Portable saccade and head-posture monitor instrument for pigeon and the study on its hemiencephalic dominance behavior

Zixuan Lu

September 30, 2020

Contents

1 Background

1.1 Bioelectrical signals involved in behavioral studies

Pigeon behavioral research may involve the analysis of various bioelectrical signals, including ophthalmic, cerebral, and neuronal discharges. Additionally, two kinematic signals - head posture and eyelid position - can be utilized as good reflections of pigeon behavior.

1.2 Motivation

The conventional experimental method involves restraining pigeons on an experimental bench and monitoring relevant signals in a fixed posture. Although the data quality in terms of accuracy and confidence may not be comparable to mainstream experimental conditions, the natural behavior of unrestrained animals holds significant value. Therefore, a lightweight experimental platform carried directly by pigeons has great research potential. To meet this need, the experimental platform must fulfill certain requirements:

1. ***Lightness:*** *animal can carry.*
2. ***Portability:*** *does not interfere with the natural movement of the animal.*
3. ***Stability:*** *non-removable load subjected to numerous removal behaviors of experimental animals.*
4. ***Data Reliability:*** *stable 'Hold and Clamp' platform.*

This project aims to measure the movement signals of pigeons' head position and eyelid position. To meet these requirements, a three-week experiment was conducted, resulting in the final scheme: head (including gripper platform and sensor) - data transmission line and interface - body (signal processing and collection).

Figure 1: head-interface-body(package)

2 System design

2.1 Function description

Based on the above requirements, the following statements refine the system functions:

1. ***Head Posture:*** *Sample 6 channels of raw six-axis gyroscope data with maximum frequency (x-acc, y-acc, z-acc, x-ang, y-ang, z-ang).*
2. ***Eyelid Position:*** *Pigeon's eyes is with* 0.5×1 *(mm) magnets, using Hall elements and signal amplification circuit to generate signals that vary with the eyelid position and thus can be distinguished by the ADC on board. The left and right eyelid position signals are collected at a sampling rate of 1000Hz. Every 10 groups (bilateral) signals are packaged with gyroscope signals and sent to the serial port.*

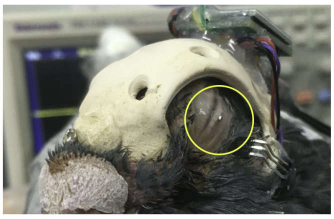

Figure 2: Magnets and Hall elements on the eyelids of pigeons

3. ***Others:*** *Recording data to SD card. The microcontroller packs data into a specific format; the serial recording module records data to the SD card, and the microcontroller uses a timer to precisely control the packing time, with a frequency of 100Hz.*

2.2 System working principle

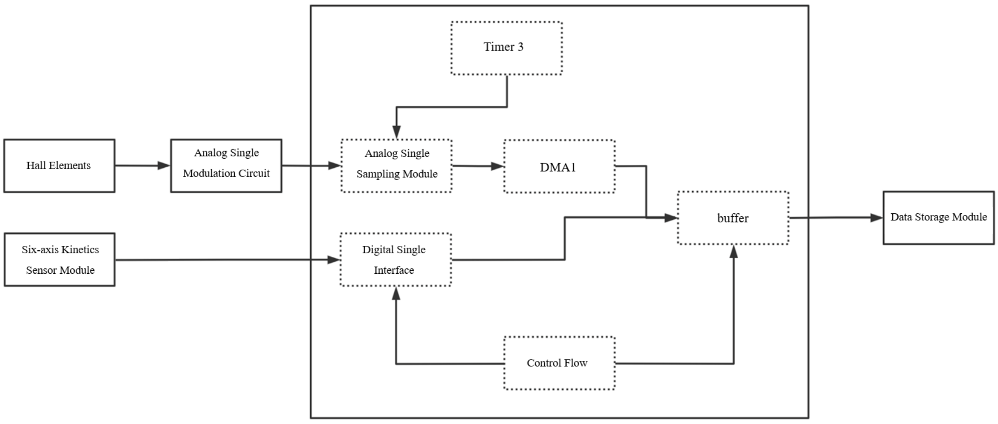

Figure 3: system workflow

2.2.1 Peripheral hardware

Peripheral hardware mainly consists of three parts: sensors, MCU modules, and data lines. The sensors include Hall sensor and MPU6050 gyro position sensor, which are integrated with the MINI-USB base and built on the pigeon's helmet to form a rigid fixation. External MCU modules are directly connected to the MCU, including data storage module, analog signal conditioning circuit, and power supply (with booster board). Data cable is also connected to the single chip microcomputer, using three core, four core enamelled wire and MINI-USB male connector. The joint can be matched with the interface on the pigeon's head.

Analog signal conditioning circuit provides the ability to add bias to Hall element signals and amplify (30x gain) small signals.

2.2.2 MCU

The microcontroller mainly realizes the following functions: MCU collects data of MPU6050 gyroscope through IIC bus; The timer triggers ADC to collect analog signals at 10Hz, and transmits data to DMA cache. The cache size is 20 ADC data (10 for each of the left and right eyes). After DMA triggers the interruption of transmission completion, MCU packages the data cached by the gyroscope and DMA in a specific format and records it to the SD card through the serial port with a packet sending rate of 100Hz.

3 Results and data pf preliminary tests

3.1 Experimental condition

Two sets of experiments were conducted:

1. *The first pigeon: stood alone in a dark box, carried the experimental instrument to collect data until the battery ran out. Recorded valid data duration: 2.19h.*
2. *The second pigeon: placed in a dark box together with the other two pigeons, carrying the experimental instrument to collect data until the battery ran out. Recording valid data duration: 2.28h.*

3.2 Data format

The microcontroller processes the signal into the following formats: the packet with packet-header and gyroscope, ADC data stored in binary format; Later on, MATLAB is utilized to process the binary data. Please refer to the attachment for further details regarding the algorithm.

Packet format

Header: $$$$

Packet number

ADC: 2 channels * 10 ps

MPU: x, y, z_acc

x, y, z_ang

Figure 4: Packet format

The data after visualization is as follows:

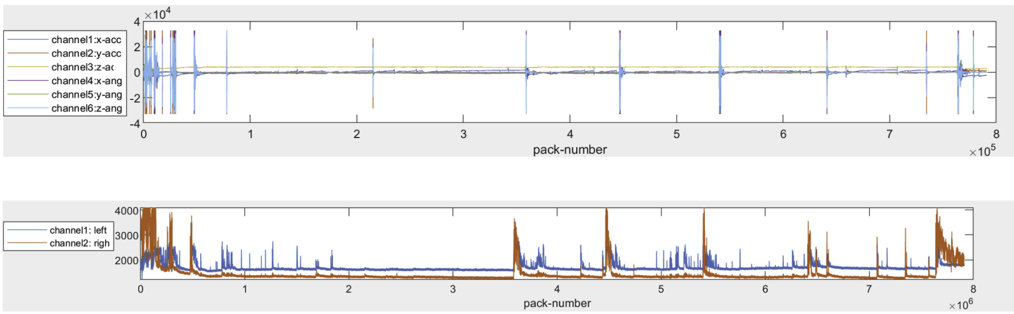

Figure 5: Data after visualization

The upper line displays data from the MPU6050 gyroscope, while the next line shows Hall sensor signals collected by the ADC. The number line indicates the number of data sets. It is important to note that each packet corresponds to one MPU reading, whereas there are ten times as many ADC readings per packet. As MCU sends packets at a strict rate of 100Hz, the horizontal axis on this image can be used to reflect time intervals. After calculations were made, it was determined that each grid (the large horizontal axis grid in the figure) represents a time span of 10 minutes.

By using the self-designed magnetic sensor, the magnetic pole direction can be determined when the magnet is applied to the pigeon's eyelid, so that the peak of ADC signal (the bottom line in the figure) represents the pigeon's eyelid is open, and the low peak represents the eyelid is closed.

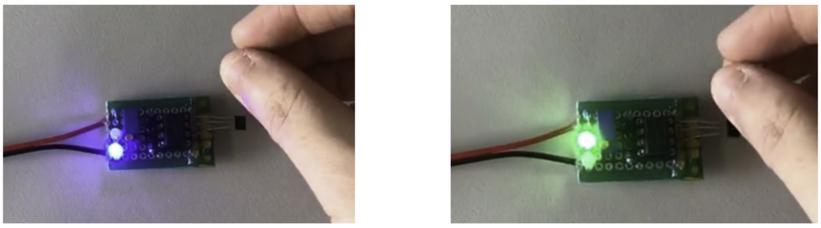

Figure 6: A self-designed magnetic sensor

4 Behavioral research

4.1 Data processing method

To enhance the signal characteristics, the following procedures are implemented: a global search for maximum and minimum values, linear stretching of the signal, alignment of baselines and amplitudes between two signals. The triaxial acceleration signal from the gyroscope is temporarily shielded to solely focus on angular velocity signals. Sparse selection of ADC signals ensures that eye movement signals maintain consistent time-domain resolution (100Hz) with gyroscope signals.

4.2 Problems and its analysis

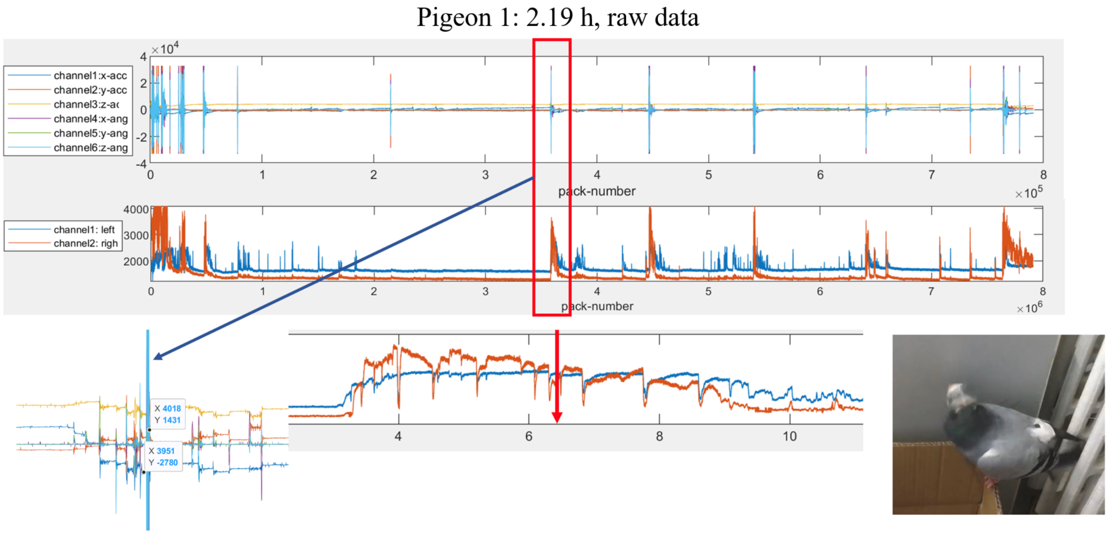

Figure 7: Analysis of the first pigeon after treatment

After conducting the aforementioned data processing, a strong correlation was observed between the active data peaks of MPU and eye movement. Given the high peak area, initial suspicion arose regarding potential instrument malfunction. After increasing the signal time resolution, the peak width was reduced to approximately 10 seconds. Combined with the burr signal, it exhibited instantaneous behavior within 1 second, indicating an effective signal and corresponding pigeon behavior. This observation is consistent with pigeons wearing the instrument exhibiting head shaking behavior during initial adaptation.

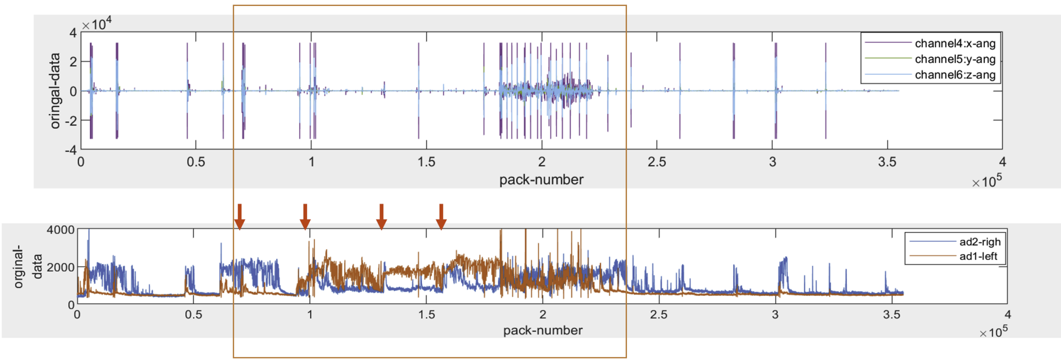

Figure 8: Analysis of the second pigeon after treatment, part 1

Phenomenon Further examination of the data from the second pigeon revealed a switch in hemispheric preferences at the four red arrows shown in the image above. That shows right-eye-left-brain dominance to left-eye-right-brain dominance, then further to binocular dominance.

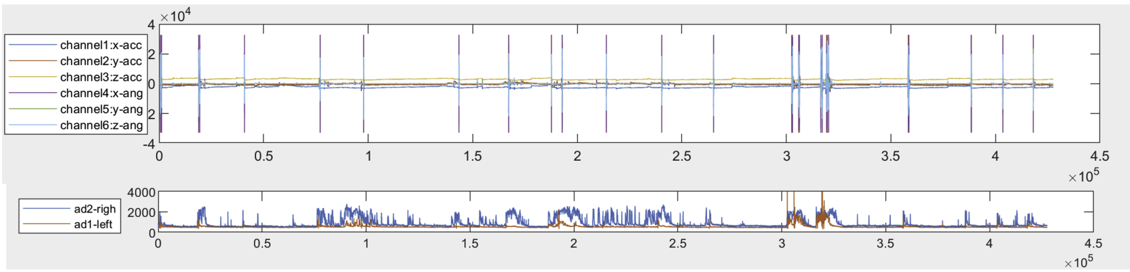

Figure 9: Analysis of the second pigeon after treatment, part 2

When the data were monitored over a larger time scale, more pronounced hemispheric preferences were observed, lasting up to 20 minutes.

4.3 Conclusion

Summary of the above analysis: vigorous head and body movements are always accompanied by frequent eyelid movements. There is hemispheric preference for eye movement. Further consider the following questions:

1. *In the case of sleep, can we observe the alternation of the left and right eyes, or the sleep of one half of the brain?*

Improvement and feasibility At present, the project has the following improvement schemes, which have considerable application prospects:

1. *Miniaturization and lightness: Further integration of peripheral hardware and MCU.*
2. *Add EEG function: The circuit needs to be redesigned and adjusted, and more objective, accurate and scientific biophysical signal indicators should be added to distinguish the sleep/wake state of pigeons*
3. *Group behavior experiment: 8 to 10 pigeons were gathered with helmets, which could be used to study the group behavior of pigeons, such as the relationship between the arousal state of the half brain of pigeons at the edge of the group and their group position*

5 Project summary and harvest

Before the project, I interned in the subject group and participated in a topic related to human behavior. Later, upon discovering a related need within the instrumentation group and my personal interest in instrumentation development, I discussed with Prof. Yang and decided on this topic. The project was established in September 2019, during which time I studied linear electronic circuits, analog circuits as well as knowledge pertaining to C language and MATLAB on my own due to their relevance to Prof. Yang's lab needs. I received enthusiastic guidance from my lab group members throughout the project completion process, and I found the entire execution period to be highly rewarding.

Due to the epidemic, the original plan of completing the project at the institute within four months during the second semester of junior year failed to materialize. Consequently, apart from acquiring fundamental knowledge, only two and a half months were spent in the institute (June 2019 summer vacation and July-August 2020 summer vacation), which made it impossible to accomplish all project ideas and behavioral experiments as well as subsequent data processing software development.

It is worth acknowledging that the project was approved by the project leader, Prof. Yan Yang. The experiments gained valid data, which complemented and corroborated the ongoing experiments of the subject in the group. It was also the first time that the knowledge learned in school was combined with the needs of the laboratory, and I felt significant to be able to create tools for front-line scientific research.

I stayed at the institute until three days before the start of the school year in September. Then it was time to back to the long-lost campus to finish the last semester of study.

Compared with the previous research experience, this project exercised practical skills and extraordinarily examined the application of book knowledge extrapolation, which will become a unique memory of my university days.

6 Contribution

Wu Chen provides part of the MCU codes and the diagram of analog conditioning circuit.

Acknowledge

The author thanks for the help from Yan's Lab (State Key Laboratory of Brain & Cognitive Sciences, IBP) to support this research, as well as many scientific issues discussion. The author also thanks to Xi Xu and Tong Xiao for their guidance during the experiment and Chen Wu for his technical support.

The Undergraduate Entrepreneurship and Innovation Program of the Institute of Biophysics, CAS provided funding support for this project.

7 Appendix: Data Post-processing

Binary data processing:

```
clear;
bin_file=fopen(data_path,'rb');%raw data path
bin2max=fread(bin_file);
mark = 0;
err=0;
%---------------------------------------Data filters
for runstep=5:70
if bin2max(runstep)==36
if bin2max(runstep+1)==36
err = runstep - 1;
break;
end
end
end
fclose(bin_file);
[data_col,data_row]=size(bin2max);
global dstruct;
dstruct={(data_col-err)/60};
%---------------------------------------Binary data processing
for point_num=1:(data_col-err)/60
tmp=bin2max(1+err+60*(point_num-1)+8:err+60*point_num,1)';
s=struct;
s.mpu=[1:6];
s.adc=[1:20];
s.mpu_mnp=[1:6];
for inner_num =1:20
s.adc(inner_num)= tmp(inner_num*2-1);
s.adc(inner_num)= tmp(inner_num*2)*(2^8)+ s.adc(inner_num);
inner_num =inner_num+1;
end
for inner_num=1:6
s.mpu(inner_num)= tmp(40+inner_num*2-1);
s.mpu(inner_num)= tmp(40+inner_num*2)*(2^8)+ s.mpu(inner_num);
if s.mpu(inner_num)≥32768
temp=bitcmp(s.mpu(inner_num)-1,'uint16');
s.mpu(inner_num)=-1*temp;
end
s.mpu_mnp(inner_num) = s.mpu(inner_num);
inner_num =inner_num+1;
end
dstruct{point_num}=s;
end
%---------------------------------------constants
colcell={};
colcell{1}=[0 0.4470 0.7410];
colcell{2}=[0.8500 0.3250 0.0980];
colcell{3}=[0.9290 0.6940 0.1250];
colcell{4}=[0.4940 0.1840 0.5560];
colcell{5}=[0.4660 0.6740 0.1880];
colcell{6}=[0.3010 0.7450 0.9330];
colcell{7}=[0 0.4470 0.7410];
colcell{8}=[0.8500 0.3250 0.0980];

mpu_flag{1}='x-acc';
mpu_flag{2}='y-acc';
mpu_flag{3}='z-acc';
mpu_flag{4}='x-ang';
mpu_flag{5}='y-ang';
mpu_flag{6}='z-ang';
adc_flag{1}=' left';
adc_flag{2}=' righ';

mpumnp_flag{1}='dx';
mpumnp_flag{2}='dy';
```

```
mpumnp_flag{3}='dz';
mpumnp_flag{4}='dx_ang';
mpumnp_flag{5}='dy_ang';
mpumnp_flag{6}='dz_ang';

adc_flag{1}=' left';
adc_flag{2}=' righ';

%------------------------------------------function call
plot_data([1,1,1,1,1,1,1,1],2,800000,colcell,mpu_flag,adc_flag,mpumnp_flag);
```

Painter

```
function: plot_data(channel_mask,datapack_start,datapack_end,colcell,
mpu_flag,adc_flag,mpumnp_flag)
global dstruct;
MPU_mnp(datapack_start,datapack_end,1/100);
figure(1)
X_mpu=[1:1:(-datapack_start+datapack_end+1)];
disp(size(X_mpu));
X_adc=[1:1:(-datapack_start+datapack_end+1)*10];
leg={};
leg_={};
leg_{1}=[];
onenum=1;

for channel_num=1:6
if channel_mask(channel_num)==1
Y=[1:1:(-datapack_start+datapack_end+1)];
for data_index=1:(-datapack_start+datapack_end+1)
Y(data_index)=dstruct{data_index+datapack_start}.mpu(channel_num);
end
%scatter(X_mpu,Y,2,colcell{channel_num},'filled');
plot(X_mpu,Y,'Color',colcell{channel_num});
leg{channel_num}=['channel' num2str(channel_num) ':' mpu_flag{channel_num}];
leg_{onenum+1}=strcat(leg_{onenum},['leg{' num2str(channel_num) '},']);
onenum=onenum+1;
hold on
end
end
onenum=onenum+1;
leg_{onenum}=leg_{onenum-1}(1:1:end-1);
an=eval('leg_{onenum}');
eval(['legend(' an ')']);
hold off;

figure(2)
onenum=1;
for channel_num=4:6
if channel_mask(channel_num)==1
Y=[1:1:(-datapack_start+datapack_end+1)];
for data_index=1:(-datapack_start+datapack_end+1)
Y(data_index)=dstruct{data_index+datapack_start}.mpu_mnp(channel_num);
end
%scatter(X_mpu,Y,2,colcell{channel_num},'filled');
plot(X_mpu,Y,'Color',colcell{channel_num});
leg{channel_num}=['channel' num2str(channel_num) ':' mpumnp_flag{channel_num}];
leg_{onenum+1}=strcat(leg_{onenum},['leg{' num2str(channel_num) '},']);
onenum=onenum+1;
hold on
end
end
onenum=onenum+1;
leg_{onenum}=leg_{onenum-1}(1:1:end-1);
an=eval('leg_{onenum}');
eval(['legend(' an ')']);
hold off;

figure(3)
for channel_num=1:2
if channel_mask(channel_num+6)==1
Y=[1:1:(-datapack_start+datapack_end+1)*10];
for data_index=1:(-datapack_start+datapack_end+1)
Y((data_index-1)*10+1:data_index*10)= ...
    dstruct{data_index+datapack_start}.adc(channel_num:2:20);
end
%scatter(X_adc,Y,2,colcell{channel_num+6},'filled');
plot(X_adc,Y,'Color',colcell{channel_num+6});
leg{channel_num+6}=['channel' num2str(channel_num) ':' adc_flag{channel_num}];
```

```
legend(leg{channel_num+6-1},leg{channel_num+6});
hold on
end

end
legend(leg{7},leg{8});
hold off;
```