motion [5].

This research was conducted at the Underwater Systems and Technology Laboratory (LSTS) where we have been designing, building and operating a significant number of heterogeneous unmanned vehicles. These vehicles range from Remotely Operated Vehicles (ROV), Autonomous Underwater Vehicles (AUV), Autonomous Surface Vehicles (ASV), and UAVs. Furthermore we made extensive use of the LSTS's existing toolchain [6] for control and development comprised by the C4I (Command, Control, Communications, Computer and Intelligence) system Neptus [7], the vehicle task manager, control and navigation software DUNE (Dune Uniform Navigational Environment) and the IMC (Inter-Module Communication) communication protocol [8].

As we extend our portfolio in terms of multi-vehicle scenarios [9][10] we also expand our retinue by cooperating with institutions who's work can greatly benefit from UAV solutions specifically if these can be employed with as little cost as possible [11].

III. Problem Analysis

Our main purpose is to determine if a UAV can be used as a network access point detector, in an operation theatre, while making only use of its native communications' modem. By doing so we reduce the need for specific extra hardware that could complicate the UAV configuration setup while, at

the same time, increasing the cost of the proposed method. Moreover this hardware restriction will also allow this type of detection to be carried out by smaller sized, preferably man-portable, UAVs which have a reduced onboard load-out typically comprised of the autopilot, communications system and small sensors.

In order to determine the position of the access point we will first need to tackle two distinct challenges:

- Determining the antenna's position;
- Determining the antenna's communication range.

We then consider the desired outputs of our system to be the Antenna Position in global coordinates $P(lat, lon, height)$ and associated Communications Range Value $R(meters)$ to each of these points. A more high-level view of the desired system can be seen in Fig. 1. This will be the basis of what we will implement using the existing LSTS tool-chain.

A. Communications Range

Calculating the communications' range of an antenna constitutes one of our challenges, still most antenna manufacturers provide *sensitivity values* for the connection established by said antenna. The sensitivity of an electronic device such as an antenna is the minimum magnitude of input signal required to produce a specified output signal. Typically an antenna data sheet provides several sensitivity values for several maximum transmission rates with an associated tolerance value.

The problem becomes a matter of, as shown in Fig. 1, convert said sensitivity values to a distance value. Converting radio signals to distance based on transmitted power is a widely discussed and studied subject and thus our problem is a matter of choosing an already developed signal propagation model and implement it in algorithm form.

In pursuance of our goal we must then choose a model that will take into account our test conditions and our variables:

- **Free-Space** - The algorithm will always be used in outdoor environments;
- **Line-Of-Sight** - The vehicle will maintain line of sight with the antennas at all times;
- **High Gain Antennas** - Since high gain antennas are used in our communications, we need to choose a model that can assume a gain different than unity.

B. Antenna Position

Generally localization systems work making use of three separate points, with known positions, to locate a fourth unknown point via localization algorithms such as triangulation.

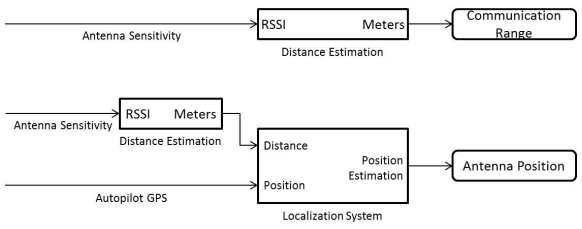

Fig. 1. Expanded version of the conceptual system's block diagram

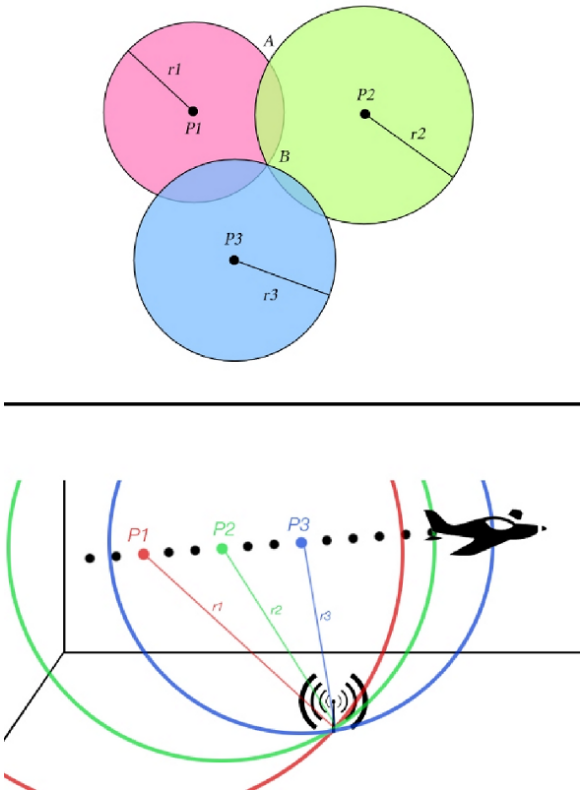

Fig. 2. Localization example using a UAV as reference node

As presented in Fig. 2, given the set of reference points $P1, P2$ and $P3$ and their distances to a target node B, we can define a circular shape with the reference point as the center of a circle and the distance as its radius. The solution is then the point that satisfies the following condition, with $P1 = (x_1, y_1)$ and its distance to target node being d_1, and so forth, with n being the number of reference nodes:

$$(x + x_i)^2 + (y + y_i)^2 = d_i^2, i = 1, 2, 3, ..., n \qquad (3)$$

The main difference in the present approach is that we are working upon the premiss that we do not have three readily catalogued points with known positions to make use normal localization systems. However, through the use of the UAV's autopilot GPS data, we can simulate an infinite number of reference points by acquiring distance estimation values throughout its trajectory as exemplified in Fig. 2 using the same nomenclature (3).

As Fig 1 shows we can define the reference points given by the autopilot's GPS as one of the inputs of our *Localization System*. The other input however will come the direct conversion of antenna sensitivity to distance. Nevertheless, selecting these reference points is far from trivial and will determine the success and mean error of the *Localization System*. We must take into consideration these factors when deciding the optimal points to input:

- **Collinearity and Co-planarity** - Reference nodes must not be collinear, when working two-dimensionally, or coplanar when working three-dimensionally;

- **Aircraft Attitude** - Speed, roll and direction will need to be tested to determine better suited reference nodes;
- **Distance Estimation Error** - Which nodes produce the less error when applying the localization algorithms.

We can clearly see that implementing a decision mechanism that chooses which reference nodes to use as inputs to our system, in order to minimize the localization error, is a decisive factor and will play a large role in the success of the whole system.

IV. Related Work

Before advancing with the approach for the intended solution we must tackle the details behind both the distance estimation from RSSI and the localization systems we will employ.

A. Signal Propagation

Received Signal Strength Indication (RSSI) is a measurement, in *dB*, of the power present in a received radio signal. Work has already been done that inquires the usability of RSSI as a *Distance Estimation* parameter, conducting practical experiments as to the variability of the signal and its conversion to a distance metric [12], as well as an analysis said variability [13]. Moreover a study presented in [14], regarding the characteristics of the RSSI signal, reached a very interesting conclusion regarding its behaviour finding no correlation between the changing of RSSI signals and sampling time stating also that *"(..)RSSI signals variance and its strength are not directly related to each other, but they are individually depended on the environment complexity.(...)"*. This allows us to not worry greatly on the frequency of data collection for the localization system, therefore reducing the on-board CPU use overhead.

One must now analyse a signal propagation model and, taking into account what was determined in I, we envision a situation where the receiver and the transmitter have a clear, unobstructed Line-of-Sight (LOS) path between them. We can then use the Free-Space Propagation Model (FSPM) as a mean to predict received signal strength [15].

The FSPM predicts that the received power decays as a function of the separation between transmitter-receiver pairs, given by the Friis' Free Space equation:

$$P_r(d) = \frac{P_t * G_t * G_r * \lambda^2}{(4\pi)^2 * d^2 * L} \tag{4}$$

P_t - Transmitted Power
$P_r(d)$ - Received Power
G_t - Transmitter Antenna Gain
G_r - Receiver Antenna Gain
d - Transmitter-Receiver Separation in meters *(m)*
L - System Loss Factor $L \geq 1$
λ - Wavelength in meters *(m)*

We can see in (4) that the received power falls off as the square of the distance, thus implying that the power decays with distance at a rate of 20 $dB/decade$.

The *path loss*, representing signal attenuation in dB, is the difference between the transmitter and receiver effective power and may include the antenna gains. The path loss for the model stated above, representing signal attenuation, is given by (5).

$$PL(dB) = 10log\frac{P_t}{P_r} = -10log\left[\frac{G_t * G_r * \lambda^2}{(4\pi)^2 * d^2}\right] \tag{5}$$

$$d_f = \frac{2D^2}{\lambda} \tag{6}$$

This model is only a valid predictor for values of d in the far-field of the transmitting antenna, also called the *Fraunhofer region*, which is given by (6) where D is the largest physical linear dimension of the antenna [15].

These formulas can help determine the distance based only on the RSSI values measured, but they are a major simplification, not taking into account several influencing effects regarding the propagation of the signal itself, and others regarding the hardware used to measure and read the RSSI values [16].

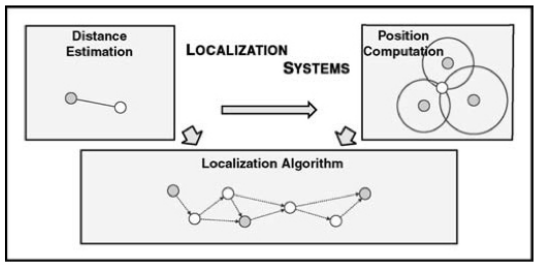

Fig. 3. Localization System [17]

B. Localization Systems

As seen in Fig. 3, a typical Localization System has three distinct components.

1) Distance Estimation: There are a plethora of methods, with different precision, that allow use to estimate distance but we determined RSSI as the Distance Estimator. This method derives the distance between the two nodes based on the strength of the signal received by one of the nodes. It has the advantage of requiring no additional hardware, but we must be wary of noise and interferences, as it's very sensitive and can produce large inaccuracies. The danger of channel noise and interferences can result in one or several of the following situations:

- **Uncertainty** - Various spheres might not intersect at a single location. This can happen due to noisy measurements;
- **Non-consistency** - A single node can have many reference neighbours. Any subgroup of them can locate this node by multi-lateration. The computed result can vary if different groups of references are chosen thus resulting in a non-consistency error;
- **Ambiguity** - A node reference creates a mirror through which the position can be reflected. This occurs very often under noisy ranging measurements or under poorly connected networks;

- **Error Propagation** - The errors from each step can propagate and accumulate.

2) Position Computation: When a target node, whose position we are trying to compute, acquires sufficient information about the distance between it and the reference nodes, it is then possible to compute its position using one of the methods presented in Table I.
Several methods can be used calculate a node's position, such as Tri-lateration, Multi-lateration, Triangulation, Bounding Boxes and several others. The choice of this method depends on available information and processing power.

TABLE I
VARIOUS POSITION COMPUTATION ALGORITHMS [17]

Method	# Refs	Complexity	Challenges
Tri-lateration	3	$O(1)$	Susceptible to inaccurate distances
Multi-lateration	$n \geq 3$	$O(n^3)$	CPU complexity
Bounding Box	2	$O(n)$	Final position error

a) Tri-lateration and Multi-lateration: Tri-lateration is the process of locating a precise point in space by measuring distance to several points. By drawing several spheres with each radii being the distance to the point we're trying to locate, on a number of spheres from three, it is possible to pinpoint the desired location.
For example, if the point we're trying to locate has coordinates $P(x, y, z)$ and the other points used have coordinates $B(x_i, y_i, z_i)$. The equation for every one of these spheres will be (7).

$$(x-x_i)^2+(y-y_i)^2+(z-z_i)^2 = r_i^2, i = (1, 2, 3, ..., n) \quad (7)$$

Unfortunately in real-world applications the distance estimation inaccuracies, as well as inaccuracies regarding the reference node's position, make it so that these circles don't intersect on a single point, resulting in an infinite set of solutions.
When a large number of reference points are available, we can use the Multi-lateration approach to compute the position estimation, resulting in an overdetermined system of equations - one where we have more equations than unknowns.

b) Bounding Boxes: Bounding Box is an alternative way of calculating the position based on the distance, proposed in [17] using squares to bound the possible location. For each reference node *i*, a bounding box is drawn as a square with the node as its centre with sides double the estimated distance, as seen in Fig. 4.
The intersection of all bounding boxes can then be easily computed without the need for floating point operations by taking the maximum of the low coordinates and the minimum of the high coordinates of all the bounding boxes.

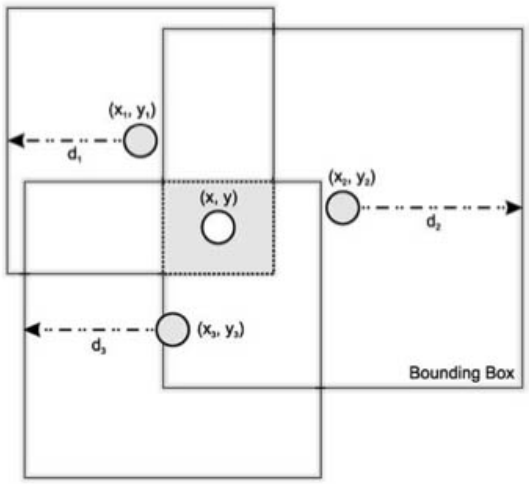

Fig. 4. Bounding Box Method [17]

C. Localization Algorithms

The localization algorithm is the main component of a localization system. It determines how the position and distance information will be manipulated in order to compute a valid position estimation for the target node. They can be classified into some categories [17]:

- **Distributed or Centralized** - The positions of the nodes can be computed in a distributed way by the network nodes (self-positioning) or by a single central node (e.g.: a more powerful node);
- **Positioning** - The computed positions can be absolute in terms of global coordinates (latitude, longitude) or relative to a node or point of the network;
- **Indoor or Outdoor** - If the system is used in indoors or outdoors scenarios;
- **One Hop or Multi Hop** - If all unknown nodes have direct communication with the beacon nodes (One Hop) or if a multiple hop communication is needed.

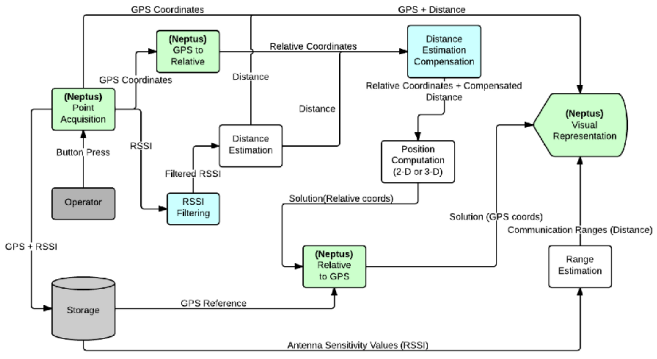

Fig. 5. Functional Architecture

V. Approach & Implementation

A. Architecture

As seen in Fig. 5 we have the functional architecture for our implemented module. We will now make a description of set of blocks briefly outlining their functions:

- Distance Estimation - Component dedicated to converting values input RSSI values to meters;
- Position Computation - Component dedicated to calculating node position, having relative coordinates and a distance as inputs.
- Range Estimation - Component similar to the Distance Estimation block except it has for inputs a set of RSSI values, returning a set of distances. These distances are the communication ranges for a given antenna;
- Localization Algorithm - Component dedicated to applying the desired algorithm, taking into account the input data;
- GPS to Relative - Component dedicated to the conversion from GPS absolute values to relative values. This function is particularly useful in order to convert the GPS position of the reference nodes to a position based on a Cartesian referential in order to input said positions into the Localization System;
- Relative to GPS - Component dedicated to the inverse operation of the GPS to Relative block.
- Visual Representation - Component dedicated to creating a visual representation on the Neptus geo-referenced map of the nodes.

B. Algorithms

Now we cover the algorithms behind the Distance Estimation, Position Computation, Localization Algorithm and Range Estimation blocks, and the theoretical foundation supporting them.

1) Distance Estimation: As stated before the model that better suits our needs for this particular body of work is the *Free-Space Propagation Model* since, in most cases, the antennas are placed at ground height, making the *2-Ray Model* unnecessary for our calculations.
Calculating an estimation of distance from the RSSI values becomes just a matter of feeding said values to (4). Converting said equation to the measurement units we are using we get (8):

$$RSSI(dB) = A + B - C \tag{8}$$

$$A = 20log_{10}(d)$$

$$B = 20log_{10}(f)$$

$$C = 27.55 + (\sum G - \sum L)$$

$$d(m) = 10^{(27.55 + RSSI - 20log_{10}(f) + (\sum G - \sum L))/20} \tag{9}$$

Observing (9), we notice that while the values of the receiver and transmitter antenna gain (G) are constant, and provided by the manufacturer of the antennas, the value of the Signal Losses (L) are not. Several factors come into play when calculating distance based on the *RSSI* value. If one takes into account that the *RSSI* measured while the vehicle is stationary is also highly unstable, the result is a very unpredictable system.
Therefore calibration is necessary to correctly account for this. We decided to calculate the sum of gains and losses using a reference distance, d_{ref}.

2) Position Computation: Now that we have an estimation of the distance between the UAV and the antennas connected to it, we can compute their position. Since we had various algorithms to choose from, we decided to implement three of them and then perform various performance tests to decide which one would be the most adequate. The methods chosen were: the *Least Squares* Method, the *Bounding Boxes* Method and the *Area of Intersection* Method and their implementation will be detailed in further sections.

a) Least Squares: This method's goal is to find a value of the error variance between the estimated values and the experimental values. Therefore, this requires us to linearise the circle equations, seen in (10), following the same linearisation method shown in [18], which uses the $j'th$ of (10) as the linearisation tool, producing (11).

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = r_i^2, i = (1, ..., n) \tag{10}$$

$$A = (x - x_j + x_j - x_i)^2$$

$$B = (y - y_j + y_j - y_i)^2$$

$$C = (z - z_j + z_j - z_i)^2$$

$$A + B + C = r_i^2, i = (1, 2, ..., j-1, j+1, ..., n) \tag{11}$$

Simplifying, with $r_j = r_i$ and d_{ij} we have produce (12):

$$A = (x - x_j)(x_i - xj)$$

$$B = (y - y_j)(y_i - yj)$$

$$A + B = \frac{1}{2}(r_j^2 - r_i^2 + d_{ij}^2) \tag{12}$$

It is irrelevant which equation is used as a linearization tool, so we'll arbitrarily choose $j = 1$, which is to say, we choose the first reference node and compare it to all the other reference nodes resulting in a linear system of equations (13) with $n - 1$ equations and two unknowns [19].

$$(x - x_1)(x_2 - x1) + (y - y_1)(y_2 - y1) = \frac{1}{2} * (r_1^2 - r_2^2 + d_{21}^2)$$

$$(x - x_1)(x_3 - x1) + (y - y_1)(y_3 - y1) = \frac{1}{2} * (r_1^2 - r_3^2 + d_{31}^2)$$

$$...$$

$$(x - x_1)(x_n - x1) + (y - y_1)(y_n - y1) = \frac{1}{2}(r_1^2 - r_n^2 + d_{n1}^2) \tag{13}$$

We can now write the problem in the $Ax = b$ form, and proceed to solve it using the *Least Squares Method*. Solving the *Least Squares* problem is a matter of applying the *Euclidean Norm*, which minimizes the sum of the squares

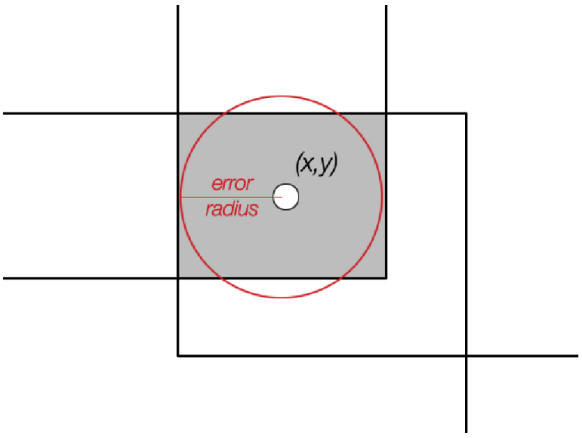

Fig. 6. Bounding Box Error Approximation

(14). In turn, this equals to solving the system described in (15).

$$Minimize x \ \epsilon \ \Re^n \ \|Ax + b\|^2 \tag{14}$$

$$x = (A^T A)^{-1} A^T b \tag{15}$$

This method was implemented in the module in both two-dimensional and three-dimensional variants.

b) Bounding Boxes: Implementing the bounding boxes solution is a matter of, instead of drawing a circle with radius equal to the distance to the target node (d_i), drawing a square with side equal to twice the distance to the target node $(2d_i)$. The intersection is then computed without the need for floating point operations as it is as simple as finding the maximum of the low coordinates and the minimum of the high coordinates.
The centre point of that rectangle will then be given by:

$$A = \frac{max(x_i - d_i) + min(x_i + di)}{2}$$

$$B = \frac{max(y_i - d_i) + min(y_i + di)}{2}$$

$$(\widehat{x}, \widehat{y}) = (A, B) \tag{16}$$

As seen in Fig. 6, the area of error would then be a circle of the same area as the rectangle of intersection, as a simplification, since every other method's error is displayed as the radius of a circle calculated as shown in (17):

$$A_{rectangle} = \pi * r_{error}^2 \Leftrightarrow r_{error} = \sqrt{\frac{A_{rectangle}}{\pi}} \tag{17}$$

This method was also implemented in 3-D, adding the Z coordinate without increasing the complexity, since it is just a matter of performing the exact same operations to the coordinate in the plane of zz.

c) Area of Intersection: This method is the classic Trilateration implementation, calculating the **Area of Intersection** between the 3 (or more) circles. On a first approach, we will calculate the intersection between two circles (18).

$$d = \sqrt{(x_{P1} - x_{P2})^2 + (y_{P1} - y_{P2})^2} \tag{18}$$

- If $d > r_0 + r_1$, there are no solutions since the circles are separate;
- If $d < |r_0 + r_1|$, there are no solutions because one circle is contained in the other;
- If $d = 0$ and $r_0 = r_1$ then there are an infinite number of solutions, since the circles coincide.

Looking once more at Fig. 7, considering the triangles $[P_0P_2P_3]$ and $[P_1P_2P_3]$, we can write (19) and using $d = a + b$ we can solve for a (20). We can then calculate h using (21).

$$a^2 + h^2 = r_0^2 \ and \ b^2 + h^2 = r_1^2 \tag{19}$$

$$a = \frac{r_0^2 - r_1^2 + d^2}{2d} \tag{20}$$

$$h = r_0^2 - a^2 \tag{21}$$

So, the centre point of the area of intersection is given by (22).

$$x_{P2} = x_{P0} + a(x_{P1} - x_{P0})/d$$

$$y_{P2} = y_{P0} + a(y_{P1} - y_{P0})/d \tag{22}$$

Now, by adding h to P_2, we can get the coordinates of P_3, the two intersection points (23).

$$x_{P3} = x_{P2} + -h * (y_{P1} - y_{P0})/d$$

$$y_{P3} = y_{P2} - +h * (x_{P1} - x_{P0})/d \tag{23}$$

From the deductions above we can see that, when the circles intersect on one single point, i.e: $d = r_0 + r_1$, we get $a = r_0 = r_1$ and $h = 0$. With a value of zero for h, P_3 is a single point, and is equal to P_2.
Iteratively, we calculate the various intersection points for each circle pair. After calculating those points, the difficulty lies in selecting the points. The easiest way to achieve this

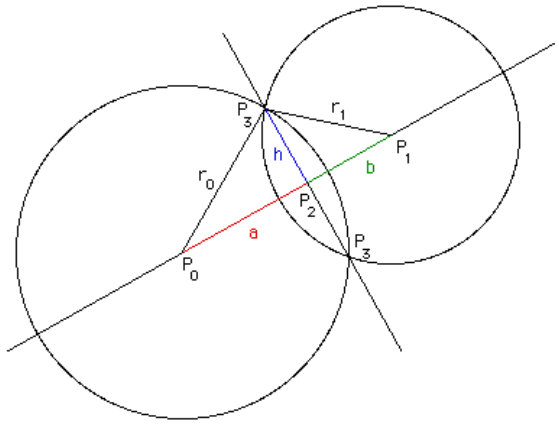

Fig. 7. Calculating the Area of Intersection of Two Circles

is to sort them according to their distance relative to each other and to select as many unique points as the number of the circles we have. To estimate the error margin, the area of the intersection, much like in the *Bounding Boxes* method, we will approximate it to a circle with radius r_{error}. To create the best approximation, r_{error} is in fact the average of the distance between the centre point and every intersection point. This method was only implemented in two dimensions due to the added complexity of calculating the volume created by the intersection of three spheres

3) Localization Algorithm: The localization algorithm is the code segment that will decide how the previous algorithms are used in order achieve a valid position estimation for our target nodes, the antennas. Previously we presented series of categories to classify *localization algorithms*. Following that reference, we can describe our algorithm as does apply Centralized Position Compuration using Absolute Positioning. Furthermore it is also an Outdoor One Hop implementation.

a) Distance Estimation Compensation: As we've established, RSS-based localization systems often offer erroneous values, resulting in inaccuracies in the calculated distance. Some steps were then taken in order to increase the algorithm's performance. As proposed in [20] we developed a small algorithm that, if the position computation does not succeed, increases the estimated distance of each reference node by 10%. In situations like the one in Fig. 8, it allows the computation of a position where it would not be possible.

b) RSSI Signal Filtering: RSSI signals always present a certain variance, depending on the antenna, environment conditions, etc. This, in turn, means that for the same position, two different measures can have different RSSI values.

In order to deal with this issue, we've decided to filter the RSSI values. We have implemented a **Moving Average Filter** - a finite impulse response filter - to provide us with a steady stream of RSSI values, while removing the outlier values.

We've decided to use **Brown's Simple Exponential Smoothing** (also known as Exponentially Weighted Moving Average) as a model because it is more responsive to changes on the recent past and better fitted for a rapidly updating system since, as we're trying to measure precise point values on a fast moving aircraft, the values quickly lose relevance as the plane continues its motion. For a series Y, it is calculated recursively (24).

$$S_1 = Y_1 \ , \ for \ t = 0$$

$$S_t = \alpha * Y_t + (1 - \alpha) * S_{t-1} \ , \ for \ t > 0 \qquad (24)$$

The α coefficient is a *"smoothing constant"* that can take values between 0 and 1.There is no formally correct procedure for choosing α, so we've decided to choose $\alpha = 2/(k + 1)$ and will further tune this procedure during testing.

c) Range Estimation: After pinpointing the antennas location with an acceptable degree of precision, the task befalls on estimating the range of that the UAV would have in the current environment. Using (9), but with predefined RSSI values, we can calculate the distance. A quick examination of the antenna's data sheet tells us the sensitivity of the antenna for various transmission speeds based on the used protocol. This will allow us to map not only the maximum range for the antenna by using (9), through minimum sensitivity value, but also to be able to represent regions based on the network's throughput. This is very important to not only missions where high data transmission rates are critical, such as missions using real-time video or large sensor data packets, but also missions where it is important to have fast communication between ground station and aircraft.

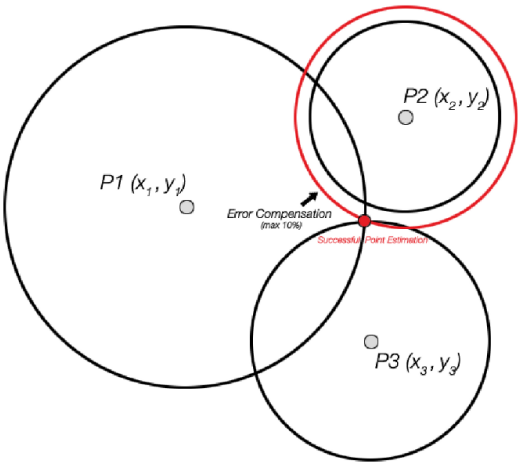

Fig. 8. Estimation Error Compensation Scenario

VI. Experimental Results

Before advancing to field tests with a simple UAV setup we've devised a series of simulations in order to give us a better knowledge of the localization algorithms developed and how they will respond to different situations regarding the target nodes we are trying to locate and the amount of reference nodes used. Furthermore, since cannot expect the inputs to be free of error in practical situations, so in order to try and simulate the unpredictability of the RSSI we've introduced a random factor into the distance prediction. This is also to simulate the unpredictability associated with the motion of the UAV and other errors associated with the Distance Estimation. Moreover we find errors not only on the RSSI values, but also on the position given by the GPS satellites. We've simulated this positioning error by adding a random value in meters to the X and Y coordinates of our reference nodes.

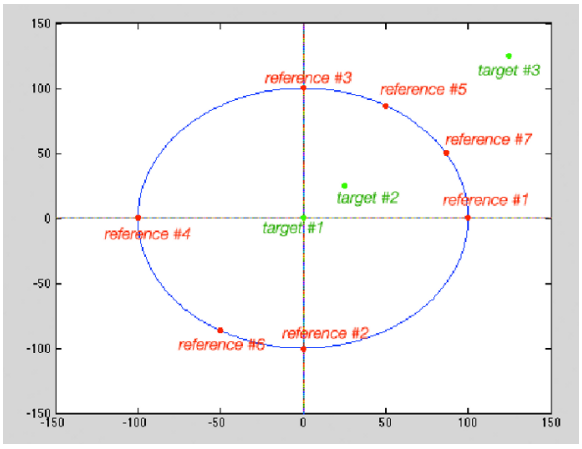

Fig. 9. Reference Nodes and Target Nodes used for Simulation

A. Simulation Tests

In Fig. 9 we can see the flight routine (the blue circle) and the various reference nodes used for testing and the target nodes we tried to locate, presenting several levels of difficulty: a perfectly centered target node, a node somewhere in the center of movement and a node located to one side of every reference node. The environment of our simulation is constituted by:

- A predefined trajectory from which to extract the reference nodes, simulating the path the UAV would make in the mission area;
- A target node, simulating the antenna;
- A flat, unobstructed area of $150x150$ meters.

We conducted localization tests of all three nodes with both Distance Estimation and GPS Positioning Errors. The number of reference nodes increased by one at each cycle of the simulation process, starting at 3 reference nodes, and ending at 7 reference nodes with 5 computations made for each set of reference nodes. In total, taking into account that tests were also executed to Distance Estimation and GPS Positioning Errors individually, we ran 225 simulations.

1) Target Node 1: This target node presents the best possible scenario, although a very unlikely situation it provide a good baseline for an initial analysis. As seen in Table II all algorithms had minimum error when using 7 reference nodes, thus indicating that more reference nodes introduced to the system will result in a lower error. We can see here that one algorithm outperforms the others when in severe conditions: the Bounding Box Algorithm. It is the least sensitive to error and kept its average error approximately constant from 4 reference nodes on.

2) Target Node 2: Target Node 2 is located inside the circle described by the flight routine, but not centred. This is a much more plausible simulation of real circumstances. It is worth saying that, while the GPS Position Error had the same influence on both simulations, we found that the error associated with Distance Estimation Error was marginally smaller on all situations. This might be due to the distribution of reference nodes, with the area around

TABLE II
AVERAGE DISTANCE ERROR, WITH ERROR IN POSITION AND DISTANCE ESTIMATION, TO NODE 1

	Number of Reference Nodes				
Method	3	4	5	6	7
Bound Box	49.39m	8.842m	9.601m	8.565m	6.731m
Circle Inter	8.523m	9.324m	10.84m	8.047m	8.330m
Least Square	9.114m	8.965m	12.14m	11.86m	8.402m

Target Node 2 being more densely populated with reference nodes than the area around Target 1. This leads us to believe that acquiring our reference nodes on the close vicinities of the target nodes might have a diminishing effect on the error of the location. We are presented with error around the same order, as seen in Table III, of magnitude as with Target Node 1 which leads us to the conclusion that, regardless of where inside the circle described by the flight routine the points are placed, the localization system behaves similarly.

TABLE III
AVERAGE DISTANCE ERROR, WITH ERROR IN POSITION AND DISTANCE ESTIMATION, TO NODE 2

	Number of Reference Nodes				
Method	3	4	5	6	7
Bound Box	26.00m	6.626m	5.665m	8.172m	10.91m
Circle Inter	7.960m	6.592m	7.390m	8.716m	7.006m
Least Square	9.114m	7.434m	8.089m	11.43m	7.184m

3) Target Node 3: Locating this node will present a challenge to our localization system. All our algorithms strive to locate one point or the **area of intersection**. In situations like the one we have on our hands, every reference node is located to the same side of our target node. Ideally, this should have no effect on our localization system, but when Distance Estimation or Positioning errors come into play, the circles will intersect in suboptimal ways.

TABLE IV
AVERAGE DISTANCE ERROR, WITH ERROR IN POSITION AND DISTANCE ESTIMATION, TO NODE 3

	Number of Reference Nodes				
Method	3	4	5	6	7
Bound Box	107.2m	101.2m	98.03m	101.3m	89.73m
Circle Inter	57.06m	49.62m	37.71m	37.40m	38.63m
Least Square	39.25m	33.89m	39.79m	44.24m	39.52m

As seen in Table IV we're immediately faced with very different results than we found in the previous simulations. Every algorithm struggled to compute a precise location, and the accuracy results had an error of, in the best scenario, three times as larger as in any previous situation.

B. Field Tests

In order to validate the simulations mad, the program was tested in real UAV mission scenarios. Integrating with the

existing LSTS toolchain, we strived to locate the antennas and predict a range estimation during operation. During these tests, antennas with GPS location were used in order to validate our location attempts and register the errors measured.
Unfortunately due to weather conditions only one field test was possible. Being that this research was developed during the autumn and winter, severe weather conditions were often present, further impeding the realization of more tests.

TABLE V
ERROR MEASUREMENTS FOR DIFFERENT POSITION COMPUTATION ALGORITHMS - FIELD TEST

Position Computation Algorithm	Error Calculated	Real Error
Bounding Box	3.88 m	8m
Circle Intersection	50.26 m	47m
Least Squares	5944.10 m	70m

We can see in Table V, there must be a problem with the Least Squares algorithm. It is very susceptible to imprecisions in the point acquisition and since we are using GPS, which is subject to an error, the Least Squares algorithm is very unreliable for estimating the antennas position.
Comparing the remaining two algorithms we can see that the **Bounding Box** was the top performer by far.

VII. LESSONS LEARNED

A. Distance Estimation

The estimation of a distance based on RSSI has been widely studied and documented as it is a very convenient means of attaining a distance without the need to resort to additional hardware. Some studies [12][14] conclude that it is not a reliable parameter for distance estimation. During our studies, however, we found that it can indeed be a usable indicator, but it requires some additional calibrations and/or training in order to be an accurate distance benchmark.
We also concluded, that the accuracy of this estimate will be one of the most important, if not the paramount, factor influencing the error of the localization system. As such, improvements done in this regard will directly benefit the performance of our program.
Improvements in the filtering section, such as an *Extended Kalman Filter* or *Unscented Kalman Filter* as proposed by [21] could help stabilize the signal and improve our distance estimates.
The location and positioning of the antennas in the UAV are also a concern since they are placed in a 90 angle with one facing the top of the aircraft and one pointing towards one of the sides, the antenna coverage will be different on either sides of the UAV. Changing the antenna placement could possibly yield much better Distance Estimation results.

B. Localization System

During the course of this dissertation we tested three position computation algorithms in order to calculate an estimated position of the Wi-Fi antennas used. As we've seen, of the three algorithms, only two of them performed with acceptable error margins. The Least Squares algorithm faced some severe performance problems and would require a rework, for example changing from an unconstrained least squares estimator to a non-convex constrained weighted least squares estimator, as used in [22].

VIII. CONCLUSIONS

We began work with the purpose of determining the possibility of using a UAV to detect a network access point using only its communication' modem. After breaking down the different parts needed to create a localization tool, and after analysing different localization algorithms, we determined that although the UAV-centred solution is still very susceptible to RSSI noise it is in fact possible for a UAV to give an adequate analysis of the networks it detects while in operation.

REFERENCES

[1] R. Gonçalves, S. Ferreira, J. Pinto, J. a. Sousa, and G. Gonçalves, "Authority sharing in mixed initiative control of multiple uninhabited aerial vehicles," in *Engineering Psychology and Cognitive Ergonomics*, D. Harris, Ed. Springer, 2011, vol. 6781, pp. 530–539.

[2] C. Fuchs, S. Ferreira, J. a. Sousa, and G. Gonçalves, "Adaptive Consoles for Supervisory Control of Multiple Unmanned Aerial Vehicles," in *Human-Computer Interaction. Interaction Modalities and Techniques*, ser. Lecture Notes in Computer Science, M. Kurosu, Ed. Springer Berlin Heidelberg, 2013, vol. 8007, pp. 678–687.

[3] D. W. Casbeer, R. Beard, T. McLain, S.-M. Li, and R. K. Mehra, "Forest fire monitoring with multiple small uavs," in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 3530–3535.

[4] E. W. Frew, C. Dixon, B. Argrow, and T. Brown, "Radio source localization by a cooperating uav team," 2005.

[5] N. Wagle and E. W. Frew, "A particle filter approach to wifi target localization," in *AIAA Guidance, Navigation, and Control Conference*, 2010, pp. 2287–2298.

[6] J. Pinto, P. Calado, J. Braga, and P. Dias, "Implementation of a control architecture for networked vehicle systems," *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV2012)*, vol. 270180, April 2012.

[7] P. S. Dias, G. Goncalves, R. Gomes, J. Sousa, J. Pinto, and F. L. Pereira, "Mission planning and specification in the neptus framework," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 3220–3225.

[8] R. Martins, P. S. Dias, E. R. B. Marques, J. Pinto, J. B. Sousa, and F. L. Pereira, "IMC: A communication protocol for networked vehicles and sensors," in *OCEANS 2009-EUROPE*. IEEE, May 2009, pp. 1–6.

[9] P. S. Dias, J. B. Sousa, and F. L. Pereira, "Networked operations (with neptus)," in *3rd annual Maritime Systems and Technologies conference. MAST*, 2008, pp. 12–14.

[10] J. Pinto, P. Dias, R. Martins, J. a. Fortuna, E. Marques, and J. a. Sousa, "The lsts toolchain for networked vehicle systems," *2013 MTS/IEEE OCEANS - Bergen*, pp. 1–9, June 2013.

[11] J. a. Fortuna, F. Ferreira, R. Gomes, S. Ferreira, and J. a. Sousa, "Using Low Cost Open Source UAVs for Marine Wild Life Monitoring - Field Report," in *2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems (2013)*, L. Rogelio, Ed. International Federation of Automatic Control, Nov. 2013, pp. 291–295.

[12] A. T. Parameswaran, M. I. Husain, and S. Upadhyaya, "Is rssi a reliable parameter in sensor localization algorithms: An experimental study," in *Field Failure Data Analysis Workshop (F2DA09)*, 2009.

[13] Z. Fang, Z. Zhao, D. Geng, Y. Xuan, L. Du, and X. Cui, "Rssi variability characterization and calibration method in wireless sensor network," in *Information and Automation (ICIA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1532–1537.

[14] R.-H. Wu, Y.-H. Lee, H.-W. Tseng, Y.-G. Jan, and M.-H. Chuang, "Study of characteristics of rssi signal," in *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*. IEEE, 2008, pp. 1–3.

[15] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.

[16] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, "Weighted centroid localization in zigbee-based sensor networks," in *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*. IEEE, 2007, pp. 1–6.

[17] A. Boukerche, *Algorithms and protocols for wireless sensor networks*. Wiley. com, 2008, vol. 62.

[18] W. Murphy and W. Hereman, "Determination of a position in three dimensions using trilateration and approximate distances," *Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado, MCS-95-07*, vol. 19, 1995.

[19] F. Reichenbach, A. Born, D. Timmermann, and R. Bill, "A distributed linear least squares method for precise localization with low complexity in wireless sensor networks," in *Distributed Computing in Sensor Systems*. Springer, 2006, pp. 514–528.

[20] P. De Cauwer, T. Van Overtveldt, J. Doggen, F. Van der Schueren, M. Weyn, and J. Bracke, "Study of rss-based localisation methods in wireless sensor networks," in *European Conference on the Use of Modern Information and Communication Technologies (ECUMIT)*, 2010.

[21] G. Mao, S. Drake, and B. D. Anderson, "Design of an extended kalman filter for uav localization," in *Information, Decision and Control, 2007. IDC'07*. IEEE, 2007, pp. 224–229.

[22] K. Cheung, H.-C. So, W.-K. Ma, and Y.-T. Chan, "Least squares algorithms for time-of-arrival-based mobile location," *Signal Processing, IEEE Transactions on*, vol. 52, no. 4, pp. 1121–1130, 2004.