Multi-node bird detection simulation

02223 Model-Based Systems Engineering E21, Group 23

Siddigi, Shuhab
El-Haj Moussa, Ali Marwan
Fleming, Kevin Thor
Markevics, Eriks
Rasmussen, Emil Skov
Hesse, Kasper

ABSTRACT

Creating a product meant for continuous bird localization and species detection for a large area is still a problem today since there is no good hardware combination which can fulfill the needs to have long range, low power and high data rate transfer. This paper tackles the problem of how to best put together a system for solving the aforementioned problem. The main focus will be put on how to emulate the sound and choosing the right kind of equipment for the best case scenario while also interpreting how a battery is influenced depending on specific configurations set by the user. We found 2 possible solutions to the problem one being a low power draw option where calculations are performed centrally and one solution where the calculations are performed on the nodes themselves. We have also found, that if the systems main priority is the low cost, is possible to have a system with cheaper medium quality microphones, by grouping them. However, if the detection ratio of bird species is the main criteria, more expensive microphones could be used with equidistant placement. Future work will be revolved around optimizing the system and proving feasibility of self-sufficient solution with solar panels.

1. INTRODUCTION

Birds attacking crops is very detrimental to farmers, with a single county in California losing more than $40 million in revenue and taxes in one year due to crop losses [14]. Protecting crops from animals by being able to identify when and where animals congregate is therefore of great interest to farmers.

We believe that for the best crop control possible, a farmer should be able to quickly and easily see exactly where birds are present on the field, as well as which species of birds are present, as some birds are more dangerous to crops than others.

In this project, we have designed a simulator that can help to decide which hardware configuration is the best for such a system. The simulator is able to simulate multiple hardware configurations, tracking the number of birds detected, species of birds detected as well as battery consumption. In general the combination of hardware can be complex to design for an engineer which is why it makes sense to create an virtual simulation of the different components to test.

This will make it possible to not only construct an architecture for a detection product but also have a general use case to test different types of hardware in combination with complex battery models and protocols.

2. SYSTEM

2.1 Simulator design

The purpose of the simulator is to determine the ”best” hardware and software configuration for a bird detection and recognition system. By ”best”, we mean a system which is relatively accurate in detecting bird’s location and species and is as cheap as possible.

The simulator is thus designed to aid in answering the question of which combination of components makes for the best product.

For example, the choice of microphone has a large impact on the final product. Choosing a more expensive high-sensitivity microphone may make it possible to cover a given area with fewer microphones, while still achieving high detection precision. Less expensive microphones may on the other hand require more sensors for the same area.

When transmitting data from sensor stations to a central server, the choice of transmission protocol has a large impact on battery consumption. A high-speed protocol such as Wi-Fi will make it possible to perform real-time bird detection due to the high bandwidth, but also carries a high power consumption. On the other hand, while low-powered protocols like LoRa do not make it possible to perform real-time detection, a sensor station is able to operate longer on the same battery. In section 2.8, a number of communication protocols are presented, alongside with their respective benefits and disadvantages.

These examples just exemplify that the performance of a physical system is highly dependent on the choice of components. In the following sections, relevant background information and theory is given, before the implementation of the simulator is presented in section 3. In section 5, the experiments that have been run to gauge the performance of certain component combinations are shown.

2.2 Scope

The scope of the system is twofold: The system should both be able to accurately detect a bird's location, and also recognize the species of a given bird. Location detection will be performed via multilateration, as outlined in section 2.6, and species recognition will be performed with an AI, as presented in section 2.5.

For this purpose, the design domain being investigated is defined as follows: We consider a rectangular field sized $W \times H$ meters on which birds can land. A number of sensor modules are placed onto the field. We define a sensor module as a microprocessor, battery, one or more microphones and a wireless transmitter. Outside of the field is a central hub which will receive data from sensors, aggregate it and upload it to a server. The hub is connected to the power grid, and does not have any power consumption or processing limits.

When a bird chirps, it will emit a sound wave travelling over the field. We assume no obstructions and interference is present. By placing microphones on the field, it is possible to intercept these sound waves, process the sound data and detect the bird's location and species.

As the radius of the sound wave increases, the intensity of the call decreases with the square of the distance. In general, the relationship between the intensity I_2 at a distance d_2 from the source, and the intensity I_1 at a distance d_1 from the source is

$$\frac{I_2}{I_1} = \left(\frac{d_1}{d_2}\right)^2 \tag{1}$$

If the sound wave's radius increased such that it has passed a microphone, but no to a point where the intensity of the wave drops below the threshold value of the sensor, that sensor will receive the sound signal.

Depending on the network architecture, two ways of grouping sensors and microphones have been identified. These are presented in section 2.9

2.3 Microphone research

Microphones are a crucial part of the overall device construction, which ultimately deems it important that a proper investigation takes place. With this in mind, we began researching microphones of different kinds, both analog and digital, that each varied in specifications. During the research we came across microphones with various signal-to noise ratios, sensitivity ratings and power consumption values. As a baseline, the closer the sensitivity values is to 0dB, the better the microphone is at intercepting weak sound signals that may have originated far away. More information about microphone specifications is presented in section 2.4.

Table 10 found in appendix D showcases the microphones we deemed suitable for the device we wished to construct, and are representative of the low, medium and high sensitivity ranges respectively. The microphones found varies first of all in SNR values, with the lowest SNR value being 62 and the highest 80. Furthermore they also vary in price, with 5,47 DKK being the cheapest and between 23,44-31,30 the most expensive. This ensures low unit price in which is a positive step towards a cheaper viable configuration.

2.4 Primer on microphone specifications

Signal-to-noise ratio, often abbreviated SNR or S/N, is a measurement that compares the signal level to the background noise level. Defined by the ratio of signal power to the noise power, SNR is often measured in decibels (dB). Interpreting SNR values is pretty simple; having a ratio higher than one-to-one (1:1) signifies more signal than noise [9].

A microphone's sensitivity rating describes the output voltage for analog microphones, or the binary output value for digital microphones, when a microphone is subjected to 1 kHZ sine wave which registers at 94 dB 1 meter from the source.

Sensitivity for analog microphones is measured in decibels with respect to 1V, being logarithmic units of dBV. It indicates the ratio of the output voltage as a function of the Sound Pressure Level (SPL) The sensitivity can be denoted as linear units of mv/Pa or be expressed in a logarithmic fashion in decibels, where the sensitivity in dbV is given as.

$$Sensitivity_{dBV} = 20 \times log_{10}\left(\frac{Sensitivity_{mV/Pa}}{Output_{AREF}}\right) \tag{2}$$

Where $Output_{AREF}$ is the reference voltage, typically 1 volt/Pascal.

Digital microphone sensitivity is measured in decibels full-scale (dBFS). The sensitivity is the ratio of the binary output when subjected to the 94 dB sine wave, compared to the maximum output that the digital microphone can generate.

$$Sensitivity_{dFS} = 20 \times log_{10}\left(\frac{Sensitivity_{\%FS}}{Output_{DREF}}\right) \tag{3}$$

Where $Output_{DREF}$ is largest digital value that the microphone can output [13].

As seen above, analog and digital microphones each differ in terms of the units of measured output, and comparing them directly can therefore be inappropriate. A common factor is however the SNR mentioned earlier, which is directly comparable. The SNR-value listed on a microphone's specifations is the dB-ratio between the inherent noise in the

microphone and the signal generated when the microphone is subjected to the 94dB reference signal. The closer to 94dB the SNR value is, the less noise will be present in a signal. [13].

2.5 Detecting Bird Species with AI

BirdNET [11] is open-source software that is capable of identifying birds based on their call.

All versions of BirdNET consist of an AI that has been trained to identify bird sounds and report back it confidence that a certain bird species made the sound. The sound that BirdNET analyzes is represented digitally as a series of 2 dimensional spectrograms. This means that BirdNET is essentially performing image recognition.

The AI method that BirdNET uses is known as a convolutional neural network (CNN). BirdNET works by first downsampling and pooling the original image. This has the effect of allowing BirdNET to identify patterns within the image and give them weights. The classification layer of the CNN takes these patterns, mapping them to specific bird species with a certainty percentage, which is then given as the output. The weights on the CNN is determined by a learning algorithm applied to a set of bird recordings that have already been labeled with the corresponding bird species.

2.6 Multilateration

To simulate the process of finding the location of an bird a *Time-synchronous Measurements* technique called Multilateration was used. When a bird chirps in a field, a sound wave is emitted. This sound wave propagates outwards with constant velocity, the intensity decreasing with the square of the radius. By placing microphones at set points the field, the time difference of arrival (TDOA) can be calculated between all nodes.

When a soundsound wave expands, it will at time t_0 intersect with the first node and t_1 the second. When the TDOA between two nodes is known, a hyperbolic locus curve may be determined. The curve consists of all points where the sound wave may have originated. When multiple nodes intercept the same signal, the loci may be determined for pairs of nodes. The intersection point of these curves is the origin of the wave that triggered the nodes. An example is shown in fig. 1.

An open-source project implementing code for performing multilateration was used to generate fig. 1, and is also used to perform multilateration in the simulator [10]. It can be seen that the intersection of the 3 loci result in the exact location of the transmitter (bird) that initiated the signal. 1 locus can only show a range of distances between nodes and the source. 2 loci will potentially give 2 possible positions a transmitter can occupy, such as the green and blue curves in fig. 1. For accurate detection, it is then necessary to have 3 loci, meaning that at least 4 nodes must be used to get the position of bird.

2.7 Simulating power consumption

Estimating the power consumption of the sensor nodes when processing and transmitting data is done by finding the average current draw and time required for a given operation.

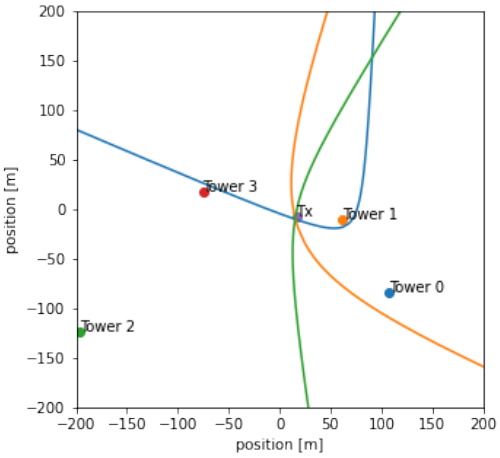

Figure 1: 3 loci (colored curves) and 4 towers. The intersection of the loci exactly lines up with the transmitter Tx. The small deviation is due to inherent measurement errors.

These values are then applied to a battery model which takes into account the current state of the battery to compute the battery discharge. In section 2.7.1, the battery model used is presented.

As the battery that is used in this project is modelled based on a 1.5V AA battery [7] there will be have to be some voltage conversion to run e.g. a Raspberry Pi using these batteries. This voltage conversion will not have 100% efficiency, while this is assumed in the simulator. This means that the dynamics of the simulation are accurate, but the exact numbers, such as how long the battery will last, may not be entirely correct.

2.7.1 Battery model

The battery is modelled as a step wise current dependent discharge [8]. The formula used is presented in eq. (4).

$$E_i = \frac{E_{i-1}}{1 + k \cdot I} - P \cdot t \tag{4}$$

Where E is the capacity in Joules, P is the power used in watts during the duration t seconds and I is the current drawn. The parameter k denotes the increased loss of capacity due to higher currents. In this way drawing a higher current results in less effective capacity than an energy equivalent lower current.
The value of k is dependent upon the characteristics of the battery; it is essentially a combined measure of battery chemistry. This parameter can be estimated from a rate discharge curve from the battery specification. The parameter will have to be recalculated for each computation of the capacity with different currents. The battery is modelled based on the AA L91 lithium battery [7].
The simulator uses a battery component which is composed of a configurable amount of cells each acting as described by eq. (4)

In addition the phenomenon of relaxation of the battery is

included in the model. This refers to regaining part of the spent capacity after a high current draw, when the current is subsequently lowered. This can be modelled by the eq. (5) [8].

$$E_i = g \cdot E_{i-1} \cdot (1 - e^{-\lambda \cdot t}) \tag{5}$$

Here, g refers to the growth ratio, that is the energy that can eventually be reached compared to when the battery began drawing a high current. λ refers to the recovery rate or the speed of the energy recovery. t is the time in seconds where the current draw has been sufficiently low after a period of high current draw.
Within the simulation there has not been a condition where the current was sufficiently low, so the relaxation has not been applied. This has been due to the fact that the processing unit of the sensor is always drawing current.

2.8 Communication Protocol Analysis

There are many types of communications protocols available to leverage, the problem is to find the best combination of power usage, data rate and range. Furthermore the architectural choice of having a centralized versus a decentralized system introduces a variety of options when looking at battery consumption as a whole. 5 protocols were evaluated as candidates to use in the system: WiFi, ZigBee, BLE, LoRa and NB-IoT.

A meeting with Charalampos Orfanidis at DTU Compute was set up to talk about the possibilities and limitations of the mentioned protocols. During that meeting, he presented table 1 comparing the protocols.

Protocol	Data rate	Range	Energy consumption
WiFi	High	High	High
Zigbee	Low	Medium	Low
BLE	Low	Low	Medium
LoRa	Very Low	Very High	Low
NB-IoT	Very Low	Very High	Low

Table 1: Summary table giving by Charalampos Orfanidis

Based on table 1, the best candidate for a physical system seemed to be ZigBee. It has low battery consumption and a good range, making it feasible for use on a large field. In [12], ZigBee is compared to two of the other protocols under consideration. A table from [12] is presented below as table 2

Info Type	Bluetooth	ZigBee	Wi-Fi
Sleeping	9 µA	12 µA	30 µA
Awake	35 mA	50 mA	245 mA
Transmitting	39 mA	52 mA	251 mA
Receiving	37 mA	54 mA	248 mA
Power Supply	3.3 V	3.3 V	5 V

Table 2: Power Consumption comparison for Bluetooth, ZigBee and Wi-Fi. From [12]

From table 2, it is clear that the power consumption of Wi-Fi is prohibitively high compared to that of ZigBee and BLE. Since the range of BLE is not long enough for our purposes, we chose ZigBee for communication protocol.

Calculating the total amount of power drawn during data transmission with ZigBee can be computed with eqs. (6) and (7), obtained from [1].

$$duration = Overhead + \frac{(8 \cdot packetsize)}{binaryrate} \tag{6}$$

Which for the Zigbee results in the duration

$$duration = 0.99 + \frac{(8 \cdot 116)}{250} \tag{7}$$

With this information it was now possible to calculate the accumulated battery consumption time period for a Zigbee.

2.9 Architecture

Based on whether a decentralized or centralized architecture is sought, different sensor placement algorithms may be beneficial. For this project, we have identified two placement algorithms of interest.

The equidistant algorithm works by placing N sensors at approximately equal distances on the field, such that each sensor covers approximately the same area. With this approach, each sensor module is equipped with one microphone. The microprocessor may either detect a bird's species locally, or it may transmit the recorded sound data to the hub. Since it is not possible to detect a bird's location with one microphone, location detection is performed by the hub. Performing species detection locally will require a more powerful microprocessor and draw more power, but will greatly reduce the amount of data that must be transferred to the hub.

The grouped approach works by placing $M \approx \frac{N}{4}$ sensors, still at approximately equal distances, but such that each sensor is equipped with 4 microphones. The microphones are mounted on the corners of a 4×4 meter frame. By attaching multiple microphones to a sensor module, it now becomes possible to also detect a bird's location locally. Species detection may be performed either locally or at the hub.

The two approaches reflect different kind of architectures. The equidistant is made with a centralized network in mind where each node will individually send their data to a central hub where data is processed. The grouped approach can either either work as a centralized network or as a decentralized since there are 4 microphones available for 1 processor, allowing species recognition and location detection to happen at each node.

3. IMPLEMENTATION

As previously outlined, the purpose of the simulator designed for this project is to compare multiple configurations for a bird detection product. For that reason, the simulator has been designed with modularity in mind, making it simple to test multiple configurations.

An object-oriented programming (OOP) approach has been taken to the simulator design, as OOP designs naturally lend themselves to modularity and easy configuration. A number of classes have been designed which each represent one part of the system being simulated. In appendix A, a UML class

diagram can be found. The class diagram shows all classes used in the simulator design, and their dependencies on one another.

The simulation flow is implemented as an event-driven simulation. When an object is created, it registers events in a central event queue, each event having a timestamp associated with it. The simulator then fetches the next event to be processed from the queue, and does this until no more events are present, signalling the end of simulation. A flowchart outlining the steps taken in a simulation flow can be seen in fig. 2.

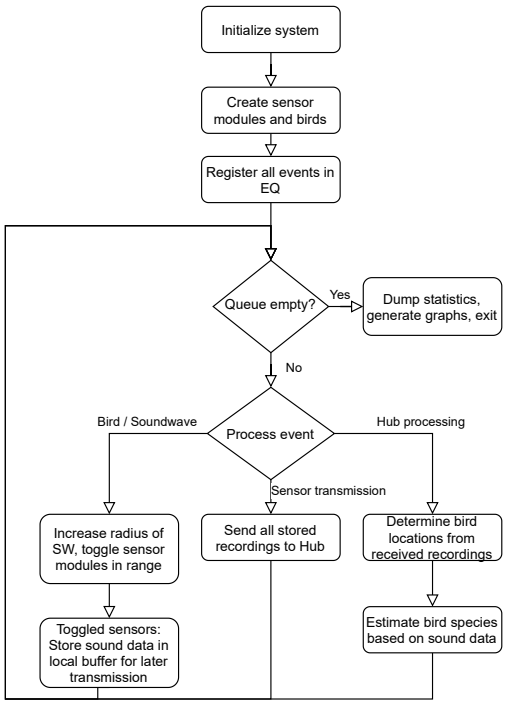

Figure 2: Flowchart presenting the overall simulation flow

The flowchart presented in fig. 2 is a high-level overview of the simulation flow. Not shown in the flowchart are some of the subprocesses used to evaluate a simulation's performance, such as the remaining battery of each sensor module being updated every time it is triggered by a sound wave (due to the microphone picking up a signal) or whenever it transfers data to the Hub (using the wireless transmission module).

3.1 Decentralized species detection

It was of great interest to us to determine whether it was feasible to detect bird species on each sensor module. To do so, we opted to run the Raspberry Pi-version of BirdNET on a Raspberry Pi, both to gauge power requirements and computational intensity. The hardware we selected was the Raspberry Pi 4b, as it was already owned by a group member, and the existence of a Pi-specific BirdNET implementatio, as we were unable to get BirdNET-Lite to run on the Raspberry Pi. After getting in contact with the developer of BirdNET-Pi, we were able to run BirdNET on a Raspberry Pi 4b. The experiments that were run are presented in section 4.1.

3.2 Battery model implementation

The functionality of the battery is calculated as described in 2.7.1. This is implemented in a *Battery* class that each *Sensor* class has as a variable. The Battery determines whether discharge or relaxation will occur. When discharging, it will perform the energy draw for a given task according to eq. (4). The energy drawn is based on the task being executed, as well as the background power consumption of the processing unit. An example of a battery graph for the sensors during a simulation can be seen in fig. 3, where 10 cells where used for each battery

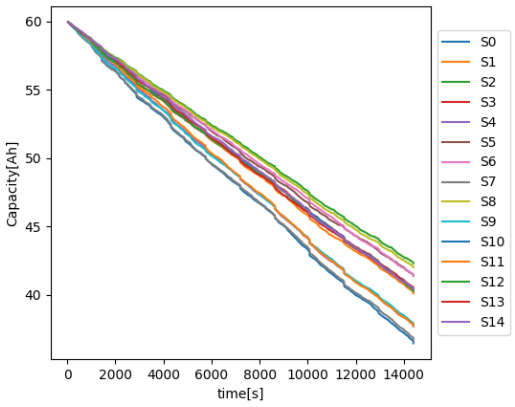

Figure 3: Battery capacity simulation for a set of sensors

Here time in seconds is the horizontal axis, capacity in Ah is the vertical axis and the colour of each line refers to each of the sensors shown to the right of the graph.
As can be seen, the consumption of the sensors differ, which is due to some sensors receiving more sounds that have to be transmitted.
The curve seems overall linear, which is due to the fact that the initial capacity is constructed by using multiple cells, whereby the current draw is shared between them. With a sufficiently high number of cells the current draw on the individual cell is very low and thus the the penalty for drawing high currents is barely in effect.

4. EXPERIMENTS

4.1 Feasibility experiment A - Architecture research

For the purpose of measuring power draw and performance of on-sensor calculations by the BirdNET-Pi application, we ran the software on the Raspberry Pi 4b. The following system configuration was used for the experiment:

RAM: 4 GB
Memory: 64 GB
Operating system: RaspiOS-ARM64

After the successful installation with the help of the BirdNET-Pi creator Patrick McGuire, the system was tested for the power consumption and processing speed.

The system continuously listens to the input audio-stream and analyzes it. Then it sends the information, such as amount of detected birds of each species to the web server every 5 seconds. It was measured that the system draws on average 0.9A with a 5V power supply.
The results found prove that a on-node processing is possible in real time and can be considered an option for the product development. This means that it is possible for the system to perform all calculations on the node itself. The only information that needs to be sent to the central server is then the end classification data left after the calculations which would be significantly smaller than sending the entire spectrogram.

4.2 Feasibility experiment B - Downsampling of audio

Low-power devices are limited in resources, be that in computing power, I/O or power supply if run on batteries. With this in mind, an idea was to investigate the possibility of on-node processing as mentioned in 4.1.

Another approach to reducing resource consumption which ties in with on-node processing, was to reduce the amount of data that needs to be processed and transferred. By downsampling audio signals, less data must be stored and transferred. By using compression, the amount of data to transfer is further reduced. When downsampling, some data is lost, but so long that the new sampling rate is not too low, the majority of the information in the original signal is preserved. A python script was created to downsample a number of sound files from Xeno Canto, a library of bird calls.

To test the effect of downsampling on species detection, 3 sets of 10 bird calls were downloaded from Xeno Canto. These were downsampled to various levels, and the resulting sound data was passed through BirdNET-Lite. The resulting confidence values were averaged for each bird species and sampling rate, the results of which can be seen in table 3.

Samplerate	Partridge	Pigeon	Starling
8 kHz	2%	29%	0%
12 kHz	55%	29%	9%
16 kHz	65%	28%	10%
32 kHz	66%	28%	23%
44 kHz	72%	26%	28%

Table 3: Samplerate, species of bird and confidence level based on 10 recording for each species

As is evidenced from table 3, once the sample rate drops below 12 kHz, the confidence level drops dramatically for the partridge and starling files. For pigeons, the confidence level stayed relatively constant. This is perhaps because pigeons do not have a lot of high-frequency components in their calls. So long that the new sampling rate is at or above 16 kHz, the confidence level does not drop by a lot for.

4.3 Simulator experiments

Using the simulator, a number of experiments have been run to determine the best system configuration. Experiments have been run on a field sized 150x150 meters. Simulation corresponds to a time period corresponding to 4 hours in real world. In appendix B, a table presenting the value of all possible configuration knobs can be found. In the simulations, the configuration values presented in table 4 were modified. The other configuration knobs kept their default values, as outlined in appendix B.

SPA	MS	NB	NM
Equidistant	High	50	5
Grouped	Medium	100	10
Random	Low	200	15
			20
			25

Table 4: Configuration values that been tested in the simulator.
SPA: Sensor Placement Algorithm, MS: Microphone Sensitivity, NB: Number of Birds, NS: Number of microphones

A total of 135 different configurations have been tested. Each of these configurations has been simulated 50 times, for a total of 1350 simulations. When processing the data, the average value over the 10 experiments with the same configuration has been used. The questions which we especially wished to answer with these simulations were the following:

1. How does the number of sensors affect the number of birds that are detected?
2. How does microphone sensitivity affect the number of birds that are detected?
3. What impact does the sensor placement algorithm have on the number of birds that are detected?
4. How does the number of birds affect battery life?
5. What is the best combination of placement algorithm, microphone sensitivity and number of sensors?

To answer these questions, the aforementioned simulations were run, and all data was dumped to a CSV file. The data was analyzed using Pivot Tables in Excel. In the following tables, a subset of the results are presented. The full data that was used to perform this analysis is attached in the file `stats.csv`, attached with the simulator's source code.

In general, using more microphones leads to a higher detection ratio (measured as the number of birds that were detected over the total number of birds present). Table 5 shows how detection ratio is affected by the number of microphones, grouped by the sensor placement algorithms used. From the table, it is evident that with a low number of microphones, the Grouped and Random placement algorithms perform the best. This is because the Equidistant algorithm is unable to cover the entire field with a limited number of microphones. Once enough microphones are available, the Equidistant placement performs as well as the Grouped algorithm.

Using high-sensitivity microphones seems to be a worthwhile investment, as the increase in detection ratio with high-sensitivity microphones ranges from a 1.7x increase to a 2.8x increase compared to using medium-sensitivity microphones. It is obvious that the difference in sensitivity between the medium- and high-sensitvity microphones ($-28\,\text{dbV}$ vs. $-22\,\text{dbV}$) corresponds to a large difference in the range at which they can pick up audio signals ($\sim 40\,\text{m}$ vs $\sim 80\,\text{m}$).

If only low or medium-sensitivity microphones are available, using the Grouped approach provides a higher detection ratio than the other placement algorithms. With low-sensitivity microphones, the detection ratio is still far too low to be useful. These results are shown in table 6.

Placement algorithm	**Number of microphones**				
	5(2)	10(3)	15(5)	20(5)	25(7)
Equidistant	2.64	23.52	40.36	46.19	57.01
Grouped	17.64	26.36	44.49	43.40	59.46
Random	6.16	23.99	35.30	43.49	49.21

Table 5: Percentage of birds detected with different numbers of microphones, grouped by placement algorithm. Grouped microphones are placed in clusters of 4 at one sensor. The value in brackets indicates the number of grouped sensor nodes with 4 microphones that were used

Placement algorithm	**Microphone sensitivity**		
	Low	Medium	High
Equidistant	0.00	32.51	69.33
Grouped	11.39	44.84	58.63
Random	0.91	29.78	64.20

Table 6: Percentage of birds detected with different microphone sensitivities, grouped by placement algorithm

The effects of sensor placement algorithm are also evident from tables 5 and 6. A low-cost implementation is best suited to using the grouped approach with medium-sensitivity microphones, whereas an implementation aiming to achieve as high a detection ratio as possible is best suited using high-sensitivity microphones with either grouped or equidistant sensors. Low-sensitivity microphones do not provide adequate detection ratios with any of the placement algorithms.

The number of birds present on the field has a relatively small impact on battery life. The more bird calls a sensor module intercepts, the more data it will have to transmit to the central hub.

It is assumed that each bird call corresponds to 30 seconds of audio data, sampled at 44 kHz at a bit depth of 64 bits, for a file size of $30s \cdot 44000 \frac{samples}{second} \cdot 64 \frac{bits}{sample} = 8448000 bits = 1.05 MB$. The simulation does not take into account that the audio signals may be downsampled, and that a compression algorithm may be used to reduce the amount of data to transfer. Therefore, the bit depth and file length have been set to an unrealistically high value on purpose, such that the calculated power consumption for transmission is a definite upper limit. Table 7 shows the average remaining battery capacity after running simulations. It is clear that the number of birds, and thus number of transmissions that each sensor module must perform, only has a very small impact on the total power consumption. Since the values used for simulation are set higher than real-life values on purpose, the actual impact on battery life may very well be lower.

Number of birds	**Percent battery remaining**
50	80.05
100	77.54
200	75.31

Table 7: Remaining battery percentage after 4 hours, grouped by number of birds placed on the field

Finally, in appendix C is shown a table presenting detection ratios for all combinations of microphone sensitivities, sensor placement algorithms and number of microphones. From the table, it is evident that the highest detection ratios occur when using high-sensitivity microphones. It is possible to achieve an 87% detection ratio using only 10 high-sensitivity microphones and equidistant placement. Using the grouped placement algorithm, it is possible to achieve a 90% detection ratio using 7 sensor modules, each equipped with 4 high-sensitivity microphones.

5. RESULTS AND CONCLUSION

In this paper we have designed a simulation to find the best hardware configuration for a bird detection system consisting of multiple nodes and a central hub. The experiments described in the paper were designed to: Verify feasibility of specific architectural solutions, find optimal physical placement of nodes, as well as finding the most optimal hardware configurations for the project. In this section you, the results of these experiments are presented, together with some reasoning reasoning about the results and future work discussion.

5.1 Conclusion

Starting with architecture, the analysis and experiments have shown two possible architectures:

1. Low power nodes transmitting the data to the central hub, where the data processing would happen. This corresponds to the equidistant approach presented in section 2.9, where all data is transmitted to the hub.
2. Our architecture research has proven the feasibility of a distributed architecture, where nodes perform all processing and only send the result back to the central hub.

Looking at the hardware configuration, experiments on audio down-sampling (Experiment B) has shown that BirdNET still performs well with a down sampled audio input. This means that the solution could be further optimized to achieve longer battery life and cheaper hardware configurations.

Furthermore the simulator experiment results have shown the optimal placement of listening nodes based on end user

preferences. If the end user prefers a cheaper solution, the best approach is to use medium quality microphones where the sensor nodes are grouped. While the best detection ratio is achieved with premium microphones, that are placed equidistantly.

In addition, as was shown in table 7, the nodes have sufficient capacity to last for 4 hours when consisting of 20 cells even when 200 birds are present. As was shown in fig. 3 even with 10 cells, the discharge curve is close to linear, so battery with 20 cells is also very close to linear. This means that for example for 50 birds, the battery with 20 cells can be estimated to last about 20 hours

The conclusions drawn from this paper can help the reader with the implementation of the bird detecting system.

5.2 Future work

The following subsection discuss parts of the project where further work is possible.

Feasibility experiment A has shown that the power consumption for on-node processing is relatively small, which opens up the question: Is a fully standalone solution using solar panels to replenish batteries feasible?

The other topic in need of further investigation is on node processing optimization. The experiment was performed using sub-optimal hardware, since its only goal was to prove the feasibility of on-node processing.

It should be possible to optimize the hardware, reducing power consumption, and still high detection ratio and location detection. Experiment B has also shown that downsampling the input sound does not affect precision greatly, allowing for even further optimization.

Finally, the GUI used for simulation experiment could be improved to include real time parameter change, since it currently only runs with a preset configurations.

6. REFERENCES

[1] E. Casilari, J. M. Cano-García, and G. Campos-Garrido. Modeling of current consumption in 802.15.4/ZigBee sensor motes. *Sensors*, 10(6):5443–5468, 2010.

[2] Digi-Key. Spu0410lr5h-qb, mar 2013. https://www.digikey.dk/da/products/detail/knowles/SPU0410LR5H-QB/2420983.

[3] Digi-Key. Inmp404, may 2014. https://www.digikey.dk/en/products/detail/tdk-invensense/INMP404ACEZ-R7/2606578.

[4] Digi-Key. Aom-5024l-hd-r, jan 2017. https://www.digikey.dk/en/products/detail/pui-audio-inc/AOM-5024L-HD-R/7898328.

[5] Digi-Key. Pom-2730l-hd-r, jan 2017. https://www.digikey.dk/en/products/detail/pui-audio-inc/POM-2730L-HD-R/7898330.

[6] Digi-Key. Mp34dt05tr-a, jun 2021. https://www.digikey.dk/da/products/detail/stmicroelectronics/MP34DT05TR-A/8535500.

[7] Energizer. Energizer l91 datasheet, December 2021. https://data.energizer.com/pdfs/l91.pdf.

[8] C. G., B. J., P. M., Z. L., and S. B. *Sense: A Wireless Sensor Network Simulator*. Springer, , Boston, MA, USA, 2005.

[9] Invensense. Signal-to-noise ratio, oct 2021. https://invensense.tdk.com/wp-content/uploads/2015/02/AN-1112-v1.1.pdf.

[10] M. Jurasovic. Multilateration, dec 2017. https://github.com/jurasofish/multilateration.

[11] S. Kahl. *Identifying Birds by Sound: Large-scale Acoustic Event Recognition for Avian Activity Monitoring*. PhD thesis, Chemnitz University of Technology, 2019.

[12] O. O. Kazeem, L. O. Kehinde, O. O. Akintade, and L. O. Kehinde. Comparative Study of Communication Interfaces for Sensors and Actuators in the Cloud of Internet of Things. *International Journal of Internet of Things*, 2017(1):9–13, 2017.

[13] J. Lewis. Understanding microphone sensitivity, may 2012. https://www.analog.com/en/analog-dialogue/articles/understanding-microphone-sensitivity.html.

[14] A. E.-A. S. Soliman Desoky. Damage Caused By Birds and Rodent in Field Crops and Their Control. *Journal of Global Innovations in Agricultural and Social Sciences*, 2(4):169–170, 2014.

APPENDIX

A. UML DIAGRAM

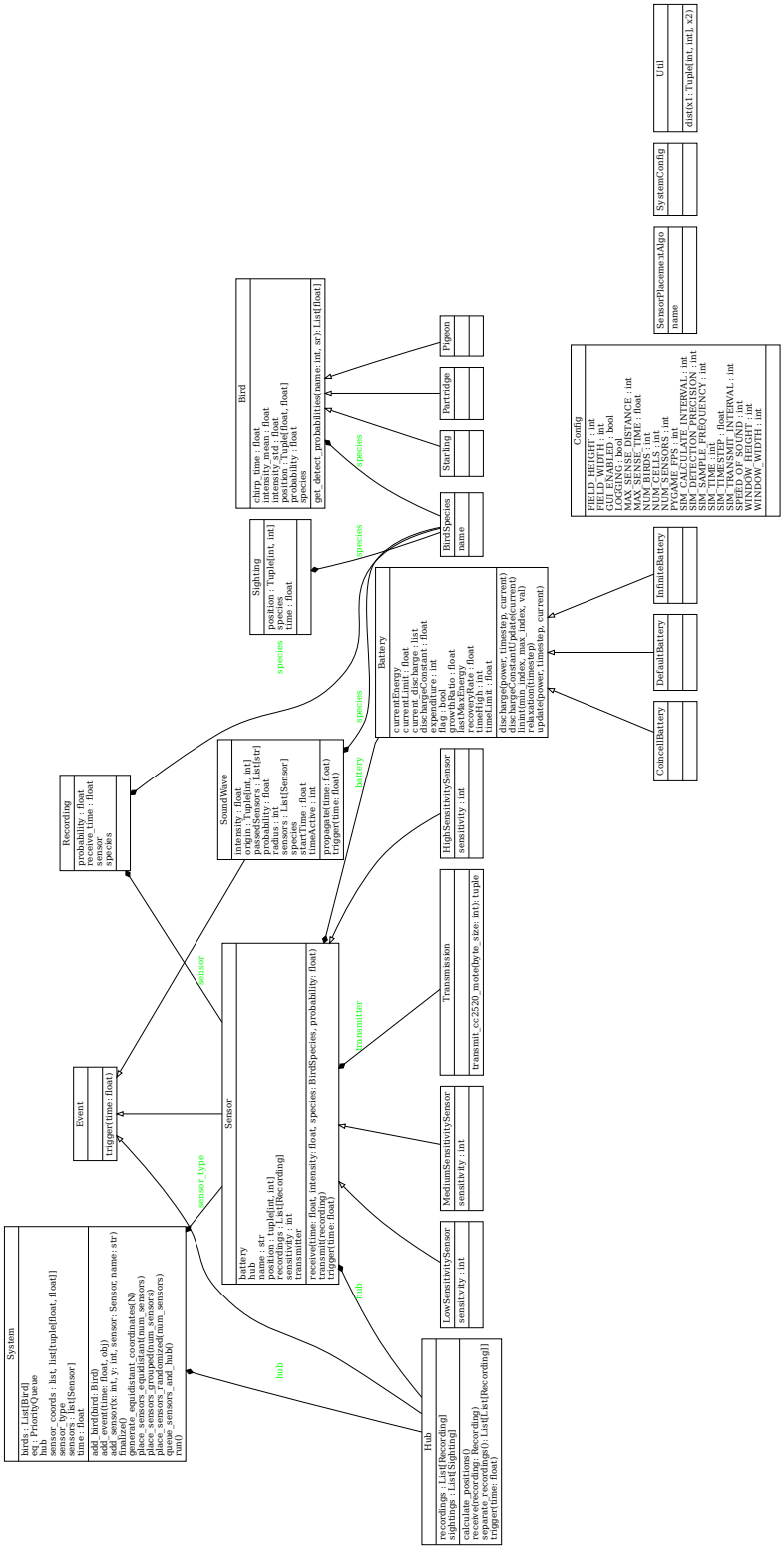

Figure 4: UML class Diagram for the simulator

B. CONFIGURATION KNOBS

Config knob name	Description	Value
FIELD_WIDTH	Width in meters of the field being simulated	150 meters
FIELD_HEIGHT	Height in meters of the field being simulated	150 meters
NUM_BIRDS	Number of birds to generate over the entire simulation timeframe	Modified during tests
NUM_SENSORS	The number of sensors to place onto the fields	Modified during tests
SIM_TIME	The timeframe in seconds to be simulated	14400 seconds (4 hours)
SIM_TIMESTEP	The timestep at which the simulation may advance. Mainly used when propagating sound waves	0.0006 seconds
HUB_DETECTION_PRECISION	Maximum allowed distance between a Bird and the calculated position of a bird to count as a detection	10 meters
SENSOR_TRANSMIT_INTERVAL	Time in seconds between data transmissions to Hub from a sensor node.	$\frac{\text{SIM\_TIME}}{10}$
HUB_CALCULATE_INTERVAL	The time in seconds between the Hub determining bird locations from received recordings	$\frac{\text{SIM\_TIME}}{9}$
NUM_CELLS	Number of AA battery cells to be used in a battery bank	20
SAMPLE_FREQUENCY	The sampling frequency to simulate when determining bird species	44000

Table 8: Description of configuration knobs that can be modified in the simulation

C. SUMMARY OF DETECTION RATIO FOR VARIOUS CONFIGURATIONS

Table 9 summarizes the findings that were obtained with the simulator, showing average detection ratio for all combinations of sensor placement algorithm, microphone sensitivity and number of microphones.

SPA	**MS**	**Number of microphones**				
		5(2)	10(3)	15(5)	20(5)	25(7)
Equidistant						
	Low	0.00	0.00	0.00	0.00	0.00
	Medium	0.00	4.00	37.03	46.37	73.83
	High	8.17	66.62	84.42	92.03	97.52
Grouped						
	Low	5.43	7.97	13.37	13.60	17.35
	Medium	19.70	30.55	52.22	52.98	68.72
	High	29.42	41.47	68.35	64.75	92.72
Random						
	Low	0.00	0.28	0.77	1.32	2.00
	Medium	2.73	14.60	31.22	44.63	56.95
	High	19.72	55.25	72.85	83.45	89.42

Table 9: Detection ratios for all combinations of sensor placement algorithm, microphone sensitivities and number of microphones. Grouped microphones are placed in clusters of 4 at one sensor. The value in brackets indicates the number of grouped sensor nodes with 4 microphones that were used.
SPA: Sensor Placement Algorithm, MS: Microphone sensitivity

D. MICROPHONE RESEARCH

Model	Powercons.	SNR (dBA)	Sensitivity (So)	Price (DKK)
MP34DT05-A[6]	**Normal**: 650 μA, **Powerdown** : 5μA	64	Min: -29 dBFS Typ: -26 dBFS Max: -23 dBFS	10,05
ADMP404/INMP404[3]	**Normal**: 250 μA	62	Min: -41 dBV Typ: -38 dBV Max: -35 dBV	18,55 23,44-31,30
POM-2730L-HD-R[5]	500 μA	74	Min: -27 dB Typ: -30 dB Max: -33 dB	20,38
SPU0410LR5H-QB[2]	**Typ**: 120 μA, **Max** : 160μA	63	Min: -41 dBV Typ: -38 dBV Max: -35 dBV	5,47
AOM-5024L-HD-R[4]	500 μA	80	Min: -21 dB Typ: -24 dB Max: -27 dB	21,26

Table 10: Microphones found with their respective model, consumption, SNR, So and price in DKK.