ESTIMATION OF THE SPACIAL POSITION OF RADAR SENSORS MOUNTED IN A ROOM

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

21-01-2022 / 28-01-2022

CITATION

Pizzi, Matteo (2022): ESTIMATION OF THE SPACIAL POSITION OF RADAR SENSORS MOUNTED IN A ROOM. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.18857813.v1>

DOI

[10.36227/techrxiv.18857813.v1](https://doi.org/10.36227/techrxiv.18857813.v1)

ESTIMATION OF THE SPACIAL POSITION OF RADAR SENSORS MOUNTED IN A ROOM

ABSTRACT

This paper presents different methods used to allow a number of radar sensors mounted in a room to derive their relative spatial position as well as the floor plan of the room itself by relying on measurements of the distance of a person (walking around the perimeter of the room) from each sensor, at various time instants.

Although further work is needed in order to improve the accuracy of the algorithm, the obtained sensor locations reflect their real location in the room to a reasonable degree, indicating that the presented method works, and can be used as a basis for future development.

Possible future implementations to make the algorithm more precise and reliable are also presented in this paper.

INTRODUCTION

The CR&T project of the Next Generation Neural Interfaces Lab aims at developing new technologies to support people affected by dementia and to help them live a better life. Radar sensors have recently been developed by the lab in order to monitor the spatial position, breathing, and heart rate of people in a room. When mounted the first time, these sensors need to be able to automatically compute their position within the room and the floor plan of the room itself, without being manually pre-programmed each time. This paper provides the explanation of a possible algorithm for such a task.

GENERAL NOTE

Following are some general notes to take into account: First of all, the input to the algorithm is a .csv file containing a list of distances at each time instant from each one of the 4 sensors in the room. This algorithm relies on 4 sensors, yet an analogous procedure relying on 3 sensors can be implemented. On the other hand, using less than 3 sensors would involve further assumptions and processing not covered in this paper.

Before starting to compute the sensor positions, it is important to clean some of the possible errors contained in the measured data. This is done because the sensors detect all reflections of EM waves within the room, meaning that some of them might come from distant surfaces and might lead to wrong sensor positioning. These errors are to be ignored or filtered out. The filtering process happens both at the level of previous signal processing algorithms not covered in this paper, as well as throughout the algorithm itself.

Lastly, python is used as programming language with the following libraries:

- NumPy
- csv
- Math
- Matplotlib.PyPlot
- Pandas
- SciPy.optimize

SENSOR POSITIONING METHOD

Input File

The .csv file is read, ignoring all headings, and the 4 distance measurements are copied in 4 separate lists (one for each sensor), as well as in a single list for which each element is an array of 4 distance values at each time. This will help make the algorithm easier later on.

Filtering

The same pieces of information are copied in a data frame too. This allows to easily calculate a moving average of the distances in order to get smoothed out values and limit the measurement errors which could not be removed by previous processing methods. A window of 5 points has been used. This allows to remove most of the uncertainties while keeping measurement values reasonably close to the given ones (a large window implies altering the measurements too much, leading to sensor misplacement later on). The moving averages are copied to other lists to be used later. In addition, using a moving average with a window of 5 implies neglecting the first 4 measurements. This, however, is not an issue as there are many more measurements afterwards, and removing the first few does not influence the algorithm. Another issue due to the use of a moving average is that the final data points are slightly “shifted” with respect to the original ones (Fig. 1). This is because each measurement depends on the average of the previous 5. Suggested future work to limit this problem consists in “shifting backward” the moving average data points. This procedure results in better measurements and therefore more accurate sensor positioning.

Fig. 1
Shifting the moving average values backwards to correct the slight error introduced when calculating it.

Motion Analysis

After loading up all the data and smoothing it out using a moving average filter, all the measurements are analyzed by an algorithm to extract the specific indexes (in the list) at which the person is moving exactly towards (or away from) a sensor and when it is not moving at all. This is done because one of the methods used to compute the sensor positions relies on the algorithm knowing the value of the displacement (how much the person moves) at each step. When the subject moves exactly towards or away from a sensor, in fact, its displacement corresponds to the difference between two consecutive measurements by that

sensor, which is very easy to compute. In order to find those points, two processes are applied:

- First, the moments in which the person is not moving at all are found and ignored. This is done because using these measurements in the algorithm would introduce errors in the localization of the sensors. To find those points, the algorithm loops through the measurements of each sensor, and finds the moments in which the absolute difference from the mean of 5 consecutive measurements is close to 0 (below a chosen threshold), simultaneously, for each sensor. In other words, this allows to find the moments in which the distances measured by each sensor do not change (or change very little) for 5 consecutive measurements, for all sensors, at the same time. By applying this process throughout all the data points (each consecutive set of 5 measurements), the indexes at which the person is not moving are found and saved in an array (Fig. 2 - crosses).
- Second, the moments in which the person is moving exactly towards or away from a sensor are found among the points not included in the previous array. In order to find those points, the following assumptions are made:
 - As long as the person walks at a constant speed, going towards or away from a sensor implies that the variation of the distance with respect to time is linear. In other words, the points that are taken are the ones that follow a straight line on the distance vs time graph. To define this condition, the ratio of the maximum and minimum distance variation across 10 measurements is set to be smaller than 2.5 (value chosen based on different experimental tests).
 - The person walks towards or away from a sensor for at least 10 data points. This is done to prevent little and non-relevant movements to be included in the list.
 - Each step must be greater than 1.5 units. This is to avoid confusing small unwanted movements or measurement errors with actual walking steps, especially for the moments in which the person turns to change walking direction.
 - The measured average variation for the 10 measurements is the highest among all the sensors for a person that is going exactly towards or away from a sensor. This means that, if more than one sensor respects all previous conditions (1 to 4) at the same time, the points that are taken are the ones for which the average absolute difference from the mean (of those 10 measurements) is the highest.

The result of this calculation is represented in the following picture (Fig. 2 - dots). Those points can therefore be used in the process in order to find the sensor locations, and, for those selected measurements, the displacement of the person between any 2 consecutive points is given by the difference of the distances measured by that sensor. Those “straight line” indexes are saved in a 2D array in which

each row contains the number of the sensor being analyzed (1 to 4) at the first position (array[0][...]), and the consecutive indexes for which the “straight line” condition applies. Each row, therefore, contains information about one single sensor for which that condition is true.

Fig. 2
Dots represent the points when the variation is almost linear, crosses represent the points when the person is not moving

Step Value Calculation

Despite what has been said up to now appears reasonable theoretically, repetitive trials of the algorithm have shown that using a fixed constant value for the displacement (instead of taking the difference between the two consecutive distances every time, and for the “straight line” points only) leads to a better positioning of the sensors. This is probably because assuming the person walks at a constant speed, as the data is collected by the sensors at a fixed rate, the displacement (distance moved each time) will approximately be constant. The reason why this method produces better results is probably due to the fact that using a fixed value of displacement each time allows to include more points in the calculation (all of those for which the person is moving, instead of only the ones which respect the “straight line condition”). With this said, the aim of the algorithm then becomes estimating a reasonable value for the average displacement. Different ways to estimate such a value are presented below:

1. The first approach consists in looping through all measurements for each sensor and taking the maximum possible variation between two consecutive measurements. This method, however, is easily affected by errors, as the processed data still contain measurement inaccuracies, despite the use of the moving average filter earlier in the program (obtained value: 20.4).
2. The second approach consists in looping through all values for each sensor, calculating the 10 greatest differences between consecutive measurements for each sensor, taking their average, and then calculating the average among the 4 obtained averages (obtained value: 9.84).
3. Alternatively, the following third method can be considered: as the “step” value is approximately

constant over time (due to approximately constant walking speed), the variation between two consecutive distances measured by one sensor will be maximum when the direction of motion is along the radius of a circumference (as explained before). Because of this, the value for the step can be found as the maximum average displacement across a number of consecutive measurements which respect the “straight line” condition defined before. To find it, calculate the indexes for which the distance measurements of at least one sensor vary linearly (explained before), take the average of n consecutive displacements, and compare it with the averages calculated for that same range for the other sensors whose distances vary linearly (not necessarily all 4). When more sensors show linear variation, the greatest average is used, as it is more likely to be the proper displacement (this because the maximum variation in distance is theoretically obtained when the person walks towards or away from a sensor [Fig. 3]). At last, take the mean value of all the calculated averages (throughout all the data) (obtained value: 7.34).

The last approach, for all that was said before, should result in the best estimation for the step value. Despite this, however, the algorithm which is detecting the motion of the person towards or away from a sensor still needs to be improved. Because of this, the obtained step value is not ideal (obtained value: 7.34). A value of 12 will therefore be used in the final approach (see later), for now (derived by manually looking at the data points for which the person is moving in a straight line, towards a sensor).

Fig. 3
Greatest displacement measured for movements along the radius

Single Sensor Positioning Approach

The next step is to find the relative position of the sensors by considering each set of two consecutive measurements. The first location of the person is considered to be the point $(0, 0)$, while the second point will be at $(\text{step}, 0)$, with the “step” value being either a constant or the difference

between consecutive values, as mentioned before (the difference is still used in the tests that were carried out). In particular, this means that the walking direction will be the x-axis for that system of reference (Fig. 4). For each position, the distance measured by all sensors (which is the same as the distance of the person from the sensor), will be used to define four circumferences centered at that point $(0, 0)$. A second set of 4 circumferences is then defined for the following point $(0, \text{step})$, using the next set of distances from each sensor. As the sensors are not moving, the intersection of those circumferences will define the 2 possibilities for the positions of the sensors. In other words,

Fig. 4
Sensor positioning (only 1 out of 4) by circumference intersection.

instead of finding the position of the person as the intersection of circumferences centered at the sensors, the algorithm finds the position of the sensor as the intersection of circumferences centered at two consecutive positions of the person (Fig. 4).

The same process (for the same set of two points) is used to derive the location of all the other sensors.

This should ideally result in the same **relative** position of the sensors for **any** starting point. The only difference would be that, if all the results were plotted on the same graph, the points would appear to be repetitively shifted along the path of the person walking, and rotated towards the walking direction as the x-axis varies every time the person changes the direction of motion (this because, as said before, the origin is given by the first location of the person and the x-axis is defined in each system of reference by the direction of the displacement).

To calculate the sensor positions, different methods are presented in the following paragraphs.

In addition, the results are also two possible sensor positions. The third consecutive measurement cannot be used to discriminate between the two possibilities (at least not in the same way) because the walking direction with respect to the defined x-axis is not known (it might be the same or might change), and it is therefore not possible to write the equation of the third shifted circumference and find the intersection (shift along x and y are not known for certain).

Moreover, a general way of choosing the initial guesses when calculating the intersections is still to be defined. This

is important in order not to end up having only one solution as a result.

Future work should also include a piece of code to discriminate between the two possible solutions. It is suggested to first use all possible results and obtain multiple sensor locations, and then assume one of the 2 possibilities for the first sensor, and discriminate the other three as a result of that. The same should be done for the other sensor possibility, and the user should be given two "final outcomes" of sensor locations to choose from.

General Sensor Positioning Approach

As mentioned above, the resulting relative sensor positions, when plotted on the same graph, appear to be shifted and rotated, and this happens because the system of reference changes at each measurement following the path and moving direction of the person. An algorithm that shifts back and rotates the sensor positions is needed in order to obtain clusters of points indicating the general sensor positions and to calculate their average location.

To accomplish this, 5 different methods have been used to test different approaches and understand which yields the best results. A final method (the 6th) encapsulating the best procedure is then presented at the end. In general, the first three methods operate on all the sensor measurements, excluding the ones for which the person does not move. Moreover, the calculation of the sensor locations used a fixed value of 12 for the step parameter. The fourth procedure involved taking only the measurements which respected the "straight line" condition (defined above) and using a value for the step equal to the difference between 2 consecutive measurements of the sensor towards which (or away from which) the person was moving. Lastly, the fifth approach still involved only the measurements to which the "straight line" condition applied, yet the constant value of 12 was used for the step.

The 5 methods are more accurately presented here, and the outcomes of the processes are included below:

1. The first measurement of the sensor locations is taken as a basis, and each consecutive measurement (each set of 4 sensor locations) is shifted so as to overlap the first sensor of each measurement to the one of the first. The obtained sets of positions are then rotated around the first sensor (used as a pivot) until the positions of the second sensor are aligned. The average of the positions for each sensor is then taken (sensors 3 and 4 will appear as clusters of points). Despite resulting in a reasonable shape, this is not an ideal method as it amplifies the error on the other sensor locations measured. To limit this issue, the following methods will change the overlap point, as well as the method of rotation, in order to make the error more evenly distributed among all sensors and get a more reasonable positioning.
2. The second method, which mathematically more reasonable than the previous one, consists in shifting the points in order to overlap the average location of the 4 sensors. The obtained points are then rotated to align the first sensor, and the average of the resulting clusters is taken. This approach slightly limits the

error on the sensor locations and results in a better final shape.

3. The third method consists in shifting the sensors to overlap the intersection point of the diagonal axis between sensors 1-3 and 2-4. After the shifting, a double rotation is applied: the first one which aligns the first sensor (and consequently the third as they are on the same axis with respect to the pivot) in a clockwise direction, and the second one which follows a clockwise or anti-clockwise rotation and brings the sensors back by 1/2 the angle identified between the third sensors of consecutive measurements (therefore equal to the angle between the fourth sensors).
4. The fourth method exploits the shifting procedure of method 2, tries the double rotation process of method 3, and redefines the step value every time as the difference between two consecutive measurements by the "aligned sensor" (as explained above). This is possible because only the measurements which respect the "straight line" condition are taken into consideration. Lastly, this method has been tested with both the single and double rotation. It has been observed that the single rotation yields better results. This is probably due to the rotation algorithm which, in the case of "shift by average value" is not precise enough, and may lead to correcting one angle, while making the others considerably worse, therefore producing less accurate results. Suggested future work might involve the implementation of a method that equally splits the specific error among the four sensors, obtaining the best possible alignment and therefore minimizing the overall error, in turn producing better sensor locations.
5. The fifth method follows the same procedure as the fourth, yet the calculations are made keeping a fixed value of 12 for the step. This was done as a result of the observation (already mentioned before) that using a fixed constant value always yields better results.

The results of all the presented methods are included below:

METHOD 1

METHOD 2

METHOD 3

METHOD 4

METHOD 5

Data Correction methods

By comparing these different methods with the real locations of the sensors, it is clear that the fifth method yields the most accurate result in terms of relative positioning.

One big difference, however, lies in the scale of the axis. This leads to an excessively amplified error when plotting the path taken by the person as the intersection of three circles centered at the sensor positions.

Among all possible explanations for why this happens, the following is the most probable: the values of the measured distances used from the beginning present an error due to the way the sensors function. As the sensors detect EM wave reflections of the body, the closest plane encountered which provides a reflection will be the one from which the distance will be measured. This, however, never represents the center of the person itself, as the sensors might detect the chest, a shoulder, or the back of the person's body. Such an error is expected to give significantly different results because of the small room the sensors were placed in.

By using the given conversion factor 1 bin = 6.42mm approximately, an average shoulder to shoulder distance of approximately 100 bins can be obtained (shoulder to centre will therefore be around 50 bins). By comparing this value with the actual distance between sensors, which is in between 400 -1000 bins, it is clear that this can lead to a significant error, which might even be amplified by the processing algorithm itself.

Because of this, a further correction to the data is needed before computing the sensor positions. Such a correction implies using the given data to understand when the person is walking towards or away from a sensor (the same approach of determining the greatest-linear variation in distances can be exploited), and estimating a proper correction factor taken into account.

Three methods have been proposed for this:

- Adding a minimum "fixed" amount, corresponding to 1/2 the shoulder to center distance (approximately 25 bins), to each measurement. This method, however, would only correct part of the errors, and would not completely solve the issue. Some measurements would in fact become "excessively large" after this correction, and others would still present a small amount of error.
- Using a correction formula that varies the correction factor based on the variation in distance between consecutive measurements. This method, although too experimental, allows a certain variability in the correction. It is based on the fact already stated multiple times that, when a person is walking towards or away from a sensor, the variation in distance will be greater, and the correction to be included should therefore be smaller (chest to center of the body). On the other hand, smaller variations would imply the sensor is located on the side, and a greater correction should therefore be added (shoulder to center of the body). The proposed correction formula is the following:

$$40(1 - d/D) + 10$$

D represents the step distance (obtained when walking towards a sensor), and d represents the measured

variation in distance. When $d=D$, only the minimum correction factor will be added (10, equal to the chest to center distance). The maximum, on the other hand, is obtained when $d=0$, implying that the person has moved around the circumference, and its shoulder faces the sensor (shoulder to center distance needs to be added). All the other values in between are obtained for any other d between 0 and D.

A graphical representation of the three cases is represented below (Fig. 5).

This method, although more reasonable than the previous one, still presents different sources of error. The first one is that the distance to be added does not vary linearly, as it is assumed by the formula; and the second is that the measured data already contains other sources of errors, which might cause d to be greater than D, thus causing the final correction to be less than 10 or even negative in some cases, which is obviously impossible

Fig. 5
Second correction method

- The last correction method proposed is based on a geometric observation (Fig 6).

Fig. 6
Correction method based on direction of motion

By looking at the picture, it can be seen that the direction of motion with respect to the radius of the circumference (angle α) can be easily derived as $\arcsin(b/\text{step})$. The values of b can be obtained by knowing the spatial coordinates of the two position points and calculating the value $y_2 - y_1$. By considering a system of reference that has the x-axis parallel to the line “sensor-position 1”, then the first point will be determined as (distance_1, 0), and the second point will be defined by the intersection of the green circumference (centered at position 1, and having a radius = step) and the circumference centered at (0, 0), with radius = distance_2.

After determining the intersection points, the value of y_2 will easily be determined. In addition, due to the defined system of reference, y_1 will be = 0, and the value of b will therefore be determined as the absolute value of y_2 (with y_2 being either the first solution or the second solution value for y).

This procedure allows to determine the orientation of motion with respect to the sensor, yet a further step can be taken in order to correct the distances. Assuming the ground projection of a person's body can be approximated as a rectangle, the following cases can be defined (Fig. 7.a and 7.b):

Fig. 7.a
Correction model for $\alpha < 0.433 \text{ rad}$

Fig. 7.b
Correction model for $\alpha > 0.433 \text{ rad}$

As observed, two separate cases are defined, in order to accurately correct the distances. The borderline case corresponds to the D segment connecting the center of the body with the angle of the rectangle. That is the case in which the angle α corresponds to $\arctan(50/10)$, therefore equal to 0.433 rad . For the two cases, the correction distance D to be added respectively corresponds to $10/\cos(\alpha)$ for the first, and $50/\cos(\beta)$ for the second case, with β being equal to $\pi/2 - \alpha$.

The algorithm should loop through all the needed data (the ones for which the person is moving), perform the calculations, and add the value D to each measurement by each sensor.

Doing so, the following new values are obtained (Fig. 8). As observed, only the values for which the person is moving are changed, as only these will be used in the algorithm later on.

Such a process allows to obtain distance measurements that are more appropriate to the functioning of the algorithm.

Fig. 8
Corrected distance values (only the once for which the person is moving)

Obviously, this is only a geometrical approach to obtain better sensor positioning, and it still contains inaccuracies that should be resolved for an optimal sensor positioning. Future work should define a correction method based on the actual point measured by the sensor, which does not always correspond to the one considered (as explained above, it should be the point in the first plane encountered by the traveling EM wave).

Sensor Positioning After Data Correction

Once the correction of the data has taken place, the algorithm for calculating the sensor locations is run again. Considering the findings of the 5 test methods explained before, the following settings have been used for this procedure: constant step value of 12, shift by average sensor location, single rotation, and only the points respecting the “straight line” conditions are used (these have already been corrected, and will therefore implement the correction factor defined before).

With these settings, the following average sensor locations are found (Fig. 9.a).

By comparing the obtained results to the real sensor locations (Fig. 9.b), many thoughts can be expressed. First of all, the shape appears to be similar, indicating that the algorithm and the corrections made to the data work. It is evident, however, that the scaling of the two axis is still wrong, despite being slightly better than method 5. This means that the correction algorithm made sense and that better results can be obtained by improving it.

As mentioned above, the errors in the distance values are probably the primary reason why the sensor locations do not fully resemble the correct ones.

Fig. 9.a
Final algorithm sensor locations with corrected distances

Fig. 9.b
Real sensor locations

Further Result Processing

This paragraph introduces a possible future method that could be implemented once the final sensor location is obtained. It should be noted, however, that this method performs better for sensor locations closer to the real one (therefore when a better correction algorithm is developed as well), and can only be applied when more than 3 sensors are used.

The method is presented by using the “correct” sensor locations together with the corrected measurement data obtained by the algorithm. The calculated sensor positions are still too far from the actual ones in order to be used in this section.

The method involves a “reverse” procedure.

When an approximately reasonable sensor positioning is obtained, the path of the person in the room can be plotted by intersecting three different circumferences centered at three sensors. Different possible paths can be therefore visualized. In particular, using 4 sensors, 4 possible combinations are found.

Moreover, being the sensor locations only a “reasonable approximation” due to the measurement errors, the different paths plotted will likely be slightly wrong (not overlapping perfectly).

When this happens, a “shift” can be applied to the sensors on both axis (contracting or expanding the sensor locations along x and y) until the overlap between the obtained shapes is the greatest.

This process tries to increase the precision of the sensor positions starting from the obtained results and working back to their location, as represented in the last figure (Fig. 10).

Fig. 10
Plot of the different paths measured by three sensors.

In addition, it should be specified that, unlike what is seen in the representation, only one set of paths should be visible per measurement (1 of the 2 possibilities should be excluded in advance when selecting the sensor location, as mentioned before).

Final Thoughts

The presented algorithm and correction methods are to be thought of as examples and a basis for future work.

As mentioned in different parts, although the final obtained shape resembles the real one to a reasonable degree, there are still many corrections and implementations to be included in order to make the algorithm work better.

For this reason, it is still not possible to derive the floor plan of the room, as initially intended, yet future improvements to the different parts of the algorithm should increase its accuracy, reduce errors, and make it reliable enough to perform this task in real life.

ONLINE REFERENCES

- <https://www.imperial.ac.uk/next-generation-neural-interfaces/research/radar-biosensors/>
- <https://www.pythonpool.com/moving-average-python/>
- https://www.w3schools.com/python/matplotlib_plotting.asp

GITHUB CODE

<https://github.com/Teo56/Radar-sensor-positioning>