BLE-based Relative Positioning in the Context of Targeted Bike-to-Vehicle (B2V) Communication

JEROEN ZWIERS, University of Twente, The Netherlands

Communication between cyclists and drivers is often difficult in an urban environment. E-bell is an existing project that addresses the problem of communicating in these environments. This paper builds on this existing project by developing a way to do relative positioning of vehicles around a bike using Bluetooth Low Energy, enabling targeted communication between vehicles. It shows that it is possible to only use Bluetooth Low Energy to step away from broadcasting and send targeted messages to a group of specific vehicles.

Additional Key Words and Phrases: B2V (Bike-to-Vehicle) Communication, BLE (Bluetooth Low Energy), Relative Positioning, Low Energy, RSSI (Received Signal Strength Indicator), AoA (Angle of Arrival)

1 INTRODUCTION

Cycling as a means of commuting is becoming increasingly popular in many urban areas [4]. However, many cities around the world lack proper bicycle infrastructure, which makes cycling less attractive [13]. Cyclists often have to navigate through heavy traffic and deal with the risks of collisions with motorized vehicles. In addition, communication between cyclists and drivers is often difficult, especially in loud and noisy city environments. Existing forms of communication for cyclists, like hand signals, bike bells [5], or sometimes even shouting, can be ineffective.

The E-bell is an existing project that addresses the problem of communicating in these environments by providing direct communication between an e-bike and another vehicle using BLE. However, this system is still in development and only supports broadcasting. This is not a feasible way to approach communication in traffic, because broadcasting can cause confusion among road users. It is likely to decrease road safety because drivers are now distracted by messages that might not even be relevant to them.

1.1 Research Questions

The goal of this research is to develop a system that helps the E-bell project step away from broadcasting. To attempt this, a method for finding the relative position of vehicles using the RSSI and AoA of a BLE signal suitable for targeted B2V communication in an urban environment will be developed. This involves achieving enough accuracy of relative positioning to send targeted messages. To achieve this goal, the following research question will be asked:

- **RQ:** How to enable targeted B2V communication using only BLE?

To answer this RQ, the following sub-questions will be asked:

- **SQ1:** How can RSSI and AoA of a BLE signal be used to determine the relative position of vehicles?
- **SQ2:** How can the accuracy of the AoA and distance estimations be increased?

2 RELATED WORK

Vehicle-to-vehicle (V2V) communication is already a well-researched topic. There are many different methods for enabling communication between vehicles. For example, in the United States, dedicated short-range (DSRC) communication was developed as a standard for V2V communication by the automotive industry [6]. Although this method is robust, its power consumption is high compared to BLE, which is not ideal for B2V communication due to the limited battery life of e-bikes.

Using BLE for V2V communication is not a new concept either. In their paper, Bronzi (2014) researched single-hop and multi-hop methods for BLE communication between vehicles [2]. They found that a robust connection between two BLE-enabled devices can be achieved up to a distance of 50 meters. This range is sufficient for B2V communication.

However, many existing V2V communication systems that make use of BLE methods still use GPS for localization [3]. The accuracy of GPS depends on many factors [12]. And while there have been attempts to enhance GPS accuracy through V2V communication [1], this is inefficient in the context of the E-bell application for various reasons. First of all, there is no need for the absolute positioning that GPS provides, merely relative positioning between vehicles at an intersection. On top of that, this approach uses both GPS and V2V communication, whereas relative positioning is possible using only V2V communication. Using both GPS and V2V communication negatively impacts power consumption, which is an important aspect to consider in the context of e-bikes.

RSSI is a popular method for distance estimations in an indoor environment. This is because of the simple nature of this method. However, plain RSSI-based distance estimations are often inaccurate and fluctuate greatly. Therefore, a lot of research has been done on making RSSI-based distance estimations more accurate. One such method, proposed by Mehra (2013) uses a Recursive Least Squares (RLS) filter to get more accurate estimations over time [10]. Another filter that is used for noise reduction is the Kalman Filter [8].

This paper aims to create a system of relative positioning to reduce the target area from broadcasting without using GPS to do the positioning.

3 METHODOLOGY

In this section, the different methods of distance and angle estimation are discussed, along with how these were improved.

TScIT 39, July 7, 2023, Enschede, The Netherlands

3.1 Angle Estimation

The angle of arrival is directly measured from the antenna array of a BLE device. These estimations are then improved by filtering over time.

3.1.1 Missing Measurements. The first way the estimations are improved is by interpolating when measurements are missing. Occasionally a packet won't be received, and the angle cannot be estimated. In this scenario, the delta velocity is tracked from previous successful measurements and used to estimate the missing value.

```
deltaAngle = deltaAngle * 0.5 ...
    + 0.5 * (currentAngle - previousAngle);
deltaAngle = max(min(deltaAngle, 0.25), -0.25);
```

The delta angle is calculated for every successful measurement and is kept between -0.25 and 0.25 to prevent large error spikes from causing giant peaks. Then, whenever a measurement is missing, the new angle is calculated like this:

```
angle = previousAngle + deltaAngle;
```

3.1.2 Filtering. Additionally, a simple windowing function is applied to minimize the jumps and smooth them out over time. The mean of the last six measured angles is used.

3.2 Distance Estimation

The first, and most obvious method for estimating the distance is using the RSSI of the signal. However, in an outdoor environment, this method of estimating the distance is not reliable. Therefore, another way of estimating the distance is tested.

3.2.1 Angle-Based Distance Estimation. It is possible to estimate the distance using the estimated angle. This method assumes the vehicles are able to send their velocity and direction with the packet. As can be seen in figure 1, using two consecutive angle estimations, there is only a single point in 2d space around the bike where the given velocity and direction line up exactly with these 2 angles.

Calculating the position of the black dot is fairly straightforward, the x-position is exactly $\frac{v.x \tan b - v.y}{\tan a - \tan b}$, where a is the first angle, b is the following angle, and v is the velocity vector of the vehicle. To get the y-position, just multiply the x-position with $\tan b$.

3.2.2 Limitations. One limitation of this angle-based distance estimation is that it only works if consecutive angles have a big enough difference. With a smaller delta angle, any tiny error in this angle measurement greatly impacts the distance estimation. When the angle difference between two measurements becomes too small, it falls back to using the RSSI-based method to estimate distance. The chosen minimum angle difference is 0.0025, however, this is an arbitrary choice and not based on any testing.

3.3 Communication

The E-bell project already supports broadcast communication through BLE, the goal of this research is to limit this to target specific vehicles.

3.3.1 Targeting. There are two main ways of targeting a specific vehicle, either every vehicle has a (semi-)unique identifier, which can be established in the direction-finding packets. If this is a globally unique identifier there needs to be some central authority that distributes these identifiers. However, since there will not be millions of vehicles at the same intersection at the same time, generating sufficiently large, new identifiers every couple of minutes should give a very low chance of vehicles with the same identifier ending up at the same intersection at the same time. The second approach would be to use the actual angle estimation to target vehicles. This can be done by sending the estimated angle and some error margin with the message. Every vehicle receiving this message will then estimate the angle of arrival of the message, and can then infer whether the packet was meant for them. However, this second method will greatly reduce the accuracy of targeting, as the angle needs to be estimated twice.

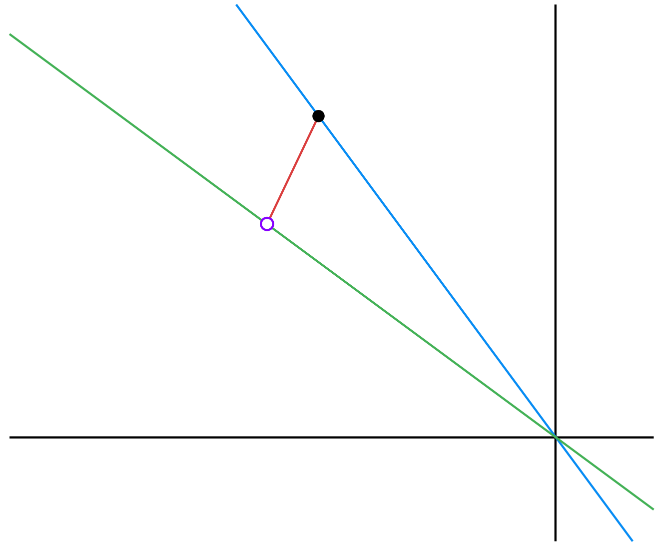

Fig. 1. Blue and green lines represent two consecutive angle estimations, the red line is the velocity and direction of the vehicle. The black dot is the only location on the blue line where the velocity vector perfectly points on the green line.

4 RESULTS

A simulation is used to develop and test these different methods. This allows for great control over the environment and setup, which makes comparing the methods easier and more repeatable.

4.1 Simulation Setup

The choice was made to use MATLAB to simulate the networking, figure 2 gives a basic overview of the setup. The simulation is setup up to focus on a single vehicle, assuming it travels in a straight line at a constant velocity. Everything is simulated relative to the bike, which is assumed to be at the origin. For all scenarios, a frequency of 10 Hz was chosen for the direction-finding packets. This means that every vehicle will send out a direction-finding packet every 100 milliseconds. This frequency was chosen because it was used in several other papers for localization [7, 11].

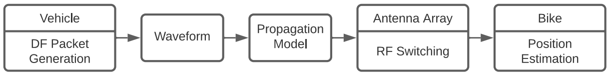

Fig. 2. Basic overview of the simulation setup.

4.1.1 Scenario Generation. The simulation was kept relatively simple, with only a single vehicle travelling at constant velocity and in a straight line. In many of the tests, the performance was measured over several generated scenarios. To generate a scenario first a starting location is chosen for the vehicle between 40 and 60 meters away from the bike. Then a direction of travel is generated in such a way it will come within at least 20 meters of the bike. Finally, a speed is chosen in such a way that the vehicle is able to travel at least 40 meters past the bike within 12 seconds. This duration was chosen because it results in speeds of roughly around 30 kilometers per hour.

4.1.2 Direction-Finding Packet. In order to estimate the angle of arrival of a signal, a specific packet is necessary. To generate this direction-finding packet, a helper function from a Bluetooth LE positioning example on the MATLAB website [9] is used.

```
data = helperBLEGenerateDFPDU(...
    "ConnectionlessCTE", ...  % DFPacketType
    160, ...                  % CTE Length
    [0;0], ...                % CTE Type
    2, ...                    % Payload Length
    '555551');                % CRC Init
```

This packet then needs to be converted to a waveform, MATLAB provides a function to generate the waveform from the generated packet, along with some additional information. This waveform is the signal that is sent by the sending antenna, and can then be manipulated to simulate signal propagation through air.

```
waveform = bleWaveformGenerator(data, ...
    'Mode',             "LE1M", ...
    'SamplesPerSymbol', 8, ...
    'ChannelIndex',     1, ...
    'DFPacketType',     "ConnectionlessCTE", ...
    'AccessAddress',    int2bit(19088743,32,false));
```

4.1.3 Propagation Channel. To simulate the propagation of the waveform a two-ray propagation channel is used. The two-ray propagation channel is a commonly used model for outdoor signal propagation simulation. It provides a simple yet effective representation of the propagation characteristics in open spaces. It takes into account both the line-of-sight path and the ground-reflected path, which are the dominant paths in outdoor environments. MATLAB provides a simple way to simulate a two-ray propagation channel.

The sample rate of the channel is the `SamplesPerSymbol` of the waveform, multiplied by the transfer speed of BLE, which is 1Mb/s. And the operating frequency is set to the default BLE frequency of 2.49GHz.

```
obj.channel = twoRayChannel( ...
    'SampleRate',         8*1e6, ...
    'OperatingFrequency', 2.49e9);
```

4.2 Angle Estimation

MATLAB provides a function to create an angle estimation config based on some criteria of the antenna array. For the simulation, an array size of 16 antennas was used. The choice of 16 antennas is based on the fact that more antennas mean better accuracy, and 16 is about the limit of what could fit inside a bike frame. The wavelength of a BLE signal is roughly 0.122 meters because BLE operates around 2.45GHz, and the antennas are spaced apart half the wavelength. This equates to a span of roughly a meter for 16 antennas.

```
cfg = bleAngleEstimateConfig( ...
    'ArraySize',        16, ...
    'ElementSpacing',   0.5, ...
    'SlotDuration',     2, ...
    'SwitchingPattern', 1:16);
```

The aforementioned MATLAB example[9] also supplies a helper function to perform antenna steering and switching on the signal.

```
waveform = helperBLESteerSwitchAntenna( ...
    waveform, ...
    angle, ...   % Real angle between vehicles.
    "LE1M", ...  % Mode
    8, ...       % Samples per symbol
    "ConnectionlessCTE", ... % Packet Type
    2, ...       % Payload Length
    cfg);
```

The resulting waveform, after having passed through the steer switching and the propagation channel, is then received by a BLE receiver. This results in IQ samples, which can be used to estimate the angle of arrival.

```
[~, ~, iqsamples] = bleIdealReceiver(obj.waveform, ...
    'Mode',             "LE1M",...
    'SamplesPerSymbol', 8, ...
    'ChannelIndex',     1, ...
    'DFPacketType',     "ConnectionlessCTE", ...
    'SlotDuration',     2);

angle = bleAngleEstimate(iqsamples, cfg);
```

4.2.1 Default Performance. The angle of arrival estimations proved to already be quite accurate by themselves without any filtering. The blue line in figure 3 shows the estimated angle without any filtering applied. In this scenario, missing measurements are also not accounted for yet, the previous successful angle estimation is used.

4.2.2 Filtering. The green line in figure 3 shows how the windowing function mentioned in section 3.1.2 smooths out those jumps. The estimations still fluctuate in other places, this is due to missing measurements.

4.2.3 Missing Measurements. Figure 4 shows the result when accounting for the missing measurements, as mentioned in section 3.1.1. The blue line shows that this interpolation at missing measurements smooths out the estimations, but amplifies the fluctuations at the start. This is a necessary trade-off to make the angle-based distance estimation work more smoothly. The green line shows

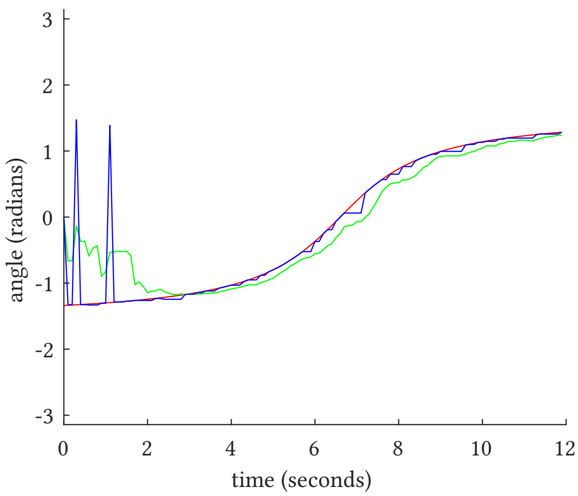

Fig. 3. The red line shows the actual angle, the blue line shows the estimated angle and the green line shows the estimated angle with a windowing function applied.

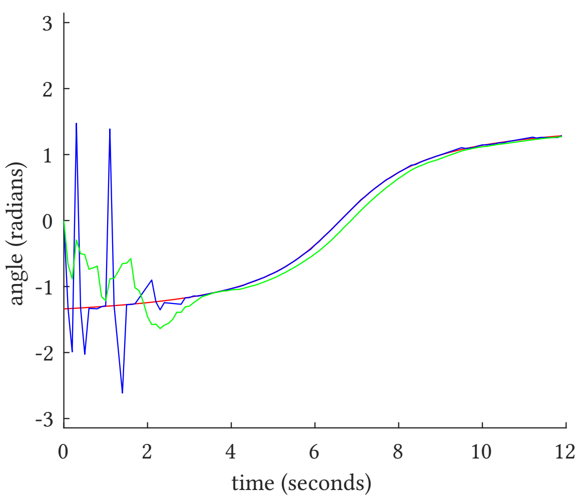

Fig. 4. The red line shows the actual angle and the blue line shows the estimated angle with interpolation at missing measurements.

that these fluctuations can once again be smoothed out using the windowing function.

4.2.4 Performance. Table 1 shows the average and highest error of the angle in radians taken over 50 different scenarios, generated as mentioned in section 4.1.1, for all the different methods. The last column shows both methods but with latency accounted for. In this case, the estimations have roughly 300 milliseconds of latency, which is due to the window filter explained in section 3.1.2. This shows that the filtering lowers the maximum error and the interpolating slightly improves the average accuracy.

	Direct	**Filter**	**Interp.**	**Both**	**Adjusted**
Mean (rad)	0.064	0.182	0.058	0.117	0.058
Max (rad)	2.801	1.357	3.657	1.521	1.545

Table 1. Mean and max error of the angle estimation in radians of several methods.

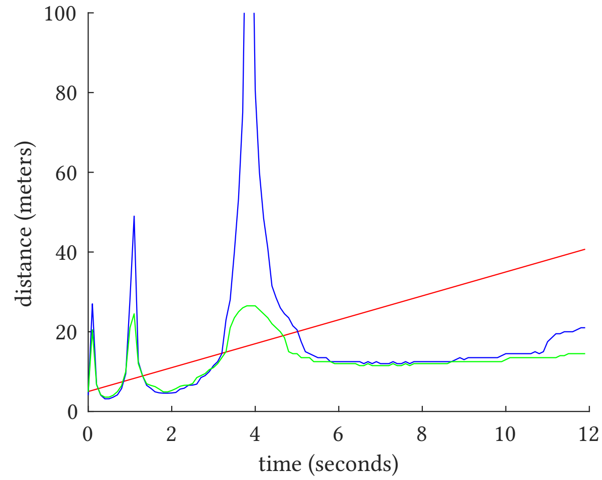

Fig. 5. The red line shows the actual distance, the blue line shows the estimated distance using the RSSI-based approach, and the green line shows that same approach with a Kalman filter applied.

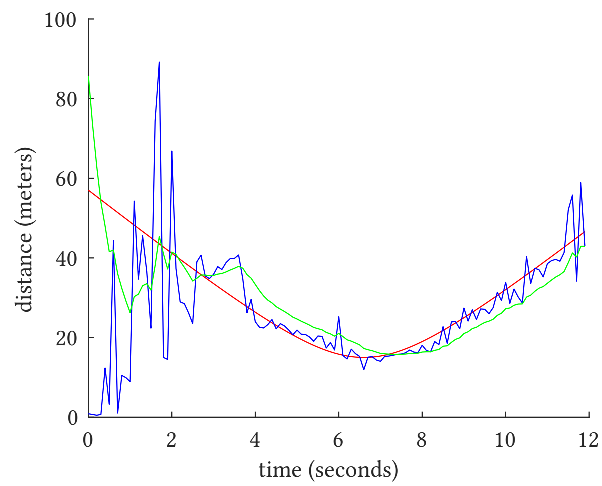

Fig. 6. The red line shows the actual distance, the blue line shows the estimated distance using the angle-based approach, and the green line shows that same method with filtering.

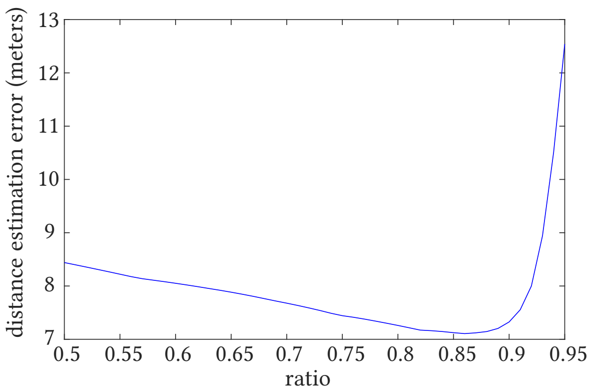

Fig. 7. The average distance estimation error for every ratio used in the moving average.

4.3 Distance Estimation

4.3.1 RSSI-based Estimations. The received signal power is directly calculated from the waveform after it has been passed through the propagation channel. It is calculated as the mean square of the waveform.

```
receivedPower = mean(abs(waveform).^2);
```

Using this signal power to estimate distance is inaccurate. In the two-ray propagation channel, the line-of-sight ray and ground-reflected ray constructively and destructively interfere, which means the received signal power oscillates with distance, instead of being a linear relation. Figure 5 shows a scenario where the vehicle is moving away from the bike, the interference is quite obviously visible in this figure. Even using a Kalman filter does not help, as can be seen by the green line. Several different values for the Q and R-value of the filter were tried, but none gave decent results. The values used in the graph are $Q = 0.9$ and $R = 0.1$.

4.3.2 Angle-based Estimations. The blue line in figure 6 shows the distance estimations using the angle-based method, as explained in section 3.2.1. Due to inaccuracies in the angle estimations, this value fluctuates greatly. The start of this figure fluctuates the most, which becomes obvious when looking at figure 4, where the angle estimations also fluctuate near the start.

4.3.3 Filtering. To remove the fluctuations from the angle-based distance estimations the estimations are filtered. Initially, a windowing function was used. However, after comparing this to a simple moving average, the moving average proved to give better results.

```
estimatedDistance = ...
    estimatedDistance * ratio + ...
    newEstimate * (1 - ratio);
```

Several values for this ratio were tried for several vehicle paths, as can be seen in figure 7, and a value of roughly 0.856 proved to give the best results. With this moving average, the estimations fluctuate less, but it does introduce some latency, which is visible when looking at the green line in figure 6.

	RSSI	Kalman	Angle	Adjusted
Mean (m)	18.33	16.15	7.54	6.95
Max (m)	135.59	33.52	36.98	39.06

Table 2. Mean and max error of the distance estimation in meters of several methods.

4.3.4 Performance. Table 2 shows the average and highest distance error in meters taken over the same 50 different scenarios as in table 1. Since the angle method uses the RSSI-based estimation with the Kalman filter as a fallback, the max value of the angle-based method is very similar to the RSSI-based method. However, the average error of the angle-based method is significantly less. Adjusting for latency caused by the moving average in the last column seems to improve average estimations, but because the estimations during the fallback to the RSSI-based method fluctuate so wildly, the highest error actually increases.

4.4 Communication

4.4.1 Positioning. The first step in targeting is looking at the achievable accuracy of positioning using the angle and distance estimations. The table below shows the position estimation error in meters for the same 50 scenarios that were used in previous tables. When adjusting for the roughly 300 milliseconds of latency caused by the filtering the position estimate is off by an average of 7.92 meters.

	Best	Latency-Adjusted
Mean (m)	9.66	7.92
Max (m)	93.74	61.62

Figure 8 shows that the accuracy of the position estimates decreases the further away the car is. This is expected, as the distance estimation relies on the delta angle, which will not be as big the further away the vehicle is. Unfortunately, it is impossible to filter out further distances to limit the range without knowing that distance more precisely.

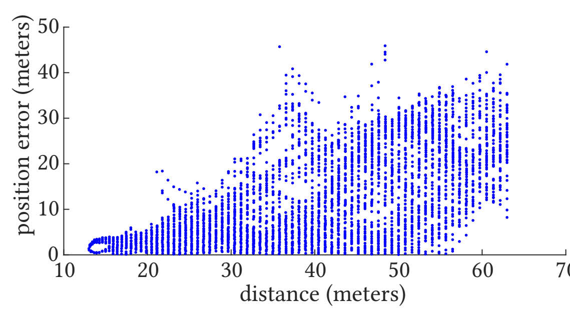

Fig. 8. Shows the position error in meters for every distance from the bike.

4.4.2 Targeting. Since cars at a bigger distance have such huge unpredictability in their estimated distance, to the point where a car at a distance of 50 meters might be estimated to be 20 meters from the bike, using this information to target vehicles at a specific distance is not feasible. It may be an option to target at a high

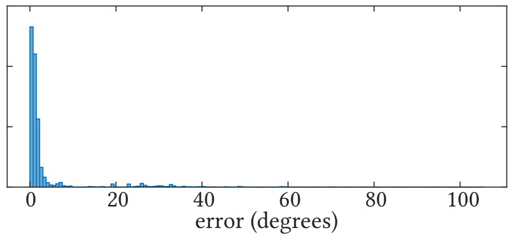

Fig. 9. Shows angle error density of latency-adjusted method.

	mean	90^{th}	95^{th}	99^{th}
Error (deg)	3.5	5.9	20.2	40.8
Reduce (%)	98	96.7	88.8	77.3

Table 3. Angle error in degrees for certain percentiles, and their respective target area reduction.

granularity, like 10 meters, but there is no guarantee that it will not include vehicles outside that range.

However, it is still possible to move away from broadcasting using just the estimated angle. In many cases, just being able to target vehicles at a specific angle is already a sufficient solution. The average error of the angle estimations using the latency-adjusted method is roughly 3.3 degrees. Figure 9 shows the error density in degrees from 50 different scenarios, the 90th percentile lies at only 5.8 degrees.

4.4.3 Performance. With the assumption that every vehicle can be targeted using a unique identifier, the angle-only approach for limiting broadcasting reduces the number of vehicles that can receive a message by about 98% on average. This number is based on the average estimation error of the angle, which is 3.5 degrees (table 1). This is the absolute error, so the actual range in which this lies is around 7 degrees, which is around 2% of a circle, reducing the targeted area by roughly 98%. Even when looking at high percentiles the target area is still greatly reduced, table 3 shows that the 99th percentile will decrease the target area by about 77.3%.

5 CONCLUSION

With the right antenna setup, estimating the angle of arrival is very simple, and by itself already pretty accurate. However, using the RSSI to estimate distance in an outdoor environment proved very inaccurate due to interference between the line-of-sight path and the ground-reflected path. Using the estimated angle of arrival to estimate the distance is also an option, but this works only in specific scenarios.

To improve the already accurate angle of arrival, the values are smoothly interpolated by keeping track of the delta angle to fill in missing measurements. Additionally, a windowing function is used to further smooth out the estimates, which does introduce some latency, but improves the accuracy of the angle-based distance estimations by removing fluctuations in the angle estimate. To improve the accuracy of the angle-based distance estimation a moving average is used. This smooths out the estimations and prevents sudden large errors. This method has to fall back to the RSSI-based distance estimate if the consecutive angle estimations are too small. To limit the extreme fluctuations caused by interference in the RSSI-based distance estimation, a Kalman filter is used.

The goal of this research is to find a method that helps step away from broadcasting. While the position estimates did not yield enough accuracy to target vehicles at specific positions, stepping away from broadcasting using only the estimated angle is still a possibility. Allowing on average a reduction of the target area of about 98%.

6 DISCUSSION

6.1 Limitations and Future Work

An emphasis was put on keeping the system simple but accurate, this proved to not be sufficient when estimating the distance of a vehicle from the bike. A more advanced network of communication between not only the vehicles and the bike but also among vehicles may offer additional accuracy in estimating relative positions.

At first, the use of additional information about the vehicles, like their velocity and direction of travel, was not deemed an option. However, after realizing the RSSI-based distance estimations would not result in sufficient accuracy, the choice was made to incorporate that information anyway to enable the angle-based distance estimation. With access to this information about the vehicles, the RSSI-based estimate could perhaps be improved as well. It may also unlock additional ways of estimating the distance.

Another big flaw in this research is the simplification of the simulation. Throughout the research, estimating the relative position was never quite accurate enough. For that reason, the simulation never moved away from just a single vehicle. That means the influence of interference from other vehicles was never considered. For future research, it may be beneficial to step away from simulations entirely and set up a network of physical BLE devices.

REFERENCES

[1] Abdul Hai Alami, Khaled M. F. Elsayed, Afif Al-Horr, Hassan Artail, and Jinhua Guo. 2018. Improving the Car GPS accuracy using V2V and V2I Communications. https://doi.org/10.1109/imcet.2018.8603032

[2] Walter Bronzi, Raphael Frank, German Castignani, and Thomas Engel. 2014. Bluetooth low energy for inter-vehicular communications. https://doi.org/10.1109/vnc.2014.7013351

[3] Ranran Ding and Qing-An Zeng. 2009. A clustering-based multi-channel Vehicle-to-Vehicle (V2V) communication system. https://doi.org/10.1109/icufn.2009.5174290

[4] Eco-Compteur. 2023. Cycling Data Tracker - Eco-Counter. https://www.eco-counter.com/cycling-data-tracker/

[5] Lisa Frohmann, Marian Weger, and Robert Höldrich. 2018. Recognizability and Perceived Urgency of Bicycle Bells. https://doi.org/10.21785/icad2018.025

[6] John B. Kenney. 2011. Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proc. IEEE* 99, 7 (2011), 1162–1182. https://doi.org/10.1109/JPROC.2011.2132790

[7] Ching Hong Lam, Kang Eun Jeon, Simon Wong, and James She. 2022. Distance Estimation Using BLE Beacon on Stationary and Mobile Objects. *IEEE Internet of Things Journal* 9, 7 (2022), 4928–4939. https://doi.org/10.1109/JIOT.2021.3120921

[8] Ching Hung Lam, Pai Chet Ng, and James She. 2018. Improved Distance Estimation with BLE Beacon Using Kalman Filter and SVM. https://doi.org/10.1109/icc.2018.8423010

[9] MATLAB. 2023. Bluetooth LE Positioning by Using Direction Finding. https://nl.mathworks.com/help/bluetooth/ug/bluetooth-le-based-positioning-using-direction-finding.html

[10] Rajesh Mehra and Abhishek K. Singh. 2013. Real time RSSI error reduction in distance estimation using RLS algorithm. https://doi.org/10.1109/iadcc.2013.6514305

[11] Michel Morold, Marek Bachmann, and Klaus David. 2022. Toward Context Awareness for Cooperative Vulnerable Road User Collision Avoidance: Incorporating Related Contextual Information. *IEEE Vehicular Technology Magazine* 17, 3 (2022), 75–83. https://doi.org/10.1109/MVT.2022.3173075

[12] Alan Oxley. 2017. *Uncertainties in GPS Positioning*. Academic Press.

[13] John Pucher, Jennifer Dill, and Susan L Handy. 2010. Infrastructure, programs, and policies to increase bicycling: An international review. *Preventive Medicine* 50 (1 2010), S106–S125. https://doi.org/10.1016/j.ypmed.2009.07.028