Freie Universität Berlin

Master's Thesis at the Department for Informatics and Mathematics

Workgroup Intelligent Systems and Robotics - Biorobotics Lab

A flying platform for behavioral and electrophysiological studies in honeybee navigation

Julian N. G. Petrasch
Matriculation Number : 466....
julian.petrasch (at) fu-berlin.de

First Examiner: Prof. Dr. Tim Landgraf
Second Examiner: Prof. Dr. Dr. h.c. Randolf Menzel

Berlin, September 5. 2018

Abstract

The primary purpose of this thesis is to develop a flying platform for extracellular recordings of the honeybee brain. The bee is attached to a quadcopter which carries the necessary equipment for stabilizing the bee and for measuring its neural activity. This requires an analog circuit for amplifying the bee's neural signals, an ADC for digitizing the signals and a system for storing and synchronizing extracellular recordings while the quadcopter simulates the bee's flight. Furthermore, software for preprocessing the neural and copter-data, for testing the system and for analyzing the data had to be developed. I present the first recordings of neural activity ever made in a living honeybee on a flying copter. A significant correlation of the copter's (yaw) rotation and the neural activity in honeybee's alpha-lobe could be found in the first experiments. The developed system will be used for experiments to study the long-range navigation capabilities of flying honeybees.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

September 5. 2018

Julian N. G. Petrasch

Contents

1 Introduction

Although honeybees (*Apis mellifera*) have an extremely small brain (around 1 mm^3 [23]), they are extraordinary navigators. Forager bees are able to recur to food sources several kilometers away. If they are released on random locations up to a few kilometers around their hive, they are able to find their way back to the hive. After finding a new food source, they give the location to other bees in the hive through the so-called waggle dance [15]. To get the distance and direction (polar coordinates) from the hive to a food source, the bee measures the optical flow [13] and uses path integration [15] to get the direct vector.

Furthermore, bees can determine the direct path between two known places even if they do not know the terrain between the two places. That cannot be explained only by path integration [25]. A possible explanation is that the bee has a cognitive map [25] [7] containing landmarks and places and some location information about them which could be used for self-localization of the bee. Only animals with much more highly developed brains like mice [21] are expected to have such capabilities. So, this hypothesis is still under discussion [8]. Newer publications are indicating that that the cognitive map exists [36] but more experiments are required.

In the next section 2 is the relevant related literature presented. The requirements of the flying platform and the selection of the copter and gimbal are discussed in section 3. Section 4 then describes the development of the electrophysiology setup, its integration on the copter and the development of the software for processing and synchronizing the neural and copter-data. In section 5, the experiments and results are described. In section 6, the work and the results are discussed. The last section, section 7, provides an outlook over possible improvements and further experiments.

2 Related Work

There are different approaches to investigate the bee navigation. Each approach has advantages and disadvantages.

A virtual environment can be used to stimulate the bee like in Taylor et al., 2013 [38]. The setup consists of four screens around the bee and an airflow machine. The screens show optical flow and the airflow machine generates corresponding wind. The setup cannot imitate nature completely. The Temporal acuity of honeybee vision is between 200 Hz (Srinivasan et al., 1985 [33]) and 300 Hz (Autrum et al., 1950 [3]). The screens used in Taylor et al., 2013 [38] had a refresh frequency of 60 Hz. They did not simulate polarization or odor. That can have effects on the results. Furthermore, they were just measuring the orientation of the honeybee's abdomen and not neural activity.

In Stone et al., 2017 [36] bees were exposed to optical flow and rotation while the neural activity was measured. Speed and compass neuron convergence was identified. Based on these results and anatomical features, a biologically constrained model of path integration in the brain was developed and evaluated. The disadvantages on trials with real bees are similar to Taylor et al., 2013 [38]. The modeling of parts of the bee's brain makes complex analyses possible. The model was used to perform path integration on a robotic platform. After a random outbound journey the robotic platform was successfully guided back by the modeled circuit. During such tasks the activity of each artificial neuron can be observed which is great for a complex analysis. The biggest problem of working on a model is that not all differences between the model and the brain of an actual bee can be eliminated. Another issue is that the input for the model used during the trials on a robotic platform was generated by a camera and sensors which differ from the inputs captured by the senses of a honeybee.

An integrated circuit for measuring neural activity carried by the insect (Harrison et al., 2010 [17]) makes it possible to do electrophysiology and other measurements on real free moving insects. But the data has to be transmitted live to a receiver which can be no more than 2 m away. So long range flights which are necessary for investigating the bee's navigation are not possible. Also, the integrated circuit weights 0.79 g. A honey bee can carry around 60 mg of pollen (Wolf et al., 1989 [42]). A 10.5 mg transponder (Menzel et al., 2012 [26]) is the limit where the bee still behaves normally. This is why the integrated circuit is too heavy for honey bees.

In the first phase of the Neurocopter-Project a custom quadcopter was developed [20] [10]. It is equipped with a powerful on-board computer (ODROID-U3) which can read the sensor data and control the flight. Different bee inspired algorithms and neural models were implemented for it like the neuromorphic extraction of the quadcopter ego motion from sparse optical flow fields [18] [41] or a collision detection algorithm executed by the ODROID-U3 [43]. So this system can be used to test models in the real world. Further, it was used as a prototype flying platform for extracellular recordings of a honeybee.

3 Development of a flying platform for bee experiments

This section describes the selection of the copter and of the mechanical components and their integration.

3.1 Requirements

The system has to be used to investigate the neuronal activity of a bee during a flight. So, the flying platform must be able to fly as similar like a bee as possible. That means it must be able to fly at least as fast as bees which is 27 km/h [40]. So, including reserve for wind it should fly at least 50 km/h. Higher air speeds are too different from what honeybee's can do and could have unknown effects on the experiment results. It must be able to change the flight direction fast and to even stop in midair. For practical reasons it should be able to take-off and land vertically (VTOL).

A plane cannot do these maneuvers and helicopter is to complicated to handle. These requirements are, however, fulfilled by a multicopter. A multicopter is controlled solely by changes of the rotational frequency of its propellers [31]. This results in a great reduction of mechanical complexity [31] which is very useful for hardware used in field experiments.

A disadvantage of a copter is that it must tilt to fly in a direction. Unlike an airplane even to fly straight forward the copter has to tilt forward. That as well as its vibrations can be compensated by a gimbal. The gimbal can also be used to change the orientation of the bee which is useful for experiments.

To be able to determine neuronal correlates and to log the experiments, the flying platform needs to measure its speed, position, orientation, acceleration and angular acceleration. Experiments have to be repeated many times to increase reliability of the data. So, the copter needs to be able to fly a programmed route and to take off and land automatically.

The copter must be able to carry the bee, the gimbal, the shielding, the amplifier board and the data-logger. So, it must lift at least 600 grams and it should be easy to mount the equipment on.

3.2 Selecting the Copter

There are many multicopters available. But most of them are for photography and not for adding own equipment. That leaves a few multicopters made for developers or the option to self-build a copter. Building a reliable and easy to use copter is a huge effort [20], so it was decided to buy a ready to fly (RTF) copter for developers.

Additionally, to the requirements in 3.1, the whole system:

- should not weigh more then 5 kg due to the German aviation law which requires permissions for copters above 5 kg. To make sure the 5 kg limit will not be exceeded, the copter's weight without the additional equipment should be below 4 kg. A lighter copter also has less inertia so it reacts faster which is closer to bee flights.
- should have a flight time of at least 15 minutes to be able to do longer trials and to be more efficient.
- must have an on-board interface to enable a self-programmed micro controller to readout the drone telemetry data and to control the drone if necessary.

Table 1 shows the copters which were considered. Only the *DJI Matrice 100* fulfilled all requirements (see figure 1). There was also good experience in the work group with DJI drones. So, the *DJI Matrice 100* was chosen.

Copter	Load >500 g	Speed >50km/h	Weight <4 kg	Flight Time >15 min	Equitment Mountable	Waypoint Following	Onboard Interface	Position Hold	Manual Control	Has Needed Sensors
DJI Matrice 100	√	√	√	√	√	√	√	√	√	√
Yuneec Tornado H920	√	-	-	√	o	√	-[1]	√	√	√
Easy Drone XL Pro	√	NA	√	√	√	√[2]	√	√	√	√
SteadiDrone Mavrik X4	√	√	√	√	o	√	√[2]	√	√	√

Table 1: Comparison of different copters

[1] SDK just for control via Apps. Not onboard. [14]

[2] Pixhawk Mavlink Protokoll. Due to the direct connection of the DJI remote to a smartphone and easy to use apps, the DJI protocol is nicer for field usage.

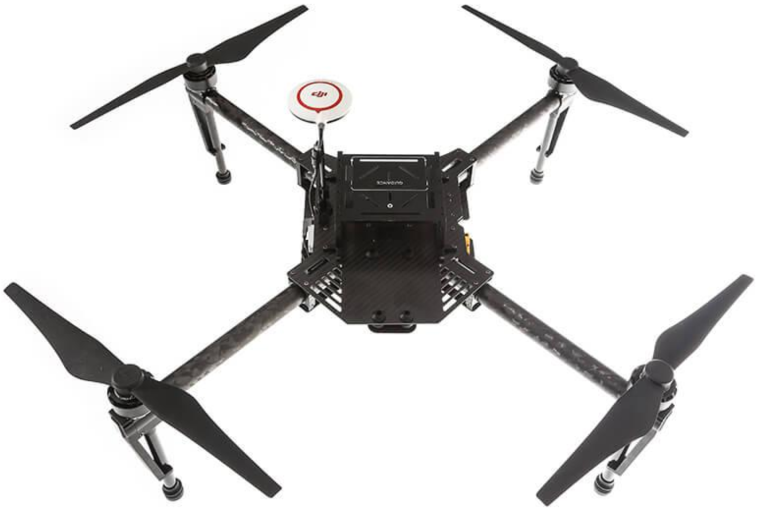


Figure 1: Unmodified DJI Matrice 100. [1]

3.3 Gimbal

In order to fly forward, a copter changes its flight attitude. But bees do that differently. So, the bees attitude has to be independent from the copters attitude. This is the reason why a gimbal is used. The bee should also not be able to see the propellers. That is why it has to be on a cantilever. A big gimbal could stabilize the whole cantilever. But that would result in a heavy construction and due to leverage of the cantilever the movements of the gimbal would be magnified. The magnified gimbal movements could be like a tremble for the bee. So, the gimbal has to be at the end of the cantilever. That means it must be as light as possible, but strong enough to carry the beeholder and the preamplifier board[3]. After some research, it was decided to use the *CopterLab - 2 Axis Mobius Stabilized FPV Gimbal for Vortex 285 Mike Version*[4]. It weights 70 grams, is designed to carry a 44 grams *Mobius Actioncam*[5] which is around the weight needed for the Neurocopter . Further the gimbal controller can be configured which keeps the system flexible.

The gimbal camera holder was replaced by an aluminum holder with space for the preamplifier board and with a mounting hole for the beeholder. All analog components and wires for analog signals are shielded with copper tape (see figure 2).

3.3.1 Gimbal Controller Configuration

During tests, it came out that the gimbal motors got extremely hot which can cause damage. That is why battery voltage applied to the motor was reduced from 5.6 V (120/255 steps) to 3.3 V (70/255 steps). To change the configuration the gimbal controller was connected via UART and *SimpleBGC GUI2.40* [2] was used. The new

[3] To reduce noise the preamplifier board has to be as close as possible to be the bee.

[4] https://copterlab.com/2-axis-mobius-gimbal-for-vortex-285-mike-version

[5] http://www.mobius-actioncam.de/shop/mobius-actioncam-hd-standard/

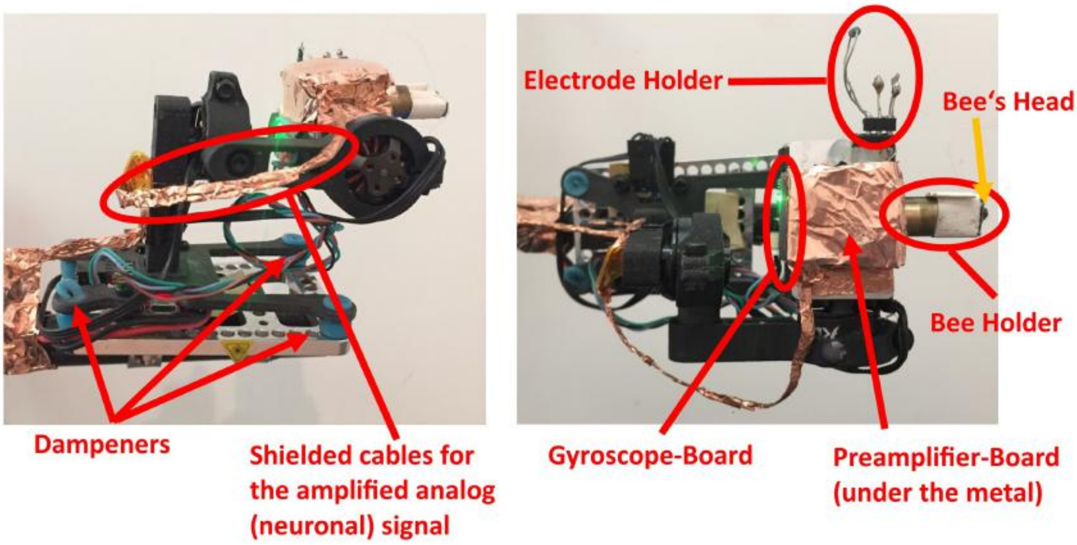

Electrode Holder
Bee's Head
Bee Holder
Dampeners
Shielded cables for the amplified analog (neuronal) signal
Gyroscope-Board
Preamplifier-Board (under the metal)

Figure 2: Gimbal with shielding (copper tape) and beeholder (upper right corner)

configuration file can be found in the Neurocopter Gitlab Repository: *https://git.imp.fu-berlin.de/ast/Master_Thesis_STM32_Neurocopter_SPI_Logger/blob/master/Gimbal/bienenausleger_standardEinstellugenNeurocopter.profile*

3.4 CAD Model

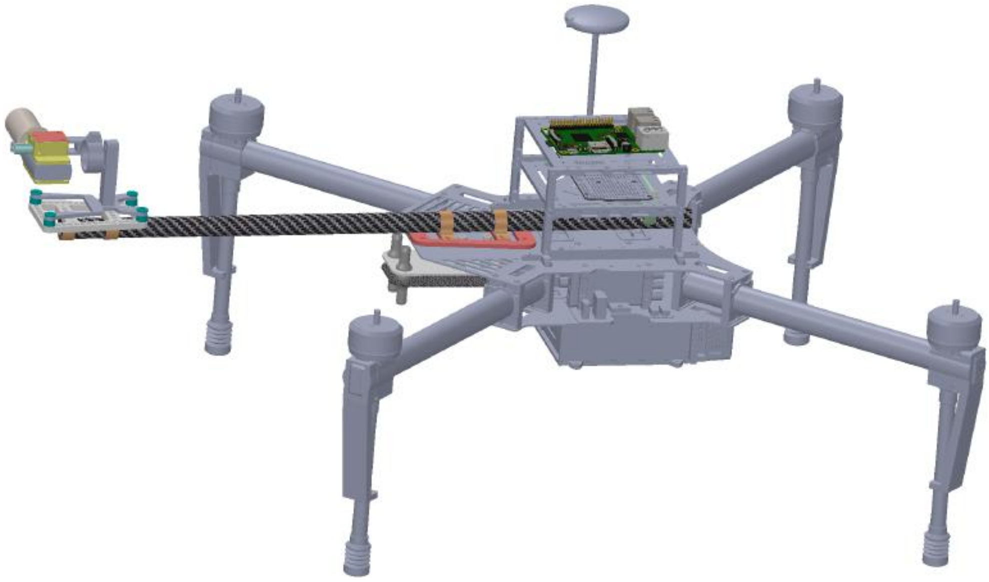


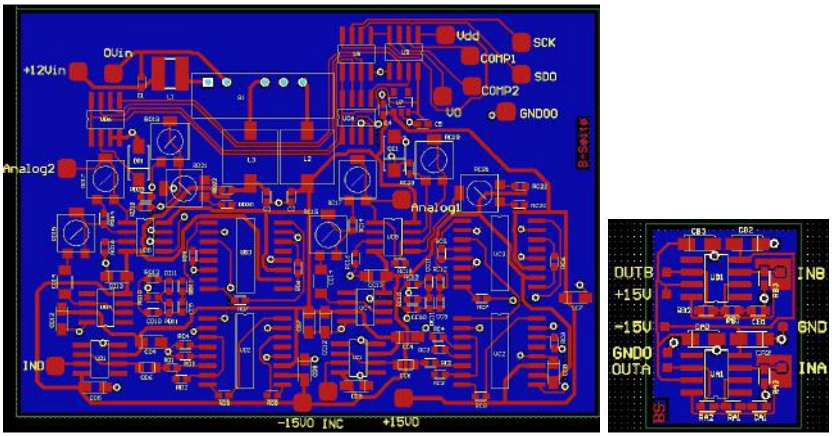
Figure 3: CAD model of the Neurocopter′s most important components.

4 Development of the electrophysiology setup and its integration on the copter

This section describes the hardware for electrophysiological recordings, the development of the on-board computers, shielding and testing of the whole system and the implementation and usage of the data reduction pipeline.

4.1 Amplifier Board

The amplifier board was designed by Dr. Benjamin Paffhausen and Thorsten Thiele based on the design showed in [4][6]. It amplifies the analog signals (two channels) by a factor 800[7]. The amplified analog signals are digitized by two *LTC1864* ADC's [9] or the analog signals can be used as an input for an audio amplifier to convert the signal to audio. The neural spikes have a frequency of around 1 kHz. Humans are very sensitive to that frequency. Each neural spike sounds like a popping popcorn. So the audio output enables to listen to the neural activity in real-time which is useful for the preparation of the bee.



(a) Amplifier board

(b) Preamplifier board

Figure 4: Layout of the amplifier boards. Graphic by Dr. Benjamin Paffhausen

In the first working iteration of the amplifier board, ADuM1200AR isolators [12] were used to isolate the amplifier board from the OBC (on-board computer)[8]. The readout with the Raspberry Pi worked well. But after changing to the STM32F4-Microcontroller, some bits of the 16-bit values were wrong. After some debugging, it came out that the isolators[9] were not fast enough. So, for further tests the sample rate

[6] Warning: The design in [4] has an error which makes the amplifier unusable. It took many hours to find the error and to fix it. There will be a publication by Dr. Benjamin Paffhausen with the corrected design.

[7] The Amplification can be adjusted by potentiometers on the amplifier board.

[8] The OBC produces noise which has strong effects on the analog electronic of the amplifier board.

[9] The used isolator consists of two circuits. Circuit 1 of an isolator is converting an electrical digital signal to a magnetic field. That field is converted to a digital signal by circuit 2 of the same isolator. Circuit 1 gets its energy from the analog circuit and circuit 2 from the OBC. So the only connection

had to be reduced to 18 kHz. After replacing the ADuM1200AR by ADuM1200CRZ isolators [12], much higher samples rates were possible. The electrodes in the honeybee's brain are directly connected to the preamplifier board. So, the preamplifier board has to be as close as possible to the bee. That is why it is directly mounted on the bee holder gimbal (see figure 2). The preamplifier board (see figure 4b) provides electrical impedance transformation so that the resulting analog signal can be transferred using longer wires. The big amplifier board is mounted in the DJI Matrice 100 expansion bay under the OBC. It has ten potentiometers (five for each channel to adjust the analog signal). The function of the potentiometer is described in table 2.

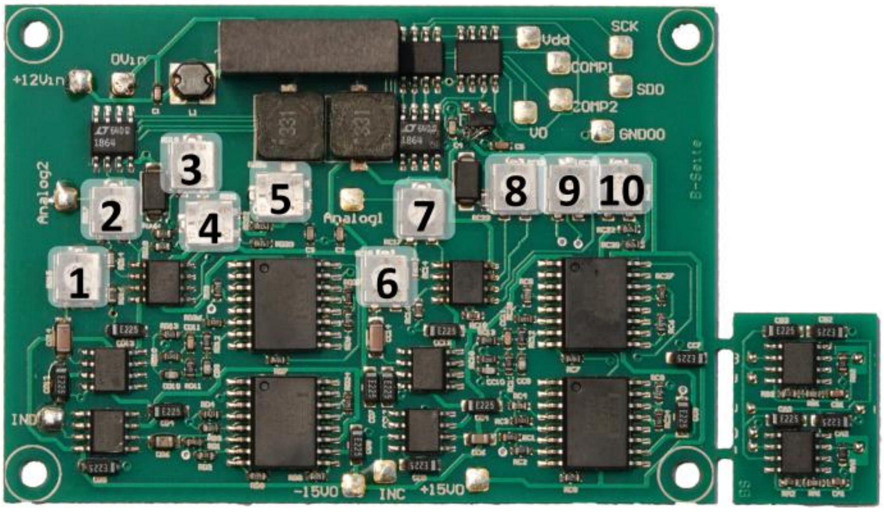


Figure 5: Amplifier boards with numbered potentiometers (see table 2 for potentiometer functions).

between the OBC and amplifier board is the magnetic field. So electrical noise from the OBC can not go over to the amplifier board.

	Channel	Function	Recommended and max. resistance [Ω]
1	0	Second Amplification	1560/1560
2	0	First Amplification	49200/49200
3	0	ADC Amplification	2860/2860
4	0	ADC Offset	5/970
5	0	Amplification of Active Filter	9600/21700
6	1	Second Amplification	1760/1760
7	1	First Amplification	26100/26100
8	1	ADC Amplification	4560/4560
9	1	Amplification of Active Filter	9400/21300
10	1	ADC Offset	10/1020

Table 2: Amplifier board potentiometer functions. Recommended resistance values are measured while the board was switched off, but the potentiometers were not soldered out from the PCB. So, these are not the values of the potentiometers but the values of the measure-points. Adjusted resistance values were measured between the left pin and the wiper pin.

4.2 Requirements of the On-Board Computer

The On-Board Computer (OBC) has to read and save the brain data from the amplifier board (see section 4.4) with a sample rate of at least 5 kHz and it must provide a way to synchronize this data with the drone's flight logs (saved on the drone). To prevent delays and overheating in the field experiments, it should work autonomously and boot every time the drone is switched on. It must have an interface to control the flight of the drone (via the DJI Onboard SDK[10]). Because it has to be mounted on the drone, it must not weigh more than 100 g and has to be smaller than 100 mm X 100 mm (size of the DJI Matrice expansion bay [37]). Due to the high-power consumption of the drone (hovering of the DJI Matrice 100 with a payload of 500 g takes around 353 W[11]), a power consumption of the OBC up to 100 W would be acceptable. In practice, the devices used in the experiments consume much less power which is why the power consumption requirements can be ignored.

4.3 Raspberry Pi as On-Board Computer

In the first iteration of this thesis The *Raspberry Pi 3* was used as the Neurocopter's OBC but later replaced by a STM32F4-Microcontroller.

The *Raspberry Pi 3* is a Single-Board Computer (SBC) with a 1.4 GHz ARM multi-core CPU. For the Neurocopter it was used with the Debian-based Raspbian Linux distribution. Thanks to the Linux operating system which includes drivers and makes it possible to use high level programming languages, developing for it is much faster and easier than for example for a microcontroller. The *Raspberry Pi 3* has an integrated

[10] For the DJI Onboard SDK see: https://developer.dji.com/onboard-sdk/

[11] Following [37] the hoovering time of the DJI Matrice 100 with a TB47D -battery (99.9 Wh) is 17 minutes. So, the power consumption of one hour hoovering is: $99.9\ Wh * (60min/17min) = 352.6\ Wh \rightarrow$ for hoovering the DJI Matrice 100 needs 352.6 W.

WiFi module which makes it possible to transfer neuronal data while the Neurocopter is flying. Live neural data of the flying Neurocopter is very useful for debugging. For example the change of the noise, due to the missing ground when the copter is launching, can seen live. Or a running experiment could be changed if necessary. If a cable without isolators (see 4.1) is used to connect the Raspberry Pi with a computer, it causes massive noise. So, the ability to use WiFi for data transmission makes the system ideal for field and lab use.

Figure 6: Raspberry Pi 3

OBC Operation When the Raspberry Pi finishes booting, a cronjob runs a bashscript *bee_logger_wrapper.sh*. That script executes the *nc-obc-binary* and writes the output in a textfile which can be used for debugging. When the *nc-obc-binary* terminates or crashes, *bee_logger_wrapper.sh* will restart it. That should prevent data loss during experiments. In the *nc-obc-binary*, first the GPIOs are initialized and then the status LEDs blink to show that booting was successful. Afterwards, the SPI buses are initialized, and the log file is created. (For fast SPI access *pigpio C libary* [29] was used.) In an endless loop, the two ADCs are readout 15 000 times per second. Every 2 000 readouts, the drone telemetry data is saved in the same log file as the data from the ADCs, so the telemetry data is synchronized with the brain data.

A camera is recording the bee, what the bee sees. To synchronize that camera with the on board computer, the camera also records LEDs which are controlled by the OBC. Every 750 000 readouts, a new thread is started which lets the ID LEDs (clock- and data-LED) give out the *id_sync*, an ID used to synchronize the data. In the moment the *id_sync* is written to the log file, both ID LEDs go on for two seconds and off for one second. Afterwards they transmit the *id_sync*. The transmission via the LEDs is like transmitting a 32-bit unsigned integer on a SPI bus with 1 Hz and Least Significant Bit (LSB) first. One LED is the clock and toggles twice per second (clock-LED). In the moment the clock-LED goes off (falling edge), the data LED gives out the current bit (If it is on it is a 1. Otherwise a 0.). The transmission is visualized in figure 7.

Every time the only button on the Neuropter OBC is pressed, a new logfile is created. That is useful if a new experiment or trial is started. For a flow chart of the OBC

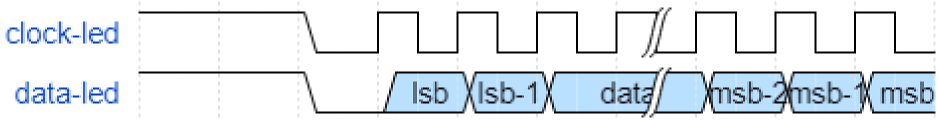


Figure 7: Timing Diagram of the *id_sync* via LED's. Logic high means LED on on logic low means LED off.

code on the Raspberry Pi see figure 8.

Optionally, the OBC can send all data as UDP datagrams over Wireless Lan or an Ethernet cable. Then the data can be visualized live via the *Test and Calibration Software* (see 4.8). UDP was chosen instead of TCP because for viewing live data, speed is much more important than data completeness. Especially on an unstable wireless connection with high packet loss, the use of TCP can result in long delays of a few seconds which would make some standard tests like triggering spikes with light hard to perform.

The *nc-obc-binary* is written in C++ and includes DJI Onboard SDK features. To build it on a Raspberry Pi, the standard make file from DJI had to be modified (see Neurocopter repository).

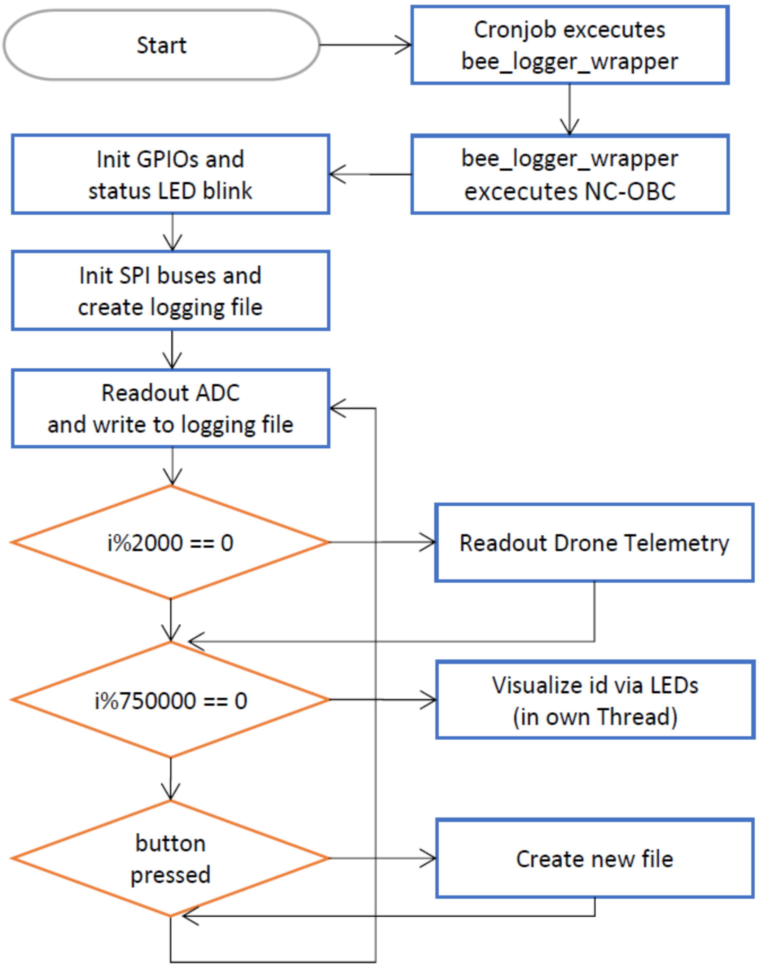

Start
Cronjob excecutes bee_logger_wrapper
bee_logger_wrapper excecutes NC-OBC
Init GPIOs and status LED blink
Init SPI buses and create logging file
Readout ADC and write to logging file
i%2000 == 0
Readout Drone Telemetry
i%750000 == 0
Visualize id via LEDs (in own Thread)
button pressed
Create new file

Figure 8: Flow chart of the OBC on the Raspberry Pi

4.4 ADC Readout

To digitize the analog signal from the amplifier board, two ADCs [9] are used (one for each channel of the amplifier board). In a first attempt, the ADCs were read out sequentially. The ESCs and Motors of the multicopter were causing high noise levels. To reduce the noise, both channels are digitally subtracted from each other which results in a virtual differential channel. The motor noise is similar on both channels but the neural signals should be different on each channel. So, on the virtual channel just the neural signals should be left.

For low frequency noise, this worked well (around 80% noise reduction), but high frequency noise was just reduced by around 50% which means the signal-to-noise ratio for measuring neuronal spikes is around 1. The problem was that the two ADC's were doing the measurement not exactly at the same time. For low frequency noise the effect is not as strong as for high frequency noise. The copter generates high frequency noise, so a solution is needed.

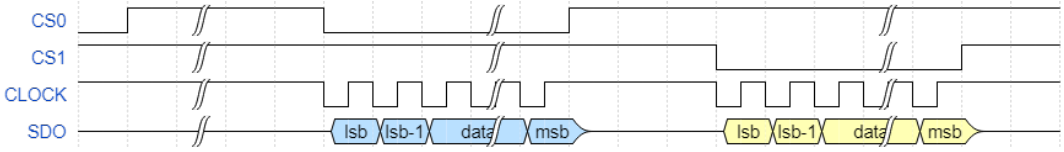


Figure 9: Timing diagram of all lines during sequential readout. Signals from ADC0 are blue and signals from ADC1 are yellow.

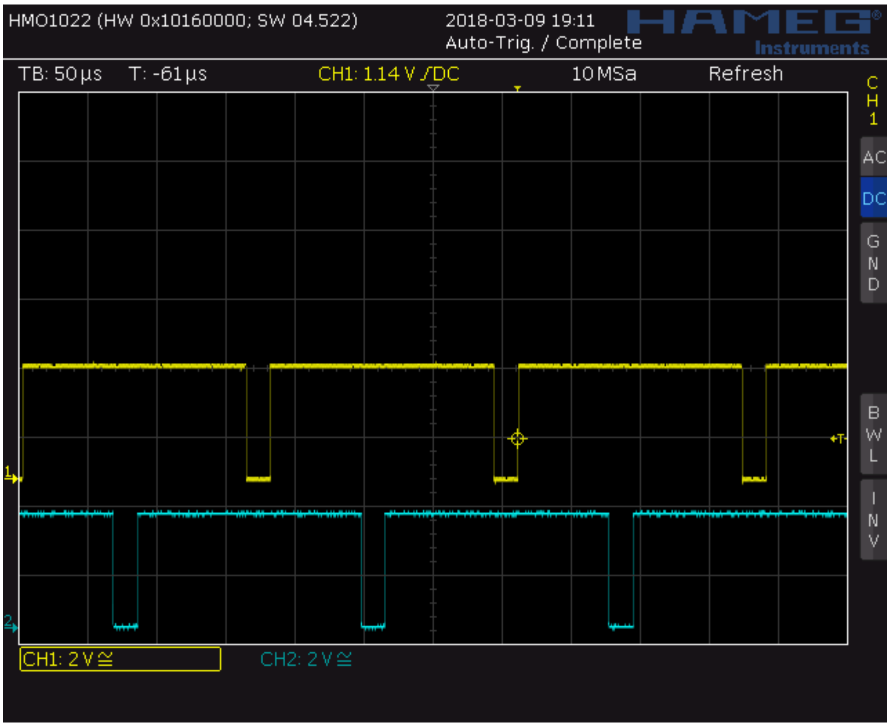

HMO1022 (HW 0x10160000; SW 04.522)
2018-03-09 19:11
Auto-Trig. / Complete
TB: 50 µs T: -61 µs CH1: 1.14 V /DC 10 MSa Refresh
CH1: 2 V ≅ CH2: 2 V ≅

Figure 10: Measured Chip Select lines of both ADCs during sequential readouts. CS0 is plotted yellow and CS1 is plotted blue.

The solution is to do the measurements of the ADCs simultaneously. But due to the layout of the amplifier board both ADCs share the data and clock line which makes it impossible to measure and readout both ADCs in parallel. To avoid that problem, one ADC has to be readout two times (see figure 11):
Therefore, the CS-lines (Chip Select) of both have to go from low to high ADCs simultaneously [9]. (On the Raspberry Pi this is possible using the *gpioWrite_Bits_0_31_Set()*-function of the pigpio C library [29]). That starts the measurements on both ADCs. Then CS0 is set to low which starts the readout phase. ADC0 is readout through clocking. After that, CS for ADC1 is set to Low to start its readout phase (see figure 9). –> Problem: Both ADCs share the Clock and Data-line –> So, after ADC0 is readout (and still in readout mode) it gives out 0's on each clock cycle. So the 1's from ADC1 go over ADC0 into ground –> No data can be readout from ADC1.
Solution: After the readout of ADC0, a measurement needs to be triggered so ADC0 does not block the SDO data line. Then the readout of ADC1 can be done. After that ADC0 has to be clocked out. The data will not be saved. And then set CS high on

both ADCs to start a simultaneous measurement.

Figure 13 shows that the motor noise reduction, through subtraction of the two channels, works significantly better. With that improvement it is possible to measure neuronal activity while the Neurocopter is flying. The disadvantage is the second readout of the first channel where no data is stored. It reduces the sample rate of the Raspberry Pi OBC from 15 kHz to 10 kHz. Further experiments showed that this sample rate is not high enough for good spike sorting (classifying the spikes). Overclocking of the Raspberry Pi can increase the sample rate up to 30% but require cooling which causes extra weight and makes the whole Neurocopter system more complicated. Furthermore, much higher sample rates than 15 kHz can make the spike sorting even better. So, a new OBC is needed.

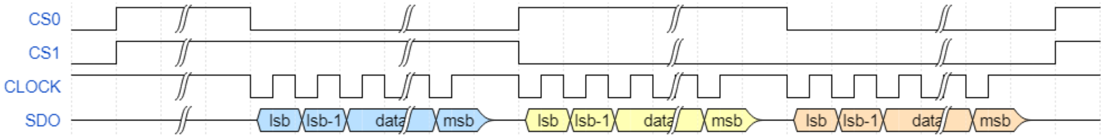


Figure 11: Timing diagram of all lines for a simultaneous measurement. Signals from ADC0 are blue, signals from ADC1 are yellow and signals from ADC0, which are discarded are orange.

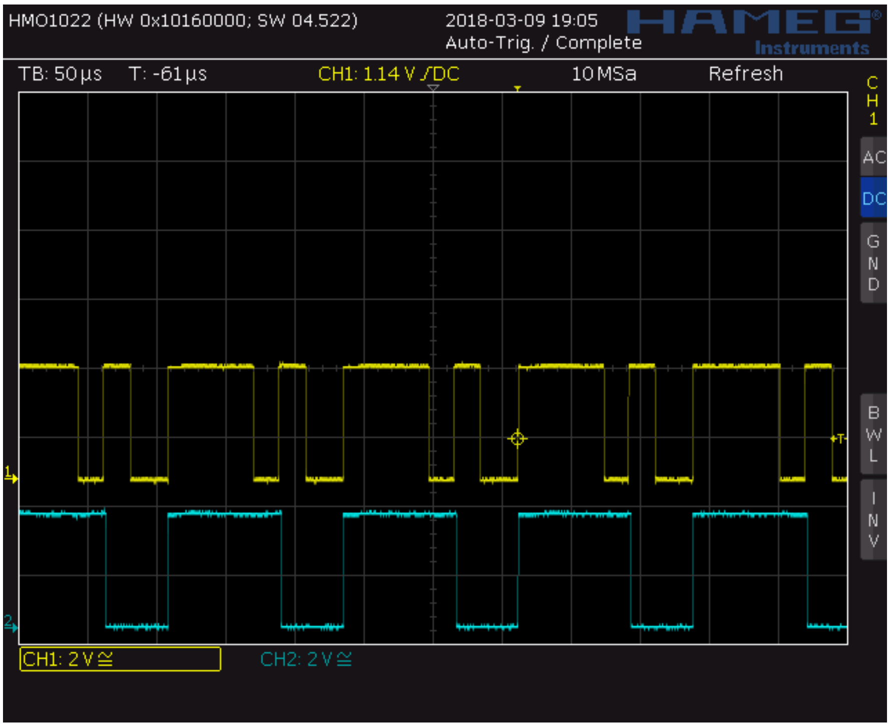
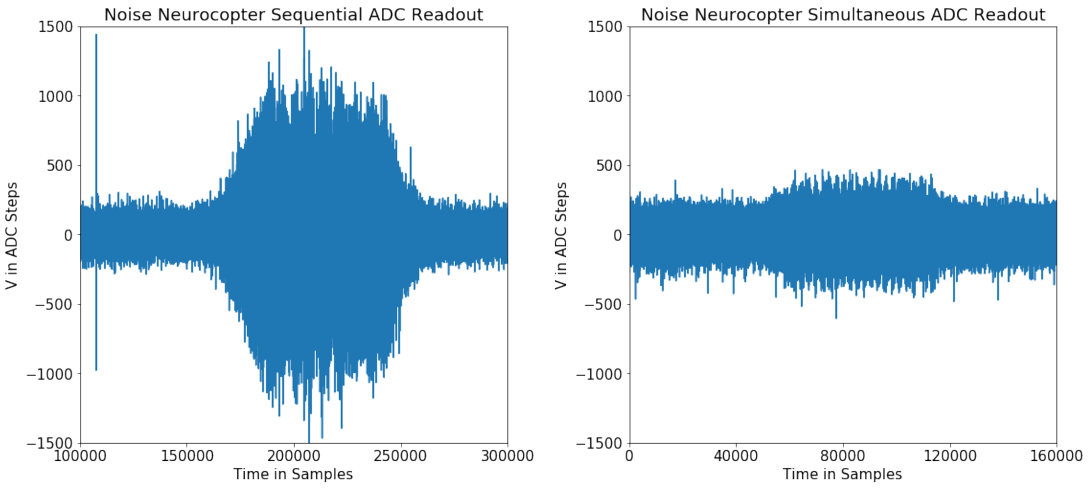


Figure 12: Measured Chip Selects lines of both ADCs during readouts for simultaneous measurements.



(a) Sequential Measurements

(b) Simultaneous Measurements

Figure 13: Test of the drone noise in the virtual differential channel $(Ch(0) - Ch(1) * amplification\_factor)$. The drone motors are running at full speed in the central third of the trials. The higher noise is clearly visible but when the ADCs measure simultaneously and their readouts are subtracted from each other, the noise is reduced significantly (see b).

4.5 STM32 as On-Board Computer

4.5.1 Advantages of a microcontroller

The operating system of the Raspberry Pi limits the systems sample rate to around 15 kHz. Using the Raspberry Pi without an operating system is possible but there is not enough documentation and build-tools to do that in a productive manner. So, a standard high performance microcontroller, the *STM32F407VET [35]*, was selected.

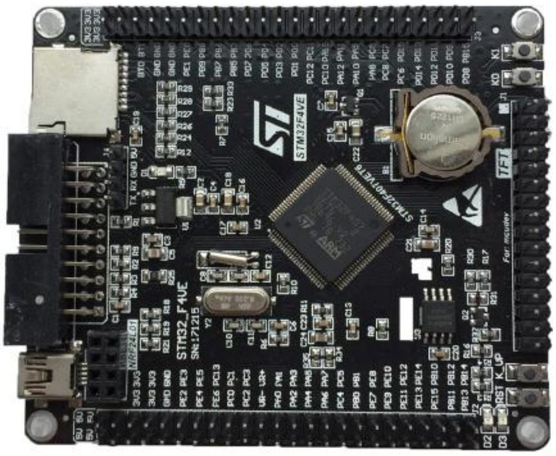


Figure 14: STM32F407VEGT on the NRF2410 board

The microcontroller has a single core with up to 168 MHz, 192 Kb of SRAM, a Real Time Clock (RTC), 140 I/O ports, 4 USARTs, 3 SPIs (the Neurocopter ADCs are connect via SPI) and a SDIO interface (for accessing SD-cards). Compared to the Raspberry Pi 3 with a 4x1.2GHz CPU it has a low performance. But due to the direct hardware access it is possible to archive ADC sample rates of up to 7 200 kHz[12]. Another advantage of the microcontroller is the higher robustness. The file system of the Raspberry Pi can get damaged when it is not shut down properly. To shut down and reboot the Raspberry Pi during experiments takes time. The *STM32F407VET* has a separate memory from which it loads the instructions every time it is powered on. Because of the lower CPU speed, the *STM32F407VET* needs less power than the Raspberry Pi which prevents overheating when the OBC is exposed to the sunlight.

Disadvantages of the microcontroller are that it is harder to program than a Raspberry Pi with an operating system. On the Raspberry Pi, new code can be transmitted via SSH and executed remotely while the microcontroller needs to be connected via a JTAG interface. It has no integrated Wireless LAN. So, the brain activity cannot be shown live. But instead of Wireless LAN, an analog audio transmitter is used to get live brain activity data from the copter. So, the only significant disadvantage of the *STM32F407VET* is that it is harder develop for it.

4.5.2 Overview

Figure 15 shows all electrical components of the Neurocopter system and the used buses except the components which are parts of the DJI Matrice 100. The blue arrows represent data transfer. The direction of an arrow symbolize the direction of the information flow. The red arrows symbolizes the power connections. The direction of the red arrows symbolizes where the power goes. For information about the grounding of the components, see section 4.6. How the STM32 interacts with all sensors and subsystems is described in section 4.5.3.

[12] Sample rates of up to 7 200 kHz were archived using the internal ADCs of the STM32 in triple interleaved mode [35], with DMA and a much lower resolution than the 16-bit on Neurocopter.

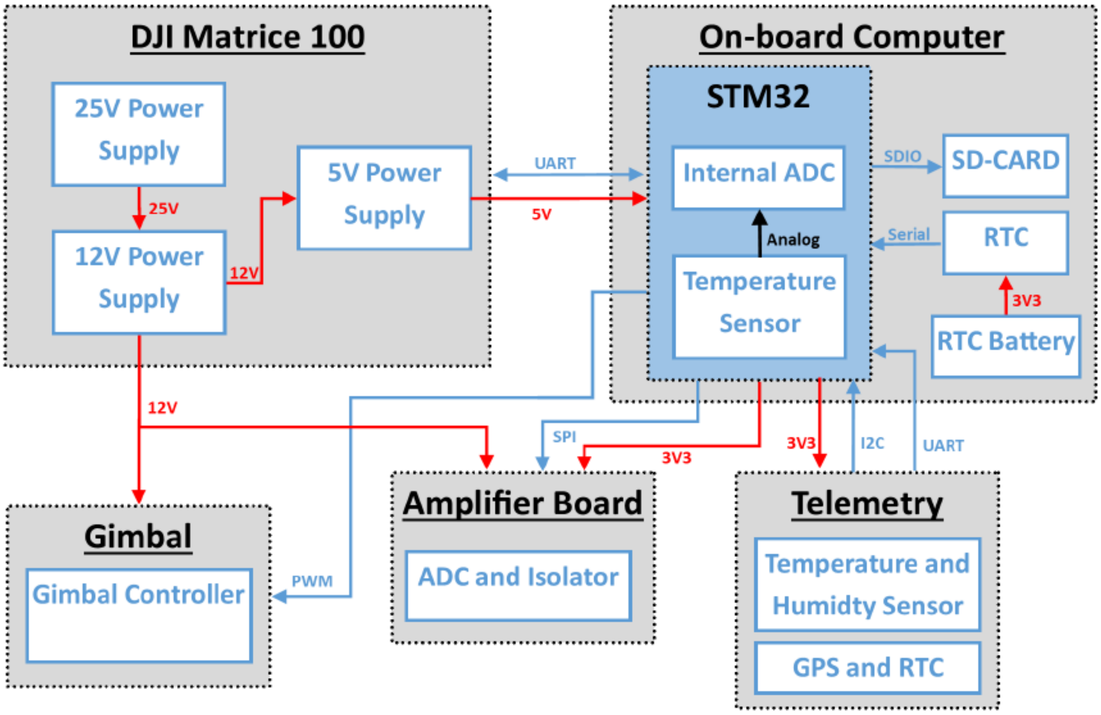


Figure 15: System overview of the Neurocopter System with a *STM32F407VEGT* as On-board Computer

4.5.3 Implementation

The code for the Neurocopter OBC is based on the code of the bee hive monitoring and logging system *Umweltspäher*[13] which was developed by the author (Julian Petrasch) and Morgan Geldenhuys. The *Umweltspäher* code architecture is more complex than needed but tested on 28 systems which have been continuously running for at least 2 months already. It has many integrated modules to readout different sensors which are not needed for the Neurocopter system. That is why all of them except for the driver for the temperature and humidity sensor [22] were removed. Also, most of the logic and data handling code had to be rewritten.

To make the code hardware independent, the *STM32 Cube hardware abstraction layer (HAL)* was used for hardware access as much as possible. For the sake of a clean and simple software architecture, a state machine was used (see figure 16). The state machine was implemented with function pointers. Each state is implemented as a function and returns a pointer to the next state (function). The functions are called in an endless loop. For prints and debug output, the USART2 bus is used.

Before the state machine begins, all the HAL-drivers initialize the needed systems on the microcontroller. When the GPS-module is powered on, it resets itself to a 9600 baud UART output and to an output rate of 1 Hz. But 115200 baud and

[13] A method paper about the *Umweltspäher* is under preparation.

10 Hz are needed for the Neurocopter system. So the STM's UART is first initialized with 9600 baud, configures the GPS to 115200 baud, changes its own baud rate to 115200 baud and then changes the GPS refresh rate to 10 Hz. The GPS is configured through NMEA [19] commands, it should work with other GPS-modules as well.

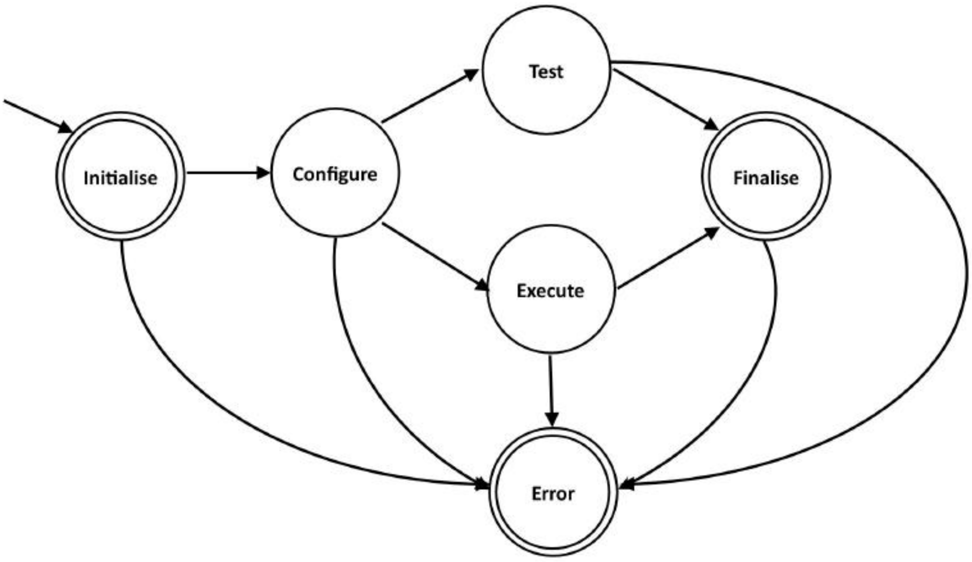


Figure 16: State diagram of the STM32 based Neurocopter OBC

In the **Initialize-state**, the SD-card on which all the measured data is stored, is mounted. For the purposes of readability by a normal computer, the file system FAT32 is used on the SD card. The used module is FatFs [6]. Then the RTC is initialized and set to the current GPS-time which is needed to save timestamps with the measured data.

In the **Configure-state**, a configuration struct is loaded. It contains parameters like maximum file size and the sample rate of the timestamps.

If some hardware needs to be tested, the **Test-state** can be used. It contains methods to test all the connected Neurocopter hardware (like the ADC, GPS, RTC and temperature sensor) and prints the results over the OBC's UART2 interface.

Under normal operation, the system switches from the **Configure-state** to the **Execute-state**. It first starts a timer for the ADCs readout. The timer generates interrupts at the aimed ADC sample rate. In the interrupt handler, the ADCs are readout and the data is stored in a buffer (*adc_buffer*). After the timer is activated the state function goes into an endless loop where the *adc_buffer* is written to the SD-card. Every 10 seconds, the RTC is synchronized by the GPS time[14] and the temperature and humidity is readout and saved with the current date and time to the SD-card. If the maximum file size is reached, a new log file is created.

[14] Test showed that the RTC is 0.7% slower than a correctly running clock. This error is so big that the RTC has to be synchronized every 10 seconds.

The **Finalise-state** can be used to deactivate all the connected hardware, to clear the allocated memory and to put the microcontroller into the standby mode for power saving. It is currently not in use but can be helpful for longer experiments. And due to the state machine architecture, it does not add complexity.

If an error is detected, the system switches to the **Error-state**. It logs the error and tries to restart the Neurocopter OBC.

Using the *STM32F407VET*, it was possible to readout a single LTC1864-ADC in full 16-bit resolution 400000 times per second (400 kHz sample rate) without using DMA. (Due to the layout of the amplifier board, DMA cannot be used for ADC readout.) While reading out both ADCs in the needed way (see 4.4), sample rates of 120 kHz were reached which is the limit of the ADCs in the current configuration[15].

4.5.4 Neurocopter Binary Converter

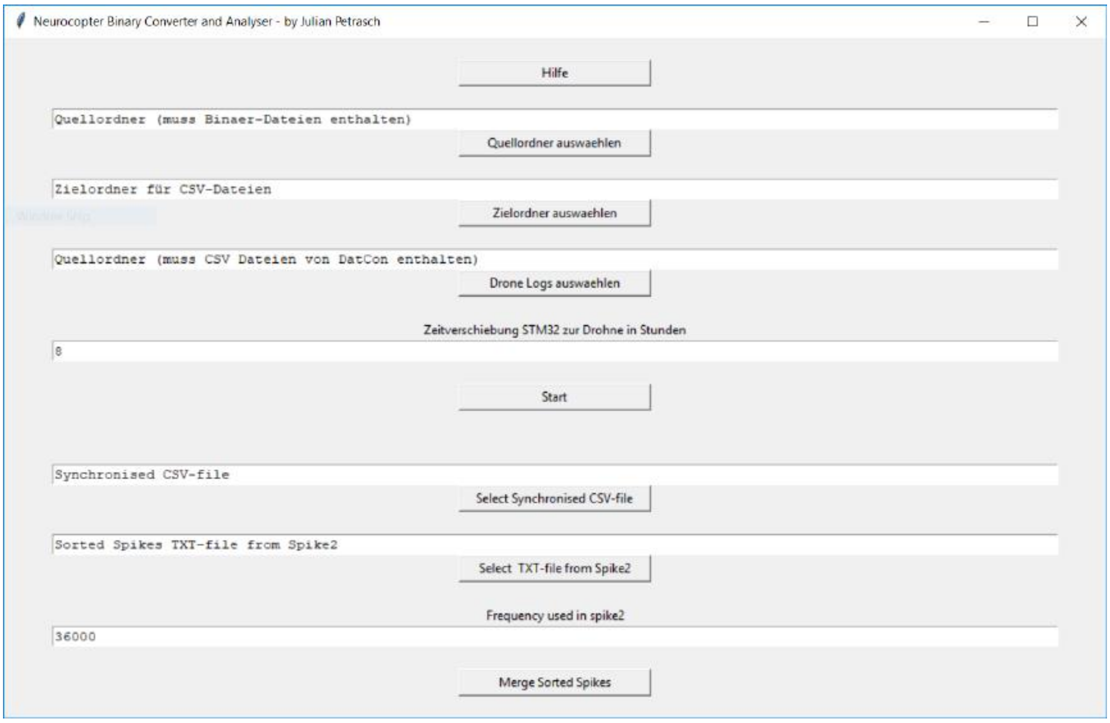
For efficiency reasons, the STM32 saves the data in binary files. To convert them to CSV-files, the *Neurocopter Binary Converter* was developed. It gets the binary files (with the neuronal activity, sensor-data and GPS-timestamps) and the logfiles created by the DJI Matrice 100[16]. It then converts the binary files to CSVs, synchronizes them via GPS-timestamps with the logs of the DJI Matrice 100 and writes the output into one CSV file. Because it should be used also by non-computer scientists, it has a graphical user interface (see figure 17). The drone GPS-timestamps have an 8 hour shift to the UTC time. This time shift can be given as a parameter to the *Neurocopter Binary Converter*.

The software was developed in Python 3.6 and consists of the following classes: **DataConUmweltspaeher** has methods to iterate over a directory, to get all binary files and to convert them to a CSV-file which is done in the *convert_and_analyse_directory*-function. At the end of *convert_and_analyse_directory*, when the conversion to a CSV-file is done, it calls the **NcDjiCsvSync-class** which synchronizes the neuronal- and drone data and saves them in a new CSV-file. It iterates over the drone data and uses the **BrainDataHelper-class** to get the timestamps created by the STM32 to do the synchronization. The **Converter_Gui-class** contains the GUI, callback-functions for file browsing and the callback-function which calls the *convert_and_analyse_directory*-function to start the conversion and synchronization.

The GitLab repository of the *Neurocopter Binary Converter* is available here:
https://git.imp.fu-berlin.de/ast/Master_Thesis_STM32_Neurocopter_SPI_Logger/tree/master /binary_converter_and_sync

[15] Due to the amplifier board design one of the two ADCs has to be readout two times per measurement. The highest possible sample rate of the LTC1864 is 250 kHz [9]. So 125kHz (with margin 120 kHz) is the absolute limit for the Neurocopter system. Higher sample rate are not required.

[16] The DJI Matrice saves the logs in a custom format. To get CSV-files the converter DatCon developed from Rowland Johnson is used.


Neurocopter Binary Converter and Analyser - by Julian Petrasch
Hilfe
Quellordner auswaehlen
Zielordner auswaehlen
Drone Logs auswaehlen
Start
Select Synchronised CSV-file
Select TXT-file from Spike2
Merge Sorted Spikes

Quellordner (muss Binaer-Dateien enthalten)

Zielordner für CSV-Dateien

Quellordner (muss CSV Dateien von DatCon enthalten)

Zeitverschiebung STM32 zur Drohne in Stunden

8

Synchronised CSV-file

Sorted Spikes TXT-file from Spike2

Frequency used in spike2

36000

Figure 17: User Interface of the Neurocopter Binary Converter

4.6 Shielding and Grounding

The aim of the Neurocopter-project is to measure extremely small signals (1-10 μV). The copter motors and Electronic Speed Controllers (ESCs) are generating strong noise on the powerlines and electromagnetic fields which interfere with the measurement of the neuronal activity. During the first tests without shielding, the noise was up to a 100 times bigger than the biological signals. That is why grounding and shielding are essential for being able to measure brain activity on the flying Neurocopter.

Grounding The Neurocopter system has 3 power supplies (see figure 15): The 25 V Lithium Polymer Battery (LiPo), a 12 V power supply which provides an own ground and 5 V power supply which provides an own ground as well. The STM32-board is just connected to the 5 V power supply. It provides the GPS and the temperature sensor with 3.3 V. The grounds of the 5 V and 3.3 V lines are connected. The 3.3 V are also used to power the amplifier-boards isolator. The 12 V power supply powers the amplifier-board and the gimbal. All isolations are connected to the LiPo battery ground.

Shielding The STM32 and its periphery uses just robust digital communication. So it was placed on top of the DJI Matrice expansion bay. Shielding it was not required (see figure 18). But the pins of the STM32 were close to the DJI Matrice GPS module and caused massive interference which caused the drone to stop on automated flights in midair. The solution was to extend the GPS holder by 10 cm.

In a few trials it was found out that the amplifier-board is sensitive to noise generated by the copter motors. So, it was placed in the expansion bay (under the STM32). It is shielded from all sides with copper foil (see figure 18). All the copper foil is connected to the ground of the copter's battery. Exposed to air, the copper foil's surface does not conduct electricity after a few days. So, all the grounding connections have to be soldered on the foil.

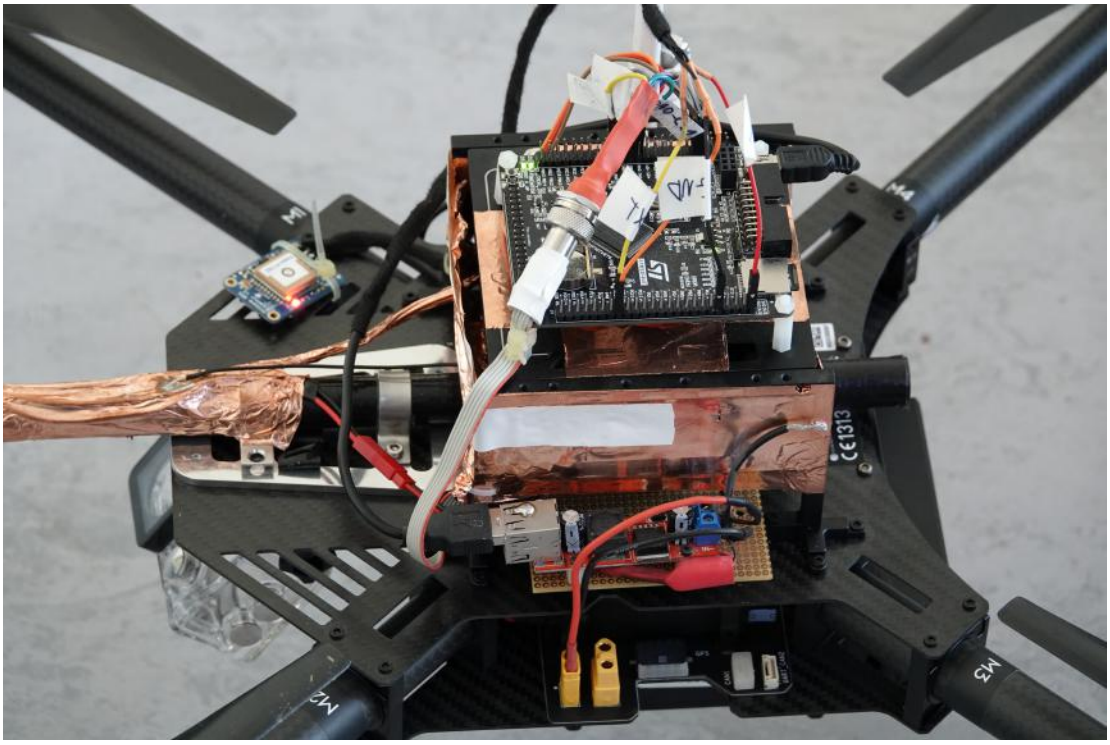


Figure 18: STM32-board on the Neurocopter and copper shielding

The small preamplifier-board is even more sensitive to noise than the amplifier board. It must be placed as close to the bee as possible which means it is also close to the gimbal motors. Gimbals measure their orientation and do up to 2000 corrections per second [16]. The motors are controlled via Pulse-Width Modulation (PWM). That is why they create strong high frequency noise. To shield the preamplifier-board as much as possible, it is in the metal structure of the bee-holder. The aluminum plate which covers the preamplifier-board has a gap smaller than a millimeter. That caused strong noise. So, the whole plate and the gap was covered with copper tape (see figure 19 upper part with copper foil). The noise created by the gimbal is now around as strong as the brain activity but can be filtered via subtracting the two channels from one another (differential measurement).

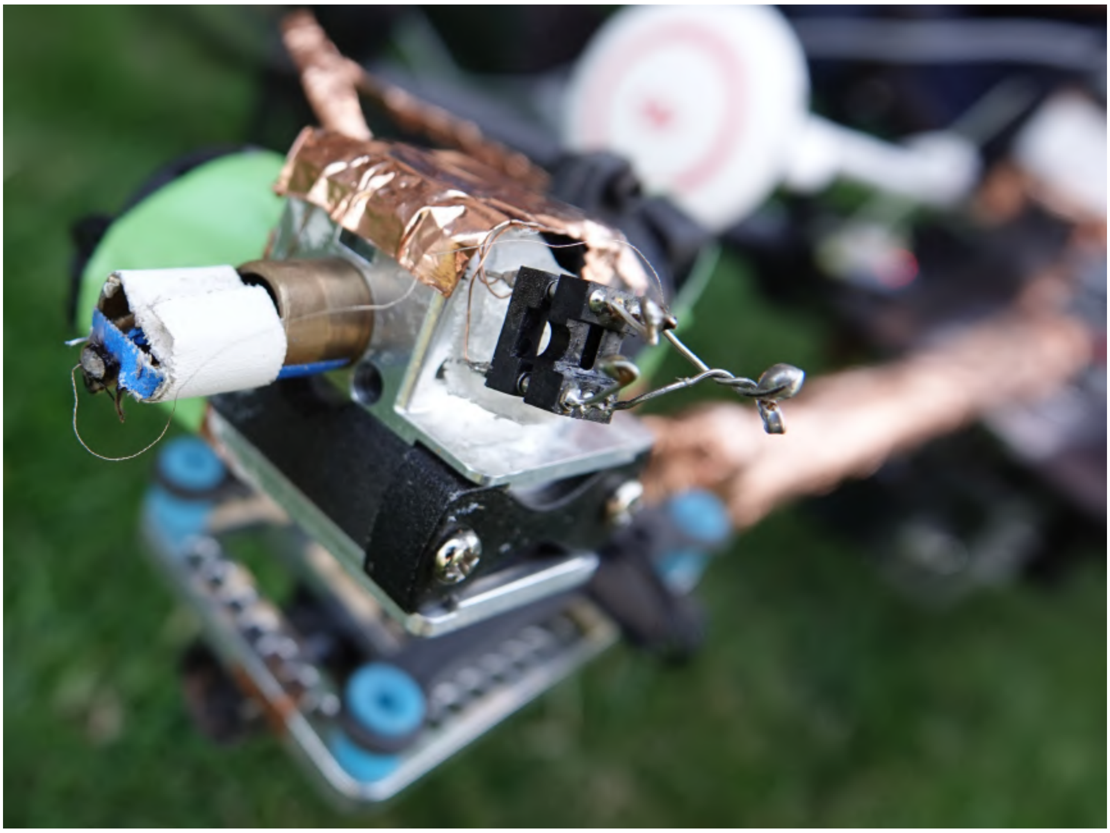


Figure 19: Honeybee (only the head is visible) on the Neurocopter with electrode

4.7 Preparing a bee for the Neurocopter

Preparing a bee for electrophysiological measurements is a long process. First, a bee needs to be caught which is a problem in winter. Then it needs to be cooled down so it does not move (see figure 20).

Figure 20: Cooled down bee

After approximately 10 minutes of cooling, the bee can be put in the bee holder on the Neurocopter. The Neurocopter stands next to a Faraday cage while the gimbal is insight the cage (see figure 21).

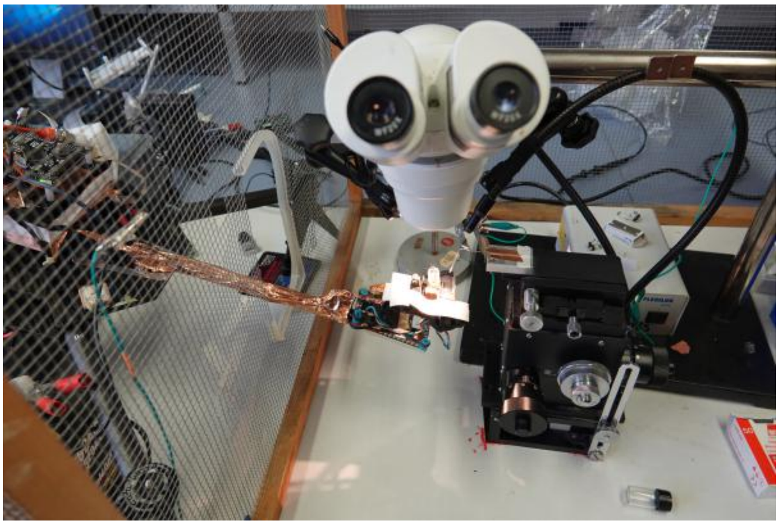


Figure 21: Bee in the Neurocopter bee holder in the preparation setup

Then the antennae of the bee have to fixated with wax to prevent the bee touching the preparation area. Now the bee's head can be opened. When the brain is visible (see figure 22) and not too hard, the micromanipulator is used to move the electrode into the brain.

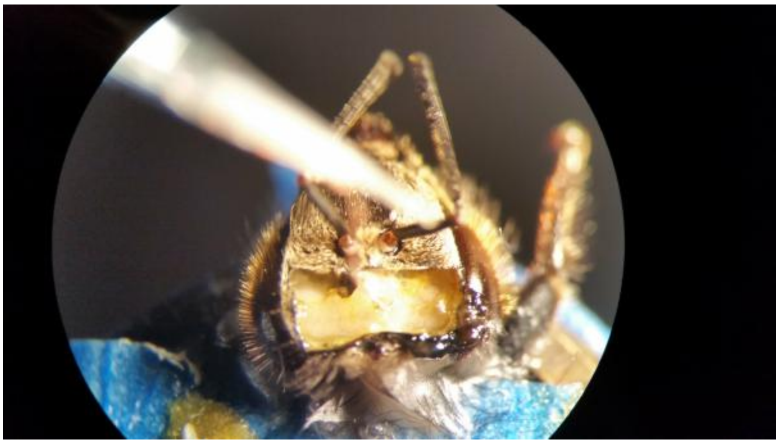


Figure 22: Bee brain and electrode (out of focus) photographed through a microscope

To find the right brain region, a stereo microscope is used. The Neurocopter is powered by its flight batteries and the analog output of the amplifier-board is connected to an audio amplifier and loudspeakers to make the signal audible. The signal can also be displayed using the Raspberry Pi OBC and the *Live View and Calibration Software* (see 4.8). If there are good signals (neuronal spikes sound like popping popcorn) and the amplitude is at least 2x over the noise, the electrode needs to be glued onto the brain. The drying glue can easily move the electrode. If the electrode moved too much, the neural recordings are too weak and a new bee needs to be prepared. If the gluing is successful, the electrode can be separated from the micromanipulator using a soldering iron (The electrode is glued to the micromanipulator with wax). Then the Neurocopter gimbal can be disconnected from the magnetic fixation and the Neurocopter can be taken out of the preparation setup. Now the bee should be fed with sugar water.

4.8 Software for Live View of the Neural Recordings and Calibration

For preparing a bee or for calibrating and debugging the Neurosopter system, live feedback is required. Making the analog channel hearable is good while the biologist prepares the bee under the microscope but not enough to get information about signal to noise ratio and shielding. For calibrating and debugging, it is necessary to have visual feedback. An oscilloscope which is connected to the normal 220 V powerline produces too much noise and does not have the right functions to evaluate neuronal spikes. So, the *Live View and Calibration Software* was developed. The Raspberry Pi-based OBC reads out the amplifier-board's ADCs and sends the data via UDP (one UDP-datagram contains 500 values for each channel) over Wireless Lan[17] to the *Live View and Calibration Software*.

The software is developed with Python 2.7. For the GUI and plotting, PyQtGraph [5] is used. It consists of three windows:

[17] When the Raspberry Pi was connected to the network via a RJ45 Ethernet cable the measurements became extremely noisy.

A plot of the last received data (see figure 23). It also shows the difference of the two channels as a green plot. To get good results on the differential channel, the noise of the two raw-channels has to be similarly high. Therefore, the digital gain for each channel can be adjusted manually or determined automatically (See figure 26 *"Channel 0: Autoadjust to Channel 1"*). The algorithm adjusts the gain of one channel in a binary search until the difference of the variances of the two channels is minimal. The differential channel does not contain any motor noise. So, spikes below the noise level are becoming visible on this channel. The raw channels (red and yellow) in figure 23 contain so much noise that no brain activity is visible. But on the green graph, spikes of neuronal activity are clearly visible.

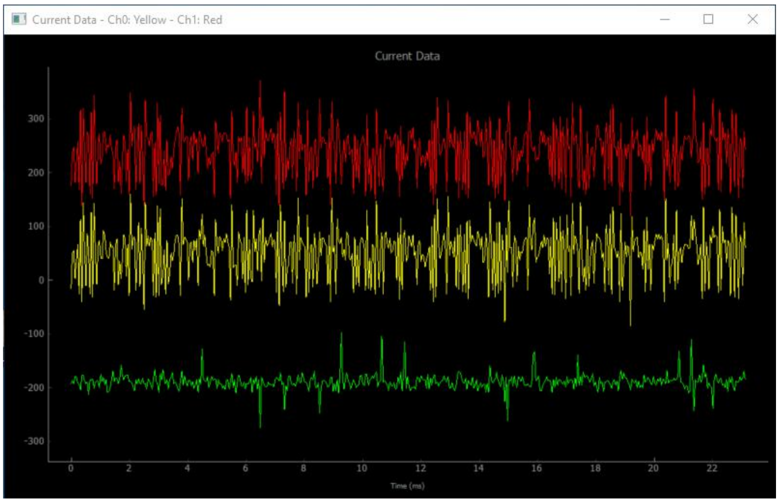


Figure 23: Screenshot of the last received data plotted in the *Live View and Calibration Software*. The two channels are colored red and yellow. The difference is plotted green. Spikes are only visible on the differential channel (green).

Thanks to PyQtGraph, it is possible to zoom into a graph, scale an axis logarithmically or to apply a FFT (Fast Fourier Transform) which is extremely useful to identify noise. For example, in figure 24 the FFT of the last received data shows a peak at 3.9 kHz.

Variances of all received data are plotted in the second window (see figure 25). The variances are calculated with the Python library numpy [11]. That can be useful to see changes in the noise floor or brain activity. For example, a drift in the variances is visible when the electrode moves while the glue is drying. It is also visible when spikes are triggered through light changes the variance.

In the **Settings Window** (see figure 26), there are different options for displaying and analyzing the data. Under *Scaling Options*, the size of the shown window can be incremented or decremented (SEC/DIV on an oscilloscope). In the channel options, the y-shifts and gains of both channels can be set or the automatic gain adjustment can be activated. *Analyses* show the variances of the two channels. Under *Trigger Options*, a rising edge trigger can be activated. If a measurement goes over the trigger level, it moves the whole plot in a way that the value causing the trigger is in the center. So every time a spike is detected, it moves the spike into the center and holds it for

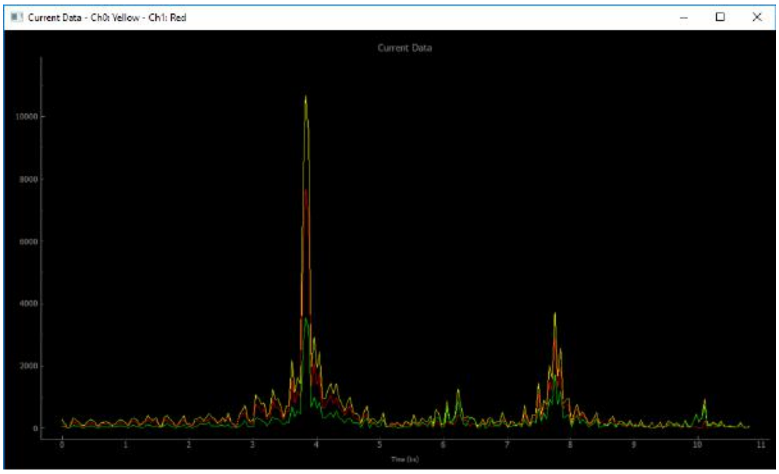


Figure 24: Screenshot of a FFT of the last received data.

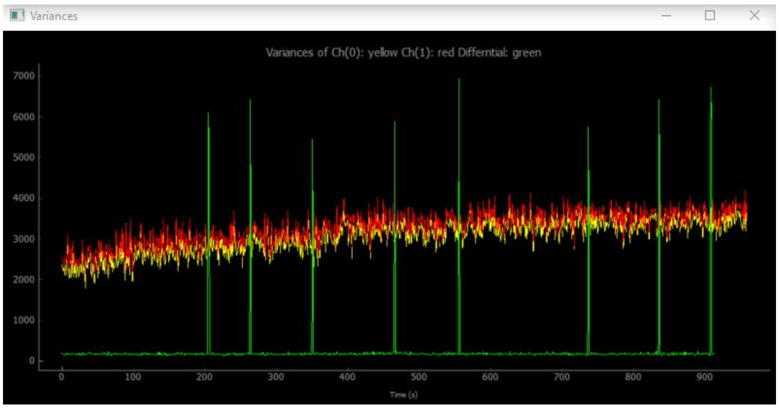
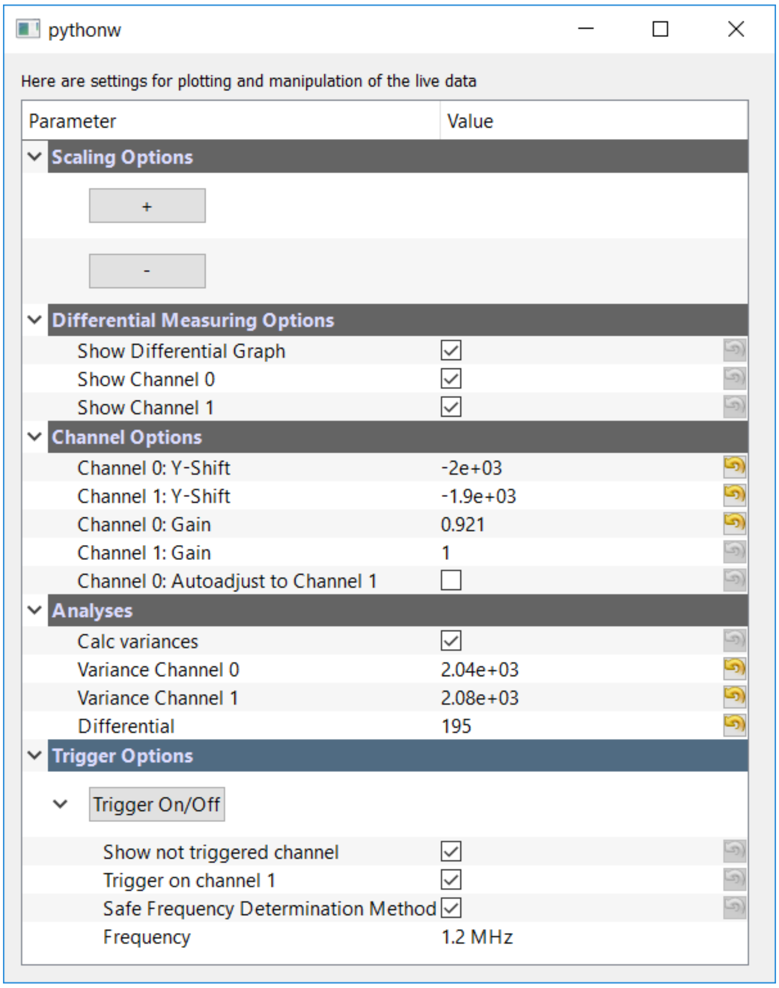


Figure 25: Screenshot of variances of all received data.

0.3 seconds. That helps to view spikes live but can also be used for testing the amplifier board with a function generator. When the trigger is activated, the frequency of the triggered function is calculated as shown in the *Trigger Options*.

The repository of the *Live View and Calibration Software* is available here:
https://git.imp.fu-berlin.de/ast/Master_Thesis_STM32_Neurocopter_SPI_Logger
/tree/master/MasterThesisNeurocopterRaspiUdpSampler



pythonw

Here are settings for plotting and manipulation of the live data

Parameter	Value
Scaling Options	
+	
-	
Differential Measuring Options	
Show Differential Graph	☑
Show Channel 0	☑
Show Channel 1	☑
Channel Options	
Channel 0: Y-Shift	-2e+03
Channel 1: Y-Shift	-1.9e+03
Channel 0: Gain	0.921
Channel 1: Gain	1
Channel 0: Autoadjust to Channel 1	☐
Analyses	
Calc variances	☑
Variance Channel 0	2.04e+03
Variance Channel 1	2.08e+03
Differential	195
Trigger Options	
Trigger On/Off	
Show not triggered channel	☑
Trigger on channel 1	☑
Safe Frequency Determination Method	☑
Frequency	1.2 MHz

Figure 26: Settings Window for the *Live View and Calibration Software*

4.9 Neurocopter data pipeline

To be able to get correlations, of the neural recordings and the drone's sensor data, the neuronal data needs to be analyzed to get spikes and synchronized with the drone telemetry data. The STM32 saves the neuronal data as binary files to its SD-Card. The DJI Matrice saves the sensor data as encrypted binary files to its internal memory. DatCon is used to convert the drone data to CSVs. These CSV-files are synchronized with the neuronal data with the *Neurocopter Binary Converter*. The resulting CSV-files can be imported in Spike2 where the spike sorting is done. That needs some manual work. The classified spikes and timestamps can be exported as TXT-files. With them and the synchronized drone and brain data, the *Neurocopter Binary Converter* can create a CSV containing classified spikes synchronized with the drone sensor data. Now it is possible to plot for example the drone flying speed against a spike rate.

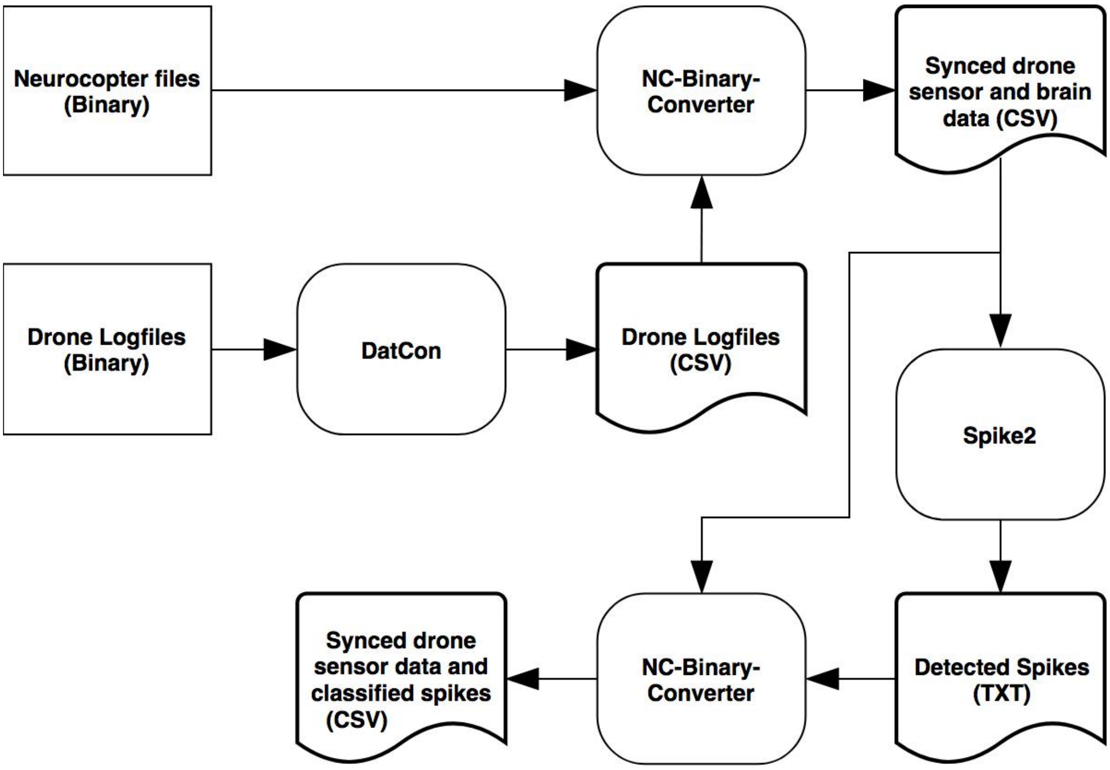

Neurocopter files (Binary)
NC-Binary-Converter
Synced drone sensor and brain data (CSV)
Drone Logfiles (Binary)
DatCon
Drone Logfiles (CSV)
Spike2
Synced drone sensor data and classified spikes (CSV)
NC-Binary-Converter
Detected Spikes (TXT)

Figure 27: Settings window for the *Pipeline of Neurocopter data.*

5 Experiments

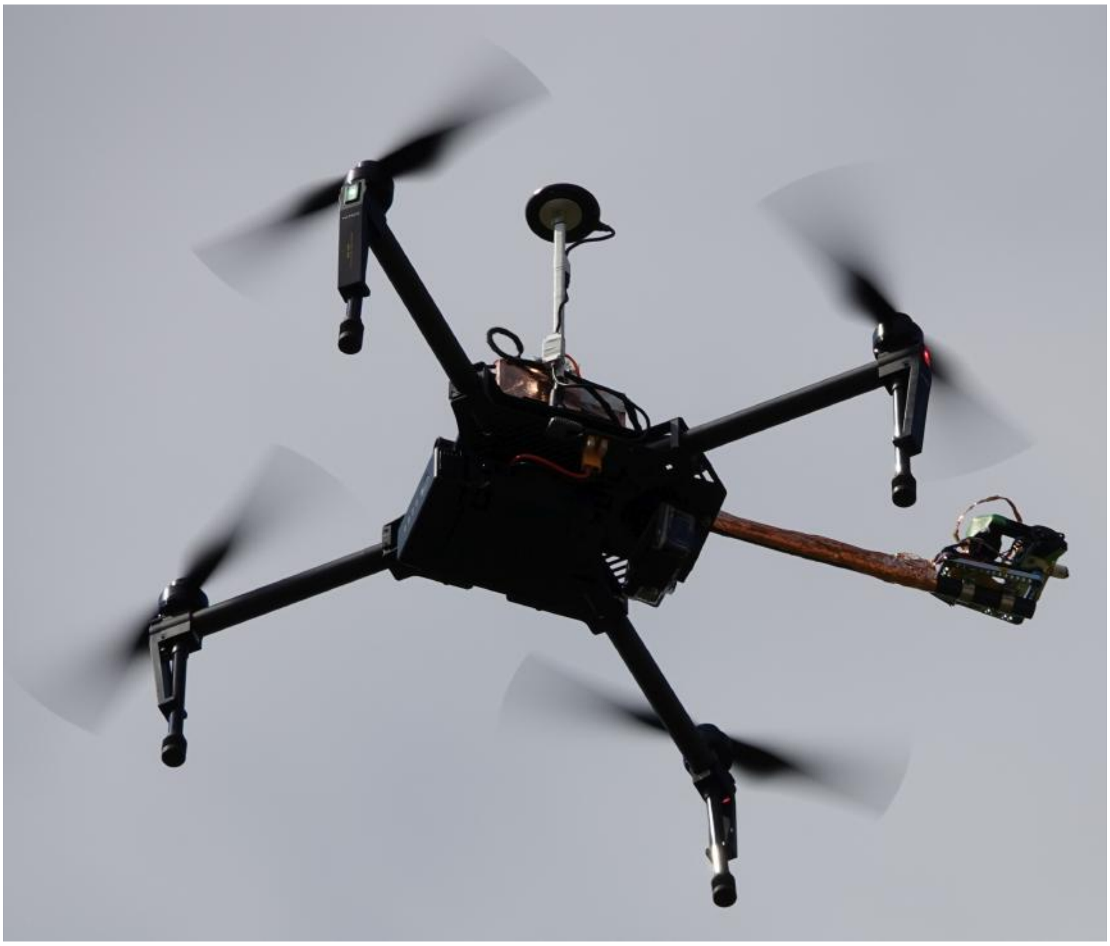


Figure 28: The Neurocopter in air doing extracellular recordings of a honeybee

5.1 Test Flights in Brandenburg

While integrating different parts into the Neurocopter system, it was tested on fields in Brandenburg. It was especially unclear how the cantilever would effect the flight stability. After a first test flight, it was found that the cantilever causes the copter to swing dangerously. So, the cantilever had to be shortened and mounted not only at the front but also at the back of the copter (see subsection 3.4). So, after ensuring the copter with the cantilever could hover, stable flying in all directions, curves and breaking was tested. The speed was increased during these maneuvers.

5.2 Experiments in Hesse in 2017 without extracellular recordings

The Neurocopter is a flying robotic platform to study the neural correlates of honey bee navigation. So, it needs to be proven that honeybees flying on the Neurocopter have a similar experience as when they are flying on their own. Therefore, two types of experiments were done:

The **Wing beat duration experiment** should prove that honeybees show longer flight behavior when they fly on the Neurocopter compared to hovering or standing on the ground.

In the **Navigational capabilities during flight experiment**, it was tested whether a honeybee which was flown by the copter and released 300 m away from the hive returned to the hive faster than a honeybee which was transported to the release site in a box. Those experiments were done by Prof. Dr. Tim Landgraf and Benjamin Wild. They showed that bees flown on the copter found their way back to the hive significantly faster than untreated bees [28].

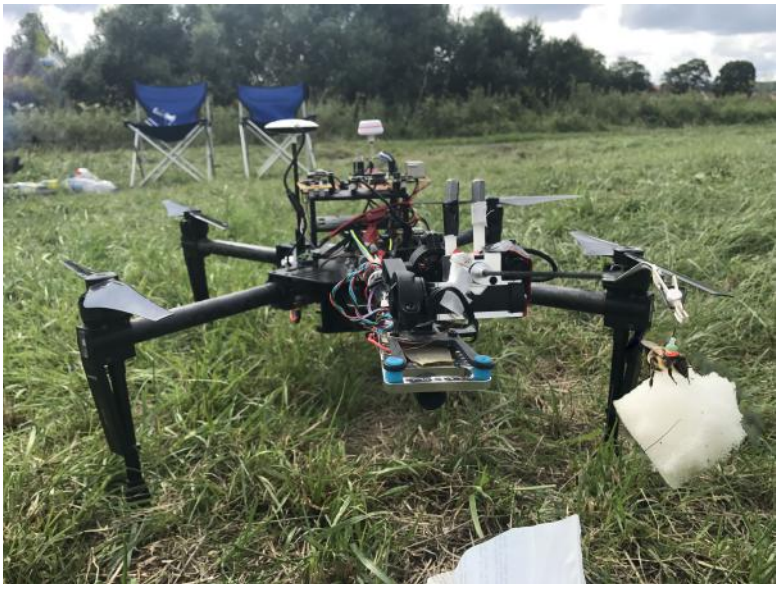


Figure 29: A bee attached to the Neurocopter in Amöneburg (Hesse)

5.2.1 Wing beat duration

The honeybee had an antenna for the harmonic radar [24] glued to its thorax. Via the antenna, it was attached to the Neurocopter (see figure 29). The copter's gimbal was activated to stabilize the flight. The angle of the gimbal was controlled by a microcontroller and showed on a display, so different angels could be tried. In the beginning, the bee was holding a piece of foam which simulated the ground. The foam was connected through a string to the ground. In **trial 0 (no flight)**, the foam was removed fast. That triggered the tarsal reflex, so the bee moved the abdomen in flight position and started beating its wings. The time during which the abdomen was in flight position and the wings were moving was measured. In **trial 1 (forward flight)**, the copter was launching forward and flying with 15 km/h at low altitude. While launching, the foam was removed via the string attaching the foam to the ground. Again, the tarsal reflex was triggered. Like in trial 0, the time during which the abdomen was in flight position and the wings were moving was measured. In **trial**

2 (hovering), the drone launched like in trial 1 but was just hovering at 1m altitude and not flying forward. After each trial, there was a one-minute recovery time for the bee. 18 trials (six of each kind) were done with each bee. The order of the trials was randomized. The bees were filmed with a camera mounted on the copter. The live video was sent via an analog 5.8 GHz FPV-sender to the ground and displayed on a monitor. Figure 30 is a screenshot of a bee filmed by that camera.

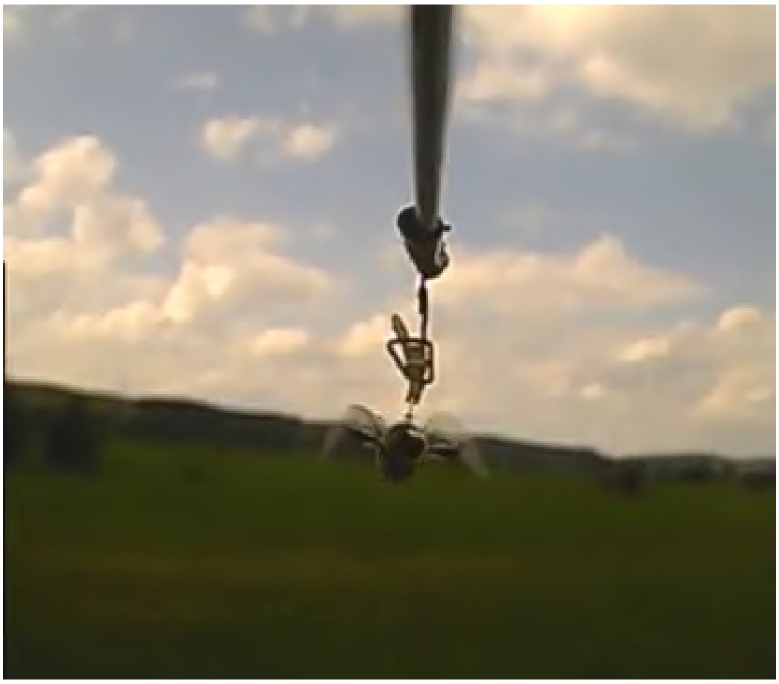


Figure 30: A honeybee attached to the Neurocopter showing flight behavior during a forward flight. The rolling shutter effect lets the bees wings look curved.

Results and discussion Figure 31 shows all flight durations for the three different trials. It is obvious that the wing beat duration of the *forward flights* is significantly longer than for the *no flight* and *hovering* trials. The p-value for *forward flights* and *no flight* is $1.01 * 10^{-20}$ (Mann-Whitney-Wilcoxon (MWW) RankSum test). For *forward flights* and *hovering* it is $8.19 * 10^{-18}$. So, it is significant that the bee shows flight behavior on the Neurocopter if it is flying. The p-value for *no flight* and *hovering* it is 0.19 which is not significant but indicates that hovering caused longer flight behavior than standing on the ground. The violin plots 41 in the appendix indicate the weak difference of the distributions.

Preparing a bee for extracellular recordings is a big effort. That is why such a bee should be used for as long as possible. So it is interesting to see if the bee gets exhausted or adapts to the flight on the copter. For the experiments, every bee was used for 18 trials. Figure 32 shows box plots of all first wing beat duration times of a series (all flights with one bee) and of all last flights of a series. The number of trials is too small for significant results (p-value: 0.29) but it indicates that the bee adapts to the drone flights. So, the adaption should be taken into account for further experiments.

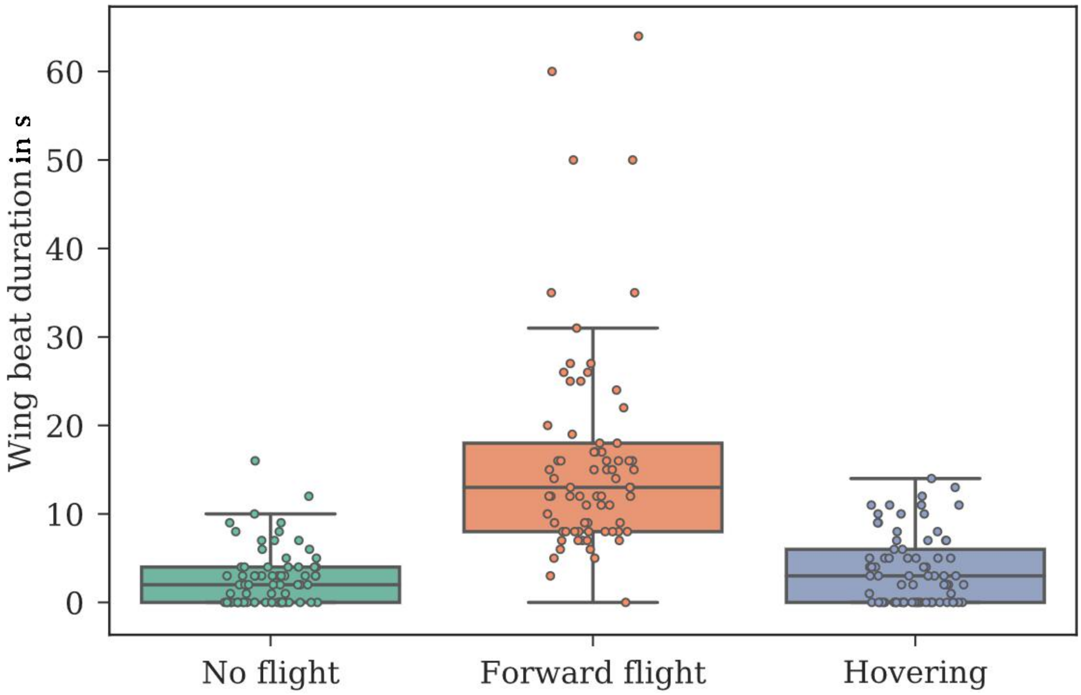


Figure 31: Box plots of the wing beat duration times

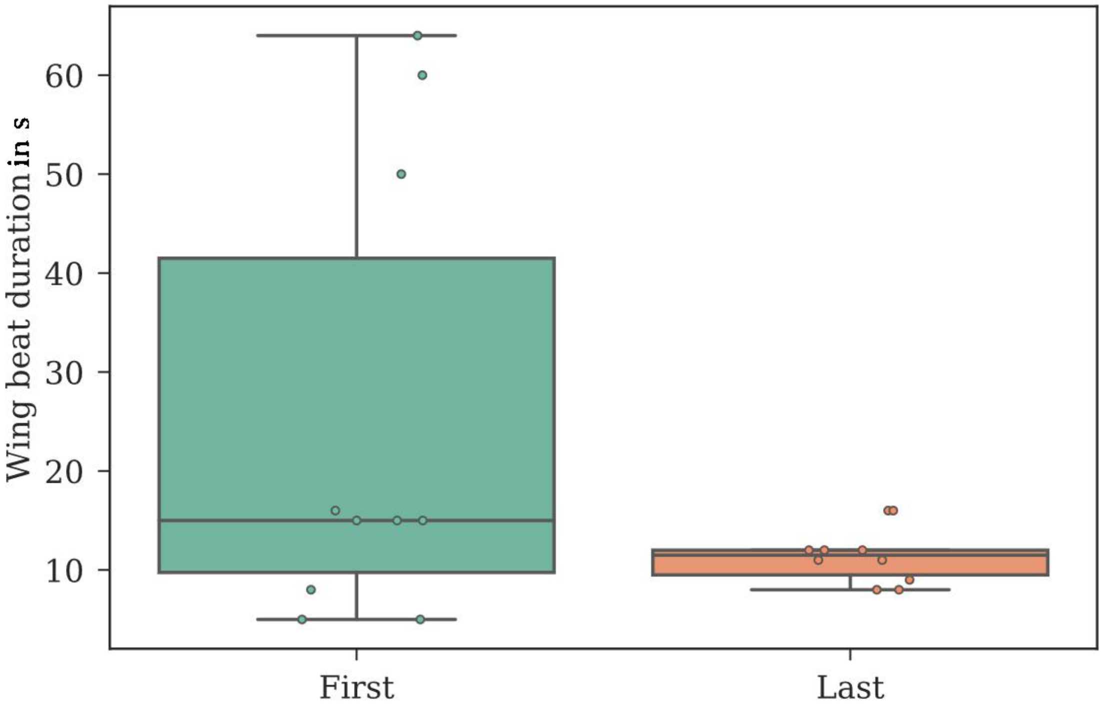


Figure 32: Wing beat duration times of the first and last flights of a series

5.3 Experiments in Berlin 2018 with extracellular recordings

5.3.1 Planning and preparations

Flight Pattern The Neurocopter is able to do preprogrammed flights. But for the first experiment, it was decided to fly manually in a GPS-stabilized flight mode, so the copter pilot could react faster if problems had occurred. The flight pattern was to fly straight lines over a path between the fields, to make sharp 180 degree turns and to fly straight lines parallel to the path over fields. Experiments from Prof. Dr. Randolf Menzel with his harmonic radar indicate that honeybees use such paths for orientation. So, there should be a difference in the brain activity of a honeybee flying over a path to a honeybee flying over fields. The sharp turns are causing optical flow changes which should trigger changes in brain activity as well. The altitude was 20m above ground, except for the launch and landing phase. That was low enough to create visual stimuli for the bee and high enough to ensure a safe flight.

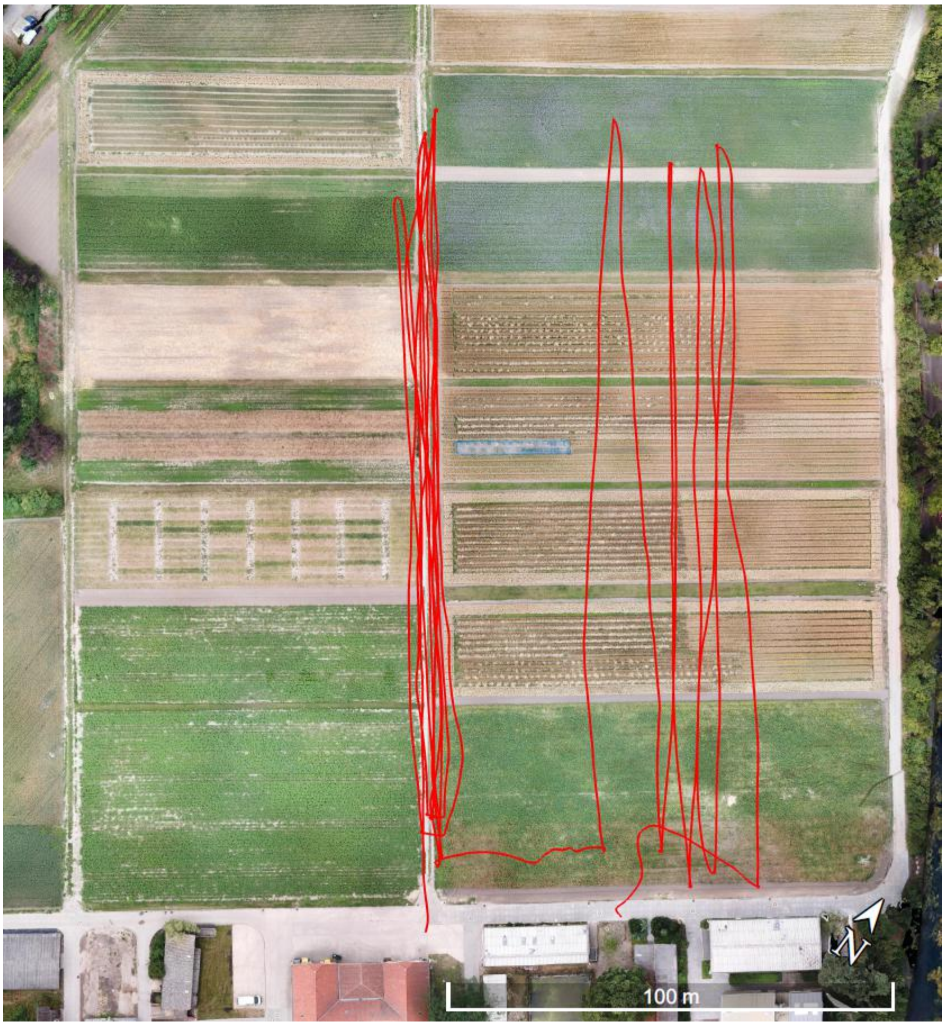


Figure 33: The flight track of the first experiment. The background map was created by the author with a drone (for the mapping process see section B.1 in the appendix).

Electrode position The brain of a honeybee is symmetric. It has two Mushroom Bodies (see figure 34). Each Mushroom Body has an α-lobe. An α-lobe has two exits. The α-lobe has many inputs and two outputs: The α-exit and the β-exit [32] [27]. For this trial, the electrode was inserted into the honeybee's α-exit of the left α-lobe (see red cross in figure 34). It was decided to place the electrode there because the Mushroom Body does multi modal sensor integration and is much easier to reach than the Central Complex.

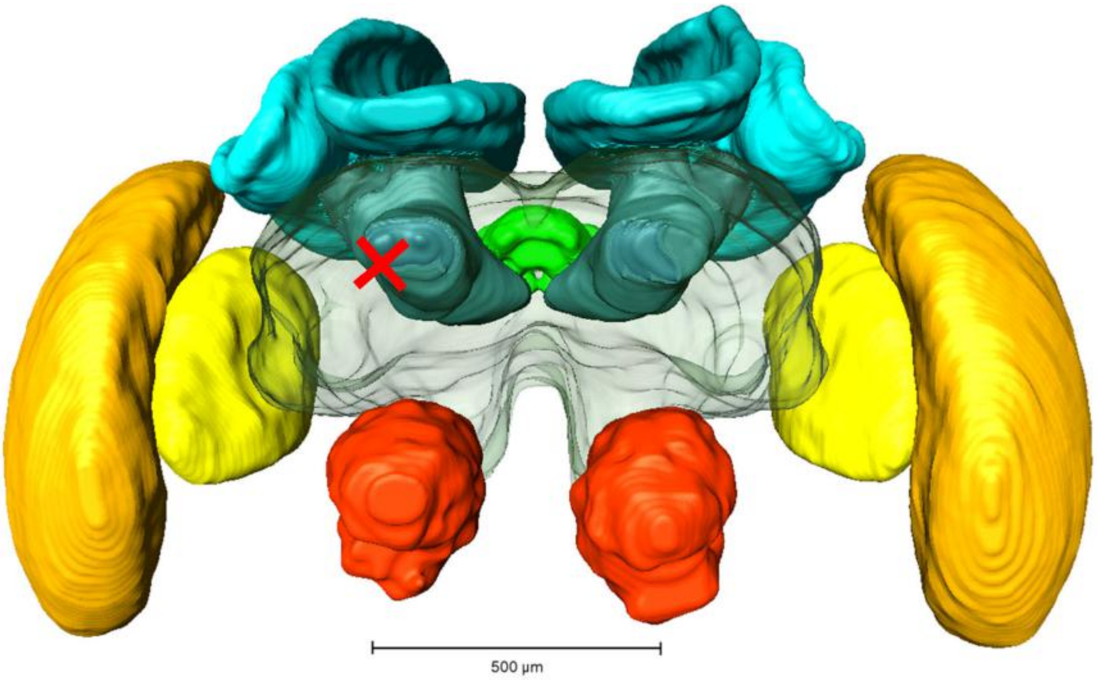


Figure 34: Electrodes position in the bee's brain marked with a red cross. The colors mark different parts of the brain: *Red, Antennal lobes, AL; Blue, Mushroom bodies, MB (comprising medial and lateral calyx and the pedunculus); Green, central complex, CB (comprising upper and lower central complex and protocerebral bridge); Yellow, Lobula; Orange, Medulla.* (Source of the figure and description without electrode marking is from [34])

5.3.2 Performing the experiments

After the forager honeybee was successfully prepared (for bee preparation, see 4.7), a break of 90 minutes was taken to let the glue dry and to let the bee recover. After these 90 minutes, the measured spikes were still correlated to light. So, the copter was taken to the Julius Kühn Institute and prepared for the flight (that took another 20 minutes). After performing two flights, the copter was taken back to the institute and the bee was fed again. Seven hours after preparation, strong brain activity was still measurable. And even in the next morning, there were weak measurable spikes.

5.3.3 Results

The brain data recorded by the STM32 and the drone telemetry data were processed and merged as explained in subsection 4.9. Three units (different kind of spikes) could be separated with Spike2 (see figure 40 in the appendix). Figure 35 shows the (yaw)-rotation of the Neurocopter and the spike rate of unit 1. It is notable that every time the copter rotates (see blue peaks), the spike rate has a peak (see red plot) as well. The spike rate has always some minor activity but there was only one peak while the copter was not rotating (720-750 s). A correlation between the rotation direction or rotation speed of the copter and the spike rate could not be found. The spikes classified as unit 2 have some similarities to the unit 1 spikes but the general rate of unit 2 spikes is much lower and noisier. Plotted together with the (yaw)-rotation of the Neurocopter (see figure 38 in the appendix), there are some risings in the spike rate when the drone rotates. But no correlation can be shown. The unit 3 spike rate shows some lower frequency changes which is not correlated with the drone telemetry data or the other spike rates.

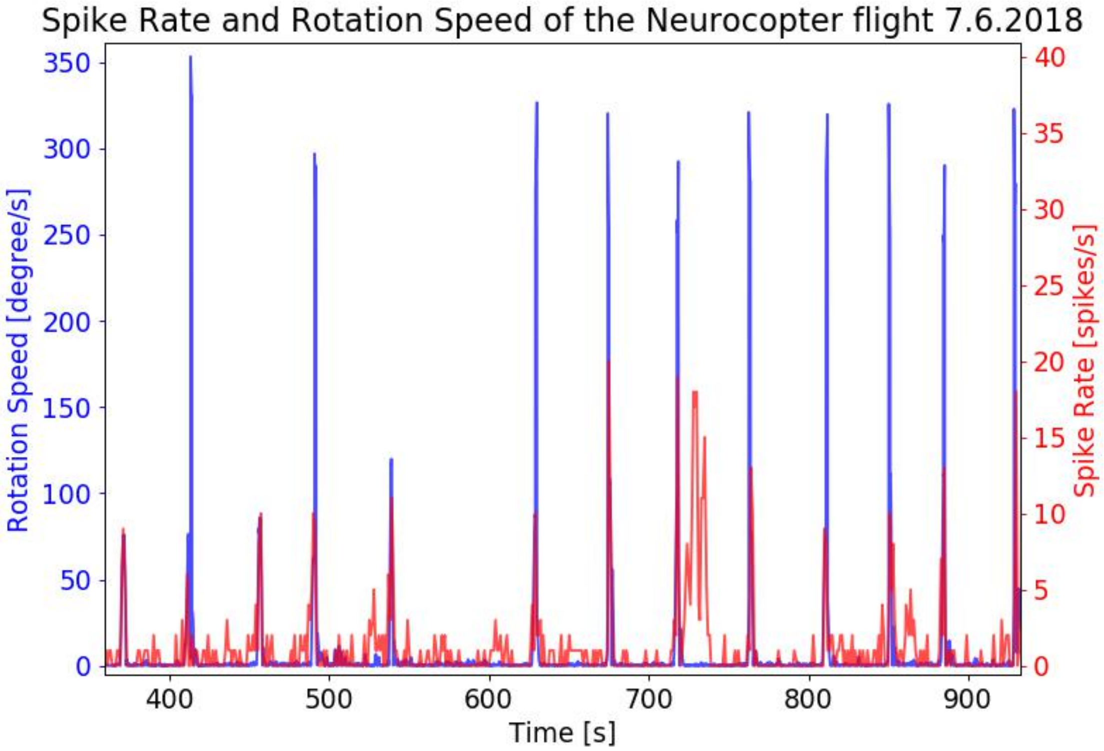
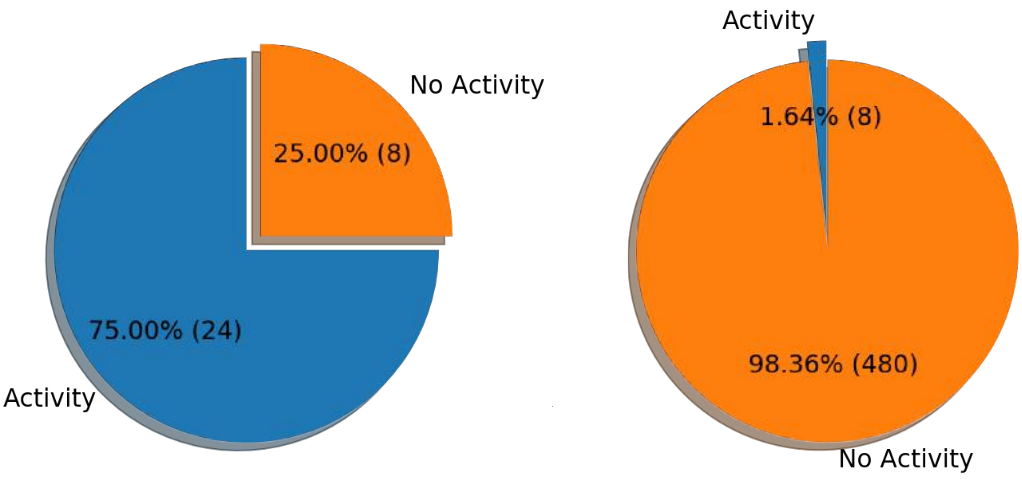


Figure 35: Spike Rate (Unit 1) and Rotation Speed of the Neurocopter flight 7.6.2018

Figure 36a is a pie chart showing the time (in seconds) of high brain activity[18] and no high brain activity during the flight when the drone was rotating. So, 75% of the time, when the drone was rotating, there was also high brain activity.

[18] A spike rate of at least 5 spikes per seconds was chosen as high brain activity.



(a) **Copter rotating**

(b) **Copter not rotating**

Figure 36: Time in seconds with Activity during Copter Rotations (a) or during flying forward (b)

Only in 1.6% of the time the drone was flying forward (and not rotating), there was high brain activity (see figure 36b). The Chi-Square test result is 279.8529. This is a p-value of 0.00001. So, there is a strong correlation of the copter's rotation and the brain activity.

5.3.4 Discussion

The experiment shows a strong correlation of the drone rotation and the brain activity. In the laboratory, the honeybee reacted similarly to sidewise light movement. So the spikes were probably triggered through the visual stimulus and by other parameters like for example the acceleration. To test a flight where the bee is on the Neurocopter in a dark box could be done. The experiments proved that the Neurocopter system is able to do extracellular recordings of a honeybee while flying.

6 Discussion

The aim of this thesis was to create a flying robotic platform for extracellular recordings of honeybees, including all necessary hardware and software, in order to find neural correlates of honey bee navigation in the recorded data.

The field experiments in Brandenburg and Hesse showed that a honeybee, flown by the Neurocopter system, has an experience similar to a flight on its own. This means that the Neurocopter system is a new tool to get data about flying honeybees.

Further this thesis presents the first recordings of neural activity ever made in a living honeybee flown on a copter (see figure 42) and the further analysis of the experiments show neural correlates of the copters rotation (see subsection 5.3.3). The Neurocopter system proved to be flexible and reliable during field experiments. It can be used in further experiments for studying new aspects of long-range navigation in flying honeybees.

The Neurocopter project requires knowledge, experience and skills in different fields. A microcontroller is integrated in an analog measurement system on a flying drone, collecting data of living animals. To succeed, this project needed a lot cooperation between biologists and computer scientists. In this project, many of the involved persons had experience in multiple areas of the project which made the cooperation much more efficient and helped whenever fast solutions for complex problems were needed.

7 Outlook

With complex projects like the Neurocopter it is always possible to improve some parts or to start exploring new aspects.

7.1 New and ongoing experiments

7.1.1 Circle pattern

New experiments are currently in progress. In one experiment, the drone flies in a circle for as long as the battery allows. With that experiment, a correlation between the neural activity and angle to the sun or to some landmarks can be found. Maybe at some point, the bee synchronizes its sun compass. Figure 43 in the appendix shows the first track of such a flight.

7.1.2 Cover parts of the bee's field of view

In the first experiments (see 5.3) a correlation between the drone rotation and brain activity could be shown. It could be tested now what in the honeybee's field of view caused the brain activity. So, the upper part of the field of view can be covered so the honeybee does not see the sun. Or the horizon and sun can be covered to see if nearby structures can be used by the honeybee to recognize rotation.

7.1.3 Overflight over a trained feeder

In a more advanced experiment, the honeybees are first trained to a food source placed on the fields of the Julius Kühn-Institut. All honeybees which visit the food source are marked with a special color. Now such marked honeybees are taken from the hive for Neurocopter flights. The flight pattern (see figure 37) begins with the same course the honeybees do when they fly from their hive to the feeder. Then it goes over the feeder and makes a half circle to fly along a path again over the feeder. After another half circle it goes again over the feeder (not parallel to any path on the fields) and after another half circle it goes back to the starting point. The main purpose of this experiment is to study how and when the honeybees identify the known food source. But the flight pattern also enables to get data about the neural activity when flying over a path or not. And lastly, the long curves can help to get data about the bee's compass.

Figure 37 shows a sample flight where the spike rate is coded in the flight tracks color. Similar to the first experiment with neural recordings (see subsection 5.3), the spike rate is high on turns (red). This flight, as well as others, indicate that the honeybee recognizes the trained feeder. On the feeder overflight from west to east, there is nearly no neural activity till the feeder is reached, and after the overflight, the activity is still high. More flights are needed to get significant results. It is also noteworthy that the spike rate rises when the bee is over a field's border. To investigate this further, raster flights like those done for the mapping (see section 46) are planned.

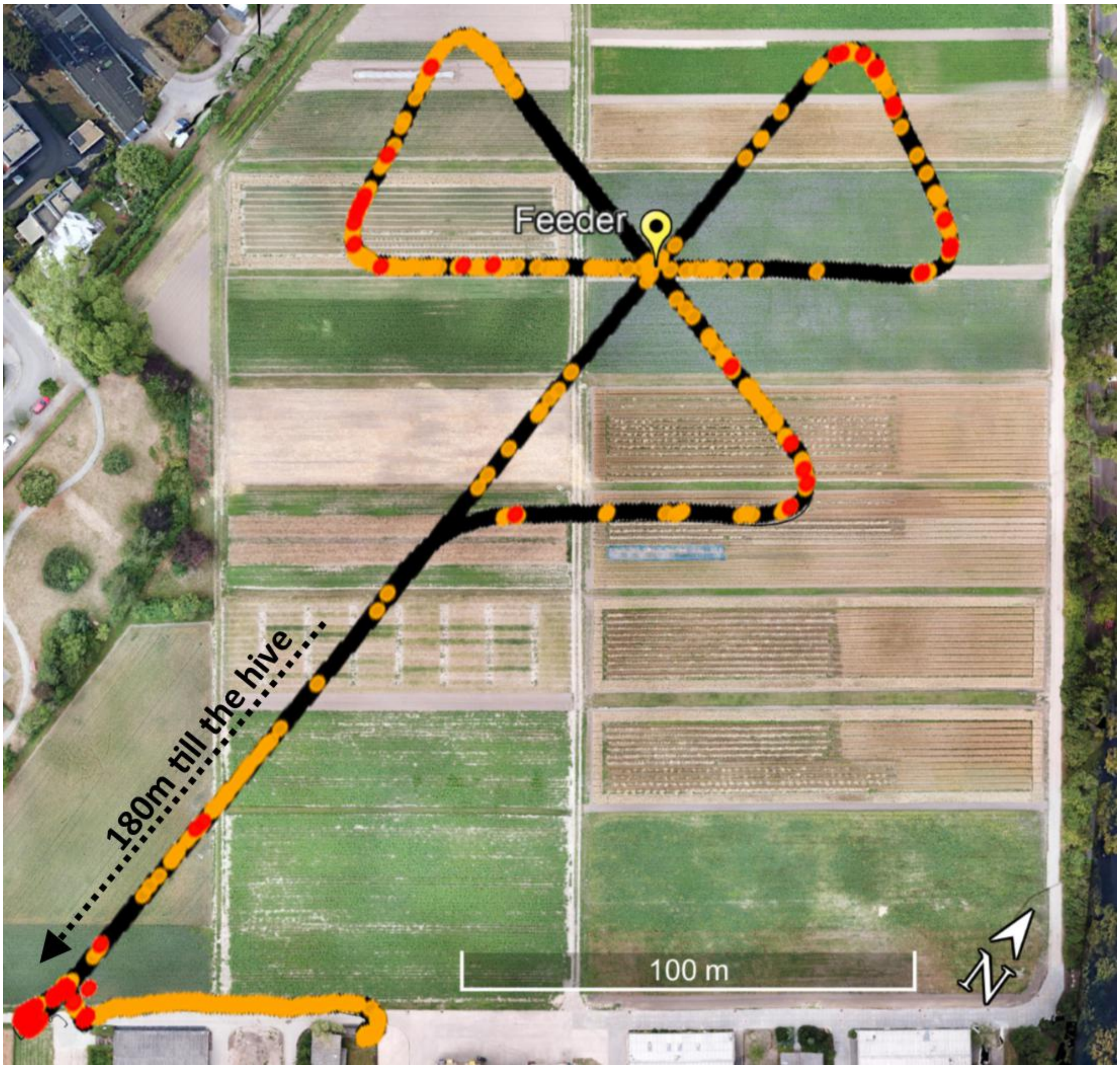


Figure 37: Flight pattern of an approach and overflights with honeybees over a known feeder flown on the 20.07.2018. The altitude above ground is 19 m and the speed is 20 km/h. The manual flight between the starting point (lower left corner) of the automatic pattern is plotted too. Black means a low spike rate (< 20 Hz), yellow a medium spike rate > 20 Hz and red a high spike rate (> 40 Hz). The background map was created by the author with a drone (for the mapping process, see section B.1 in the appendix). For an overview map containing the hive area, see figure 46 in the appendix.

7.2 Hardware

The schematics of the amplifier-board can be changed so that every ADC has separate data lines to the microcontroller. This would simplify the parallel measurement and ADC readout (see subsection 4.4). The sample rate would increase by around 50%. Further, the amplifier-board layout should be changed. In the current version, there is no ground plane, the traces have curves that are too sharp, and the analog and digital parts are not separated enough. Dr. Benjamin Paffhausen is planning a new design which will improve all those problems. A method paper for an open source extracellular recording system containing the new designed amplifier-board and the Neurocopter on-board computers code (STM32 and Raspberry Pi 3) is planned.

If the bee attached to the flying copter has all freedom to move, it shows flight behavior. This flight behavior could be analyzed[19] and the direction and speed where the bee wants to fly could be interpreted. Then the copter could be controlled by the attached bee. The feedback-loop must be very fast, so the bee thinks it is flying on its own. That would make the flight experience of the bee much more realistic and enable completely new experiments.

7.3 Software

DJI offers a SDK to connect the STM32F407VET-Microcontroller with the DJI Matrice. For this thesis, code was developed to read the copter's telemetry data via the DJI-SDK. But the DJI-SDK can also be used to send control commands. So, the copter could be controlled based on the bee's wing movement or based on the current neural activity. Or for low altitude flight, external sensors like a LIDAR could be used to hold the altitude above ground. At the moment, the minimum safe altitude for long range flights is 10-15 m which is too high for some experiments.

When the amplifier board ADC data lines were separated, the STM32 could use DMA to read out the ADCs. That would save processing time on the CPU which could be used for other tasks like reading out where the bee wants to fly and controlling the copter. Furthermore, much higher sampling rates would be possible. But the current limitation for the sample rate is the ADC. So a new ADC would be required.

The amplifier board designed for the Neurocopter-project works at least as well as the current neural recording amplifiers used by the Neurobiology Institute but is much cheaper and smaller. So, if the STM32 would send the live-data via USB and if a driver for the STM32 and spike2 would be developed, the amplifier-board and the STM32 would be a great low cost open source system for extracellular recordings. Such a system would also be useful for preparing a bee for Neurocopter flights.

[19] For a bumblebee flight simulator [39] Stefan Walter did measurements of the electric fields around the wings of bees. He could say if the bee wants to fly right or left.

8 List of abbreviations

ADC	Analog Digital Converter
CAD	Computer-Aided Design
CS	Chip Select
CSV	Comma-separated values
DMA	Direct Memory Access
Ephys	Electrophysiology
ESC	Electronic Speed Controller
FFT	Fast Fourier Transform
FPV	First Person View
GPS	Global Positioning System
I2C	Inter-Integrated Circuit
JTAG	Joint Test Action Group
LSB	Least Significant Bit
MSB	Most Significant Bit
OBC	On Board Computer
PWM	Pulse-Width Modulation
RTC	Real Time Clock
RTF	Ready To Fly
SBC	Single-Board Computer
SDO	Serial Data Out
SDIO	Secure Digital Input Output
SDK	Software Development Kit
SPI	Serial Peripheral Interface
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VTOL	Vertical Take-Off and Landing

List of Figures

Bibliography

[1] Dji matrice 100. *SZ DJI Technology Co., Ltd.*, 2018.

[2] Simplebgc 32bit 3-axis - software user manual. *Basecamelectronics*, 2018.

[3] Hansjochem Autrum and Marieluise Stoecker. Die verschmelzungsfrequenzen des bienenauges. *Zeitschrift für Naturforschung B*, 5(1):38–43, 1950.

[4] Denes Budai. Ultralow-noise headstage and main amplifiers for extracellular spike recording. 48, 11 2003.

[5] Luke Campagnola. Pyqtgraph-scientific graphics and gui library for python, 2016.

[6] N Cha. Fatfs generic fat file system module, 2014.

[7] James F. Cheeseman, Craig D. Millar, Uwe Greggers, Konstantin Lehmann, Matthew D. M. Pawley, Charles R. Gallistel, Guy R. Warman, and Randolf Menzel. Way-finding in displaced clock-shifted bees proves bees use a cognitive map. *Proceedings of the National Academy of Sciences*, 111(24):8949–8954, 2014.

[8] Allen Cheung, Matthew Collett, Thomas S. Collett, Alex Dewar, Fred Dyer, Paul Graham, Michael Mangan, Ajay Narendra, Andrew Philippides, Wolfgang Stürzl, Barbara Webb, Antoine Wystrach, and Jochen Zeil. Still no convincing evidence for cognitive map use by honeybees. *Proceedings of the National Academy of Sciences*, 111(42):E4396–E4397, 2014.

[9] Linear Technology Corporation. Ltc1864/ltc1865, upower, 16-bit, 250ksps (1- and 2-channel adcs in msop), 2007.

[10] Benjamin Daumenlang. Anbindung einer flugsteuerungsplatine an einen flugsimulator. *BioRobotics-Lab*, 2013.

[11] Numpy Developers. Scientific computing tools for python-numpy, 2018.

[12] Analog Devices. Dual-channel digital isolators - data sheet - adum1200/adum1201. 2016.

[13] H.E. Esch, Shaowu Zhang, M.V. Srinivasan, and Juergen Tautz. Honeybee dances communicate distances measured by optic flow. 411:581–3, 06 2001.

[14] YUNEEC Europe. Yuneec sdk - software development kit. *YUNEEC Europe*.

[15] Karl von Frisch. Tanzsprache und orientierung der bienen. *Springer Berlin*, 1965.

[16] Gremsy. Gremsy t3 - user manual. 2017.

[17] Reid R Harrison, Ryan J Kier, Anthony Leonardo, Haleh Fotowat, Raymond Chan, and Fabrizio Gabbiani. A wireless neural/emg telemetry system for freely moving insects. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 2940–2943. IEEE, 2010.

[18] Tim Landgraf, Benjamin Wild, Tobias Ludwig, Philipp Nowak, Lovisa Helgadottir, Benjamin Daumenlang, Philipp Breinlinger, Martin Nawrot, and Raúl Rojas. Neurocopter: neuromorphic computation of 6d ego-motion of a quadcopter. In *Conference on Biomimetic and Biohybrid Systems*, pages 143–153. Springer, 2013.

[19] Richard B Langley. Nmea 0183: A gps receiver interface standard. *GPS world*, 6(7):54–57, 1995.

[20] Tobias Ludwig. Neurocopter - eine fliegende experimentierplattform zur erforschung der hirnaktivität von honigbienen. *BioRobotics-Lab*, 2016.

[21] Fyhn Marianne, Hafting Torkel, Witter Menno P., Moser Edvard I., and Moser May-Britt. Grid cells in mice. *Hippocampus*, 18(12):1230–1238.

[22] Inc Measurement Specialties. Hpc199_3 htu21d(f) sensor datasheet. 10 2013.

[23] Randolf Menzel. Searching for the memory trace in a mini-brain, the honeybee. *Learning & Memory*, 8(2):53–62, 2001.

[24] Randolf Menzel, Uwe Greggers, Alan Smith, Sandra Berger, Robert Brandt, Sascha Brunke, Gesine Bundrock, Sandra Hülse, Tobias Plümpe, Frank Schaupp, et al. Honey bees navigate according to a map-like spatial memory. *Proceedings of the National Academy of Sciences*, 102(8):3040–3045, 2005.

[25] Randolf Menzel, Andreas Kirbach, Wolf-Dieter Haass, Bernd Fischer, Jacqueline Fuchs, Miriam Koblofsky, Konstantin Lehmann, Lutz Reiter, Hanno Meyer, Hai Nguyen, et al. A common frame of reference for learned and communicated vectors in honeybee navigation. *Current Biology*, 21(8):645–650, 2011.

[26] Randolf Menzel, Konstantin Lehmann, Gisela Manz, Jacqueline Fuchs, Miriam Koblofsky, and Uwe Greggers. Vector integration and novel shortcutting in honeybee navigation. *Apidologie*, 43(3):229–243, 2012.

[27] PG Mobbs. Neural networks in the mushroom bodies of the honeybee. *Journal of insect physiology*, 30(1):43–58, 1984.

[28] Julian Petrasch, Benjamin Wild, Benjamin Paffhausen, Thierry Meurers, Inga Fuchs, Randolf Menzel, and Tim Landgraf. Tbd - a flying experimental platform for the in-flight investigation of honey bee navigation. *TBD*, 2018.

[29] PIGPIO-Team. Pigpio c-libary, 2018.

[30] Johannes Polster. B.sc. thesis: Simulating bee vision: Conceptualization, implementation, evaluation and application of a raycasting rendering engine for generating bee views. *BioRobotics-Lab*, 2017.

[31] Pauline Pounds, Robert Mahony, and Peter Corke. Modeling and control of a large quadrotor robot. 18:691–699, 07 2010.

[32] Jürgen Rybak and R Menzel. Anatomy of the mushroom bodies in the honey bee brain: the neuronal connections of the alpha-lobe. *Journal of Comparative Neurology*, 334(3):444–465, 1993.

[33] Mandyam Srinivasan and Miriam Lehrer. Temporal resolution of colour vision in the honeybee. *Journal of Comparative Physiology A*, 157(5):579–586, 1985.

[34] Karin Steijven, Johannes Spaethe, Ingolf Steffan-Dewenter, and Stephan Härtel. Learning performance and brain structure of artificially-reared honey bees fed with different quantities of food. *PeerJ*, 5:e3858, 2017.

[35] STMicroelectronics. Stm32f405xx and stm32f407xx - data sheet - arm cortex-m4 32b mcu+fpu, 210dmips, up to 1mb flash/192+4kb ram, usb otg hs/fs, ethernet, 17 tims, 3adcs, 15 comm. interfaces & camera. 09 2016.

[36] Thomas Stone, Barbara Webb, Andrea Adden, Nicolai Ben Weddig, Anna Honkanen, Rachel Templin, William Wcislo, Luca Scimeca, Eric Warrant, and Stanley Heinze. An anatomically constrained model for path integration in the bee brain. *Current Biology*, 27(20):3069–3085, 2017.

[37] Ltd. SZ DJI Technology Co. Dji matrice 100 - user manual. 03 2016.

[38] Gavin J Taylor, Tien Luu, David Ball, and Mandyam V Srinivasan. Vision and air flow combine to streamline flying honeybees. *Scientific reports*, 3:2614, 2013.

[39] Stefan Walter. Development and comparison of three different retainer prototypes for stationary flying bumblebees (bombus terrestris) in a virtual reality. 2015.

[40] A. M. Wenner. The flight speed of honeybees: a quantitative approach. *Journal of Apicultural Research*, 2(1):25–32, 1963.

[41] Benjamin Wild. Bestimmung der 6d-eigenbewegung eines fliegenden roboters anhand von monokularen messungen des optischen flusses. *BioRobotics-Lab*, 2014.

[42] Th J Wolf, P Schmid-Hempel, CP Ellington, and RD Stevenson. Physiological correlates of foraging efforts in honey-bees: oxygen consumption and nectar load. *Functional Ecology*, pages 417–424, 1989.

[43] Eric Zetzsche. Kollisionserkennung für single-/multicopter mit hilfe des optischen flusses. *BioRobotics-Lab*, 2014.

A Acknowledgements

Besides the author, there are several people involved in the successful elaboration of a thesis. Hereby I want to thank some of them.

I am grateful to Prof. Dr. Tim Landgraf of the Biorobotics Lab at Freie Universität Berlin for giving me the opportunity to work on this project in his lab. I always received every support I needed but I also had a lot of freedom.

I want to thank Prof. Dr. Dr. h.c. Randolf Menzel for helping with his unique knowledge and experience with honey bees. He also gave me the opportunity to test the Neurocopter in Hesse with his Bee Radar.

Without Dr. Benjamin Paffhausen, the Neurocopter project would have been impossible. He designed all analog electronics and prepared most of the bees for the extracellular recordings. When it got critical, he spent days and nights in the Neurobiology institute with me.

Benjamin Wild helped in Hesse with the experiments, did the navigational capabilities during flight experiments together with Prof. Dr. Tim Landgraf and helped when I had questions about data analysis.

My great friend Thierry Meurers helped me during a software project to develop the initial version of the Neurocopter. He also helped for 3 weeks of his holiday in Hesse doing the experiments.

Inga Fuchs caught and prepared many bees for the Neurocopter and helped me with her experience in electrophysiology.

Morgan Geldenhuys helped with the software development of the Umweltspäher-project from which the code base was taken for the STM32-based Neurocopter OBC. I also want to thank him for his very fruitful comments towards this thesis.

If I wanted to discuss technical computer science problems, Hauke Mönck always had time.

Rowland Johnson developed a tool for converting DJI-DAT files into CSVs. For the Neurocopter, he added support for the DJI Matrice 100.

I want to thank my girlfriend Jana Cavojska for always being there for me and for helping with her knowledge about data analyses.

Last but definitely not least I want to thank my parents Elke and Rudolf Petrasch who always supported my scientific career.

B Appendix

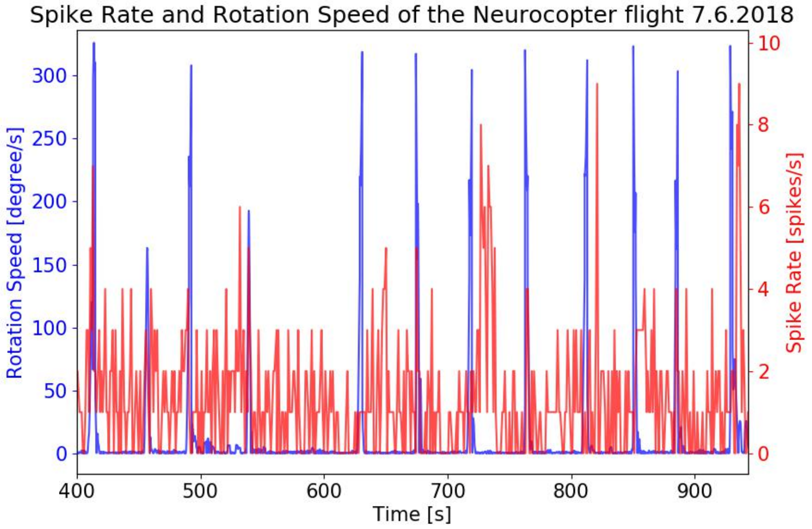


Figure 38: Spike Rate (Unit 2) and Rotation Speed of the Neurocopter flight 7.6.2018

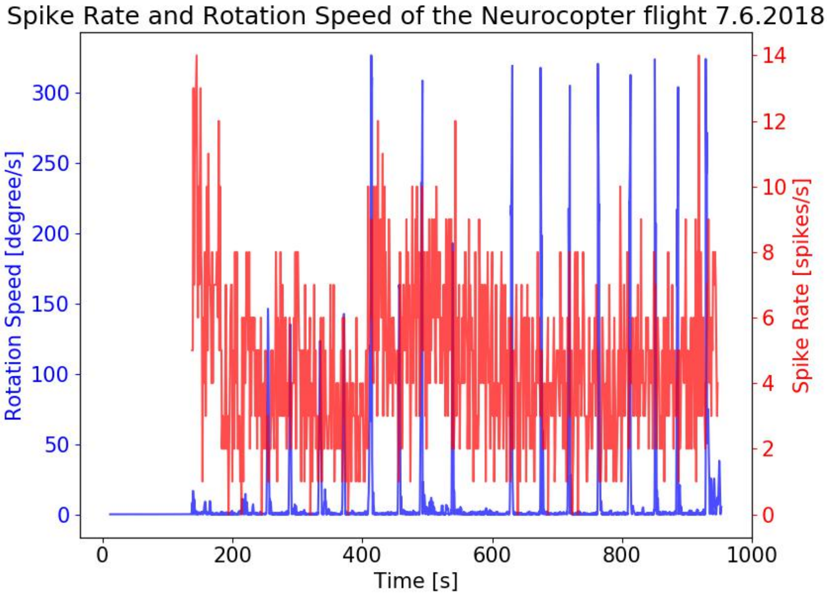


Figure 39: Spike Rate (Unit 3) and Rotation Speed of the Neurocopter flight 7.6.2018

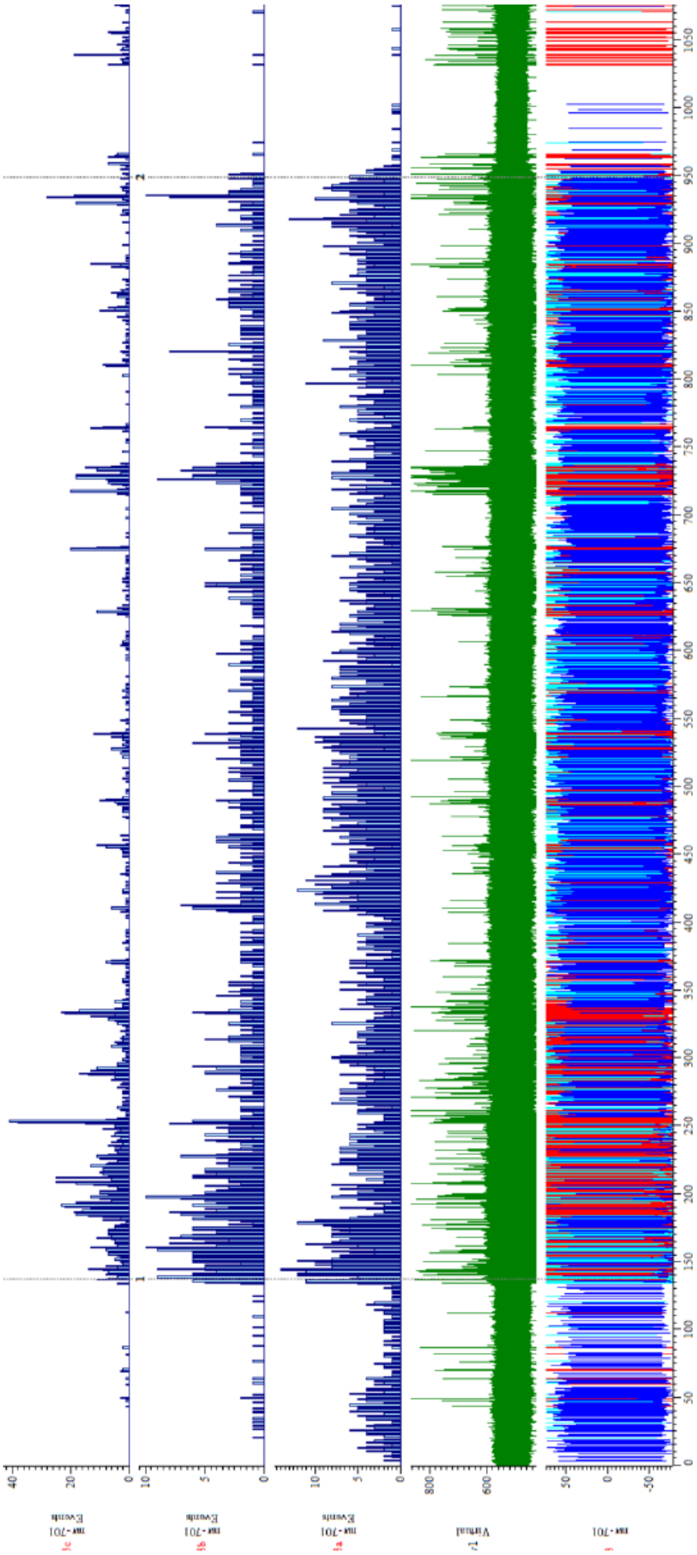


Figure 40: Spike sorting result of the first flight from the 07.06.2018. Three units could be separated from the digital differential channel. The first three plots are the spike rates (spikes per second) of those different units. The fourth plot shows the digital differential channel. The rising of noise when the drone was flying is clearly visible. The last plot shows the detected spikes.

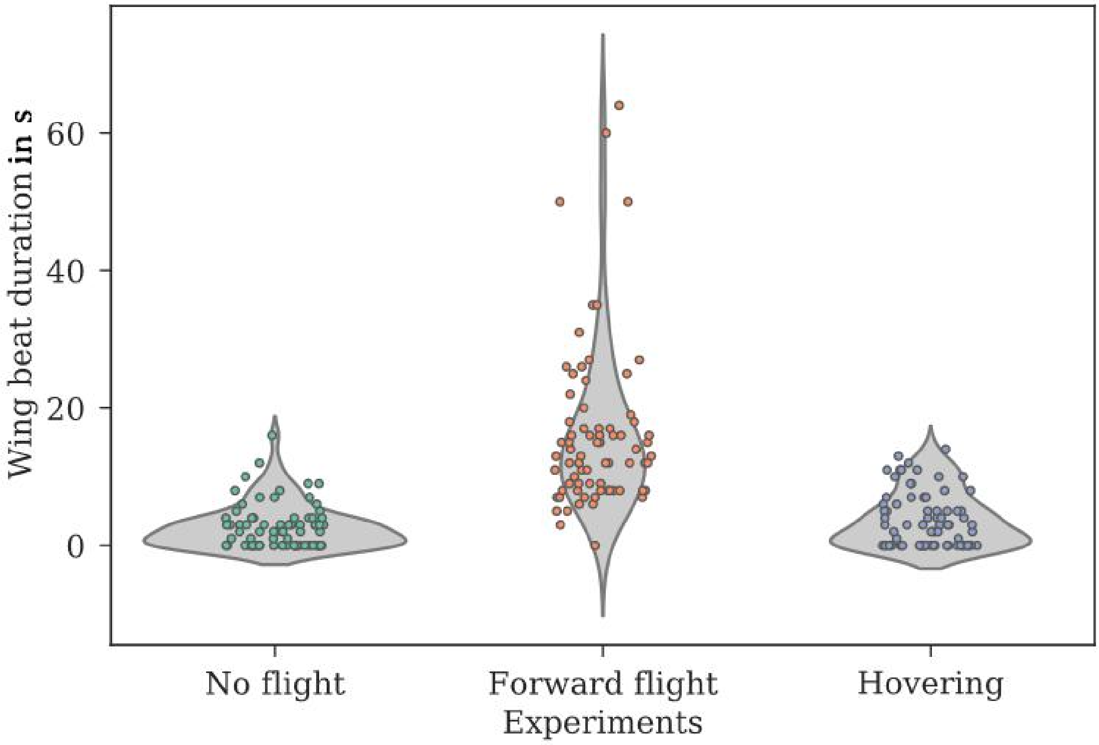


Figure 41: Violin plots of the wing beat duration experiment results. Compared to figure 31 which shows the same data points the difference between *No flight* and *Hovering* becomes a bit clearer.

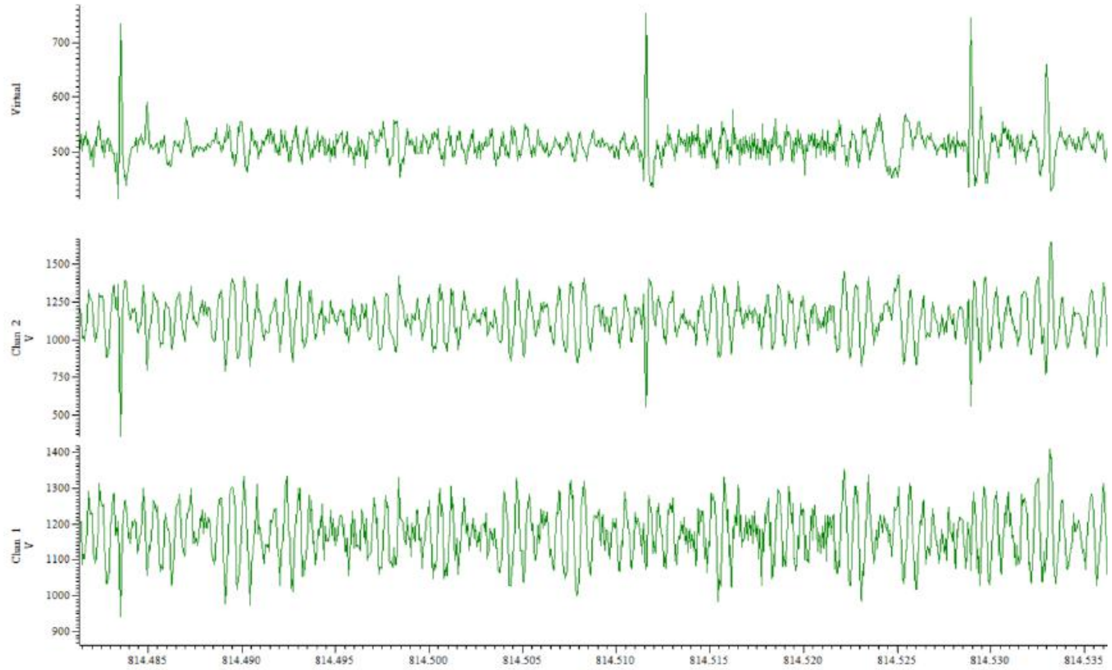


Figure 42: Strong neural signals recorded during a Neurocopter testflight on 07.06.2018. The second and third plot are the two different channels. The spikes are mainly on channel 1 (last plot). Some noise caused by the copter is visible on both channels. The first plot is the difference of the two channels (subtracted with an amplification factor). The subtraction eliminated most of the copter noise which is the only way to detect smaller spikes during a copter flight.

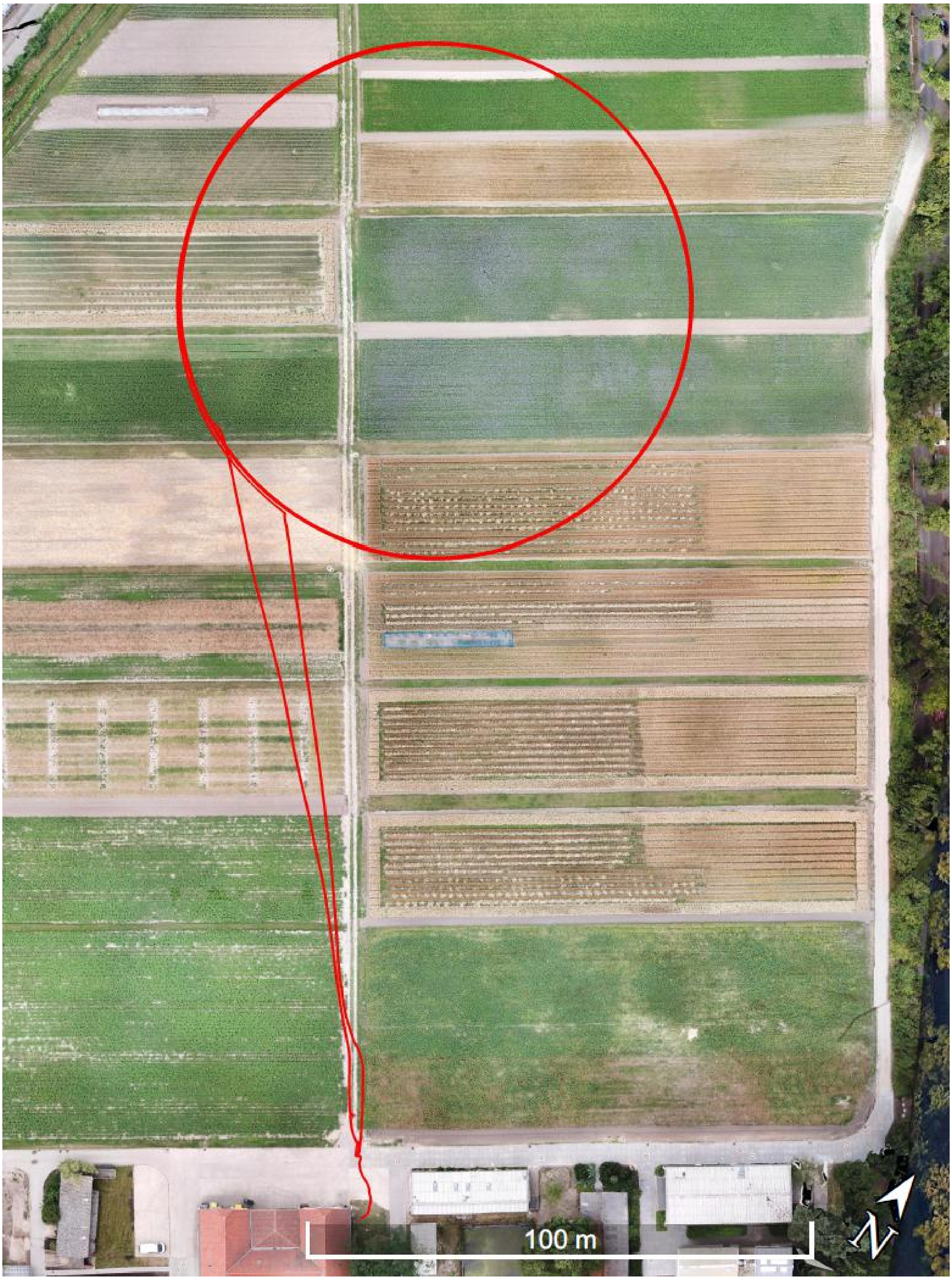
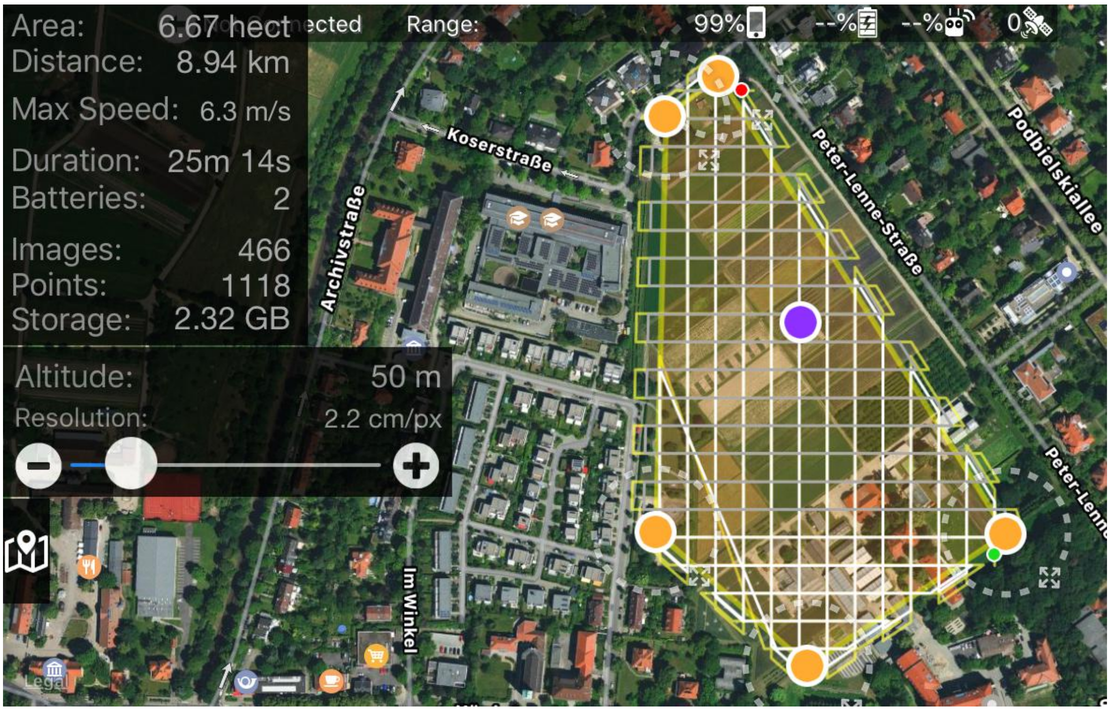


Figure 43: Circle pattern flown on the 05.07.2018 at the Julius Kühn-Institut. The radius of the circle is 50m, the altitude 19 m and the speed was 23 km/h. The way to and from the automatically flown circle is also plotted. For more information about that experiment read section 7.1.1. The background map was created by the author with a drone (for the mapping process see section B.1 in the appendix).

B.1 Aerial mapping for the Neurocopter project

Many Neurocopter experiments study how the neural activity of a honeybee is correlated to landmarks and aerial structures. Therefore, up to date maps of the experiment area are required. Further, the Biorobotics lab wants to develop a flight simulator for real and artificial honeybees. Already, software was developed which could simulate the visual input of a honeybee flying over the Neurocopter experiment areas [30]. The software needs 3d-models of the areas. The process of the aerial imaging and data processing for 2d-maps and 3d-models is explained in this section.

The mapping is done with a gimbal-stabilized camera on a copter. It is possible to attach a DJI-X3 camera to the Neurocopter. To do the automatic mapping, flight and imaging, the IOS-Application *Map Pilot* is used (see figure 44). The mapping area and then the flight altitude has to be selected. A lower altitude enables higher ground resolution but takes more flight time. The flight and imaging plan is calculated automatically and can be uploaded to the drone.



Area: 6.67 hect
Distance: 8.94 km
Max Speed: 6.3 m/s
Duration: 25m 14s
Batteries: 2
Images: 466
Points: 1118
Storage: 2.32 GB
Altitude: 50 m
Resolution: 2.2 cm/px

Figure 44: Planning of the Mapping flights for the map of the Julius-Kühn-Institute fields in the IOS-Application *Map Pilot*.

The images taken by the copter contain GPS-coordinates and the camera orientation. With that information and enough overlap between the images via photogrammetry, the 3d-model and 2d-map can be generated. In the Neurocopter-Project, *Pix4d-Mapper* and *Agisoft PhotoScan* were used for that. Figure 45 shows a 3d-model of the experiment area for extracellular recordings (Julius-Kühn-Institute) and the copter camera positions. The 2d-maps are georeferenced (they contain information on how to map the pixels to geographic coordinates). With that, it is possible to do measurements (like distances or areas) in the maps and to overlay them over other maps like in figure 46.

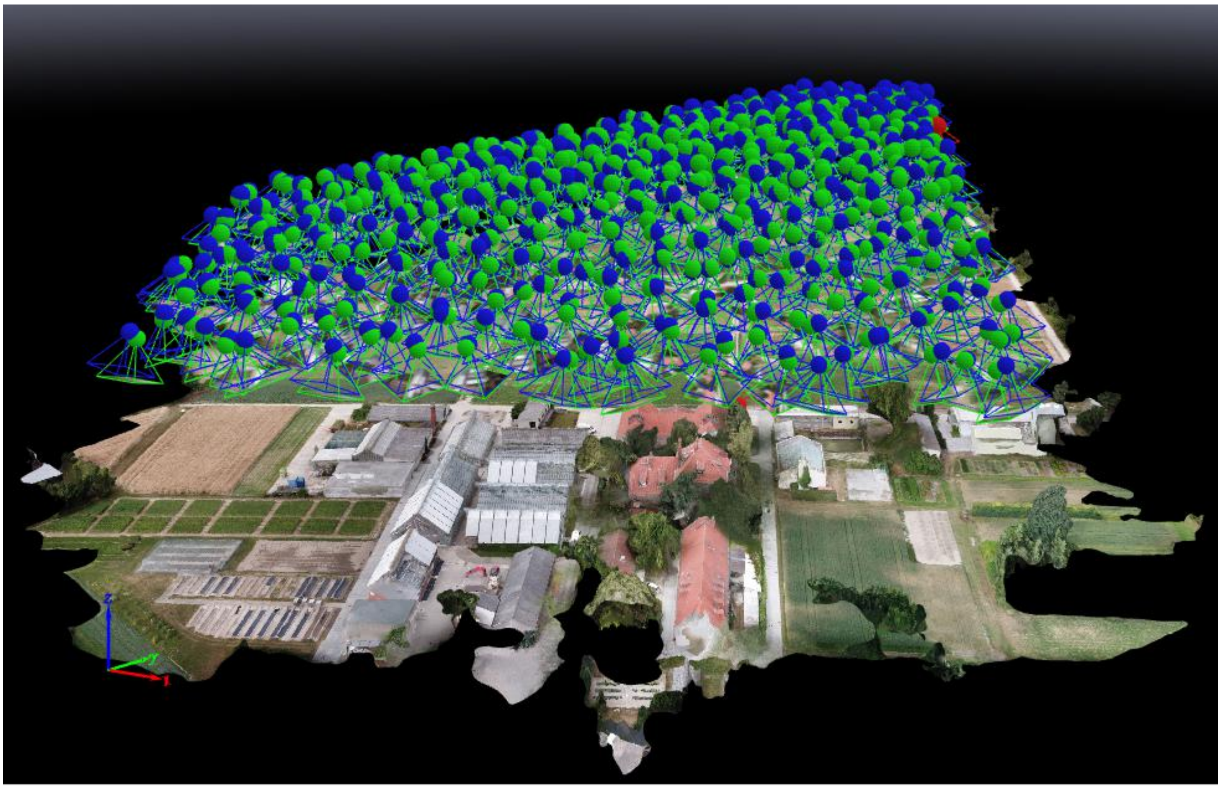


Figure 45: 3d model of the Julius-Kühn-Institute opened in Pix4d-Mapper with copter-camera positions. The copter's altitude was 50m above ground.

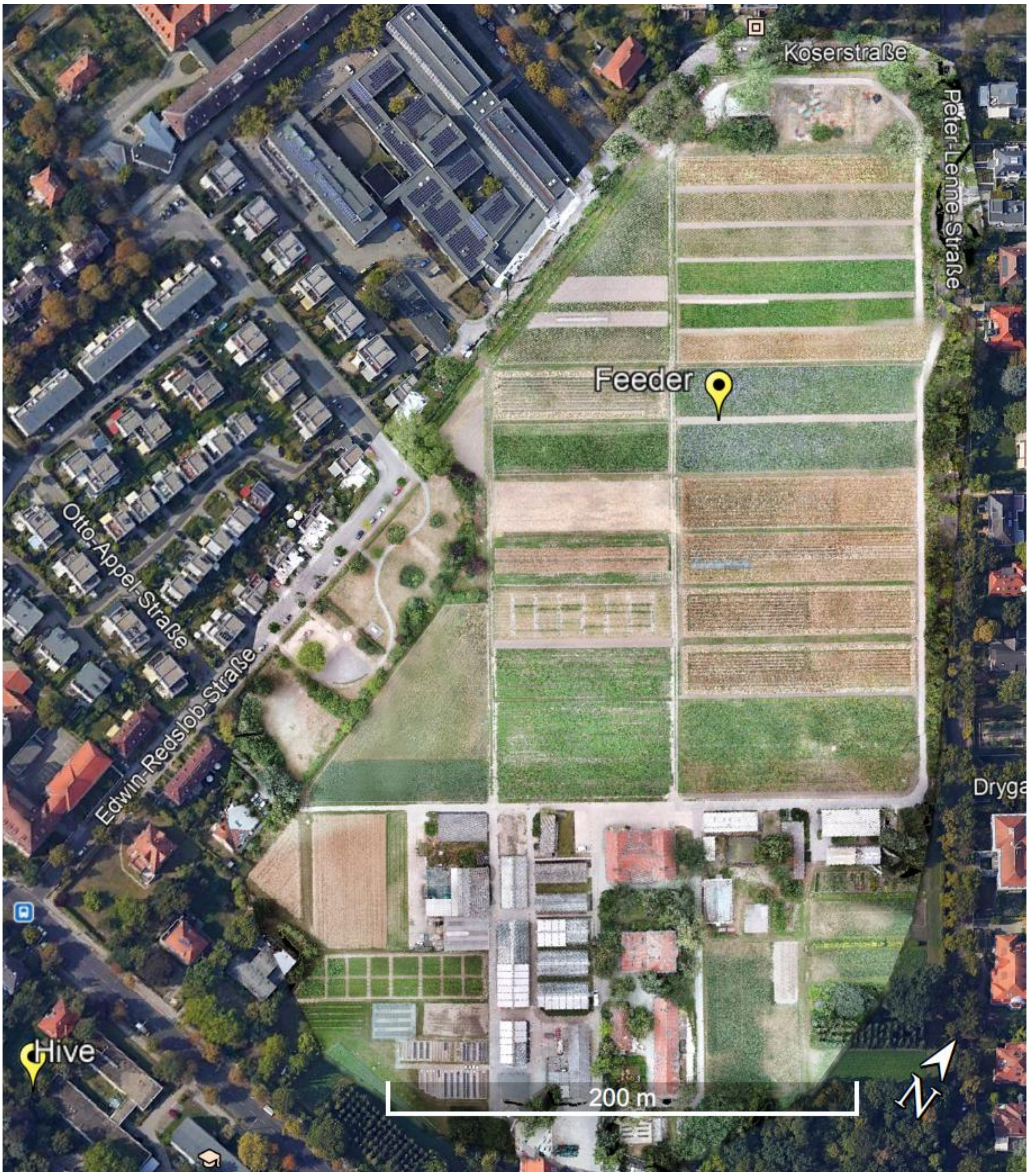


Figure 46: The 2d map of the Julius-Kühn-Institute overlaid over a Google Earth background map. The feeder for the *Overflight over a trained feeder experiment* (see 7.1.3) and the honeybee's hive location are marked.