

Active Aerodynamic Tail Enhances Agile Locomotion of Legged Robots

Authors: Brian Goodluck Mmari, Prof.
Upinder Kaur, Dr. Jiajun An

Purdue University, School of Agriculture
and Biological Engineering



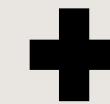
Agility in animals enables complex adaptation mechanisms



Agility

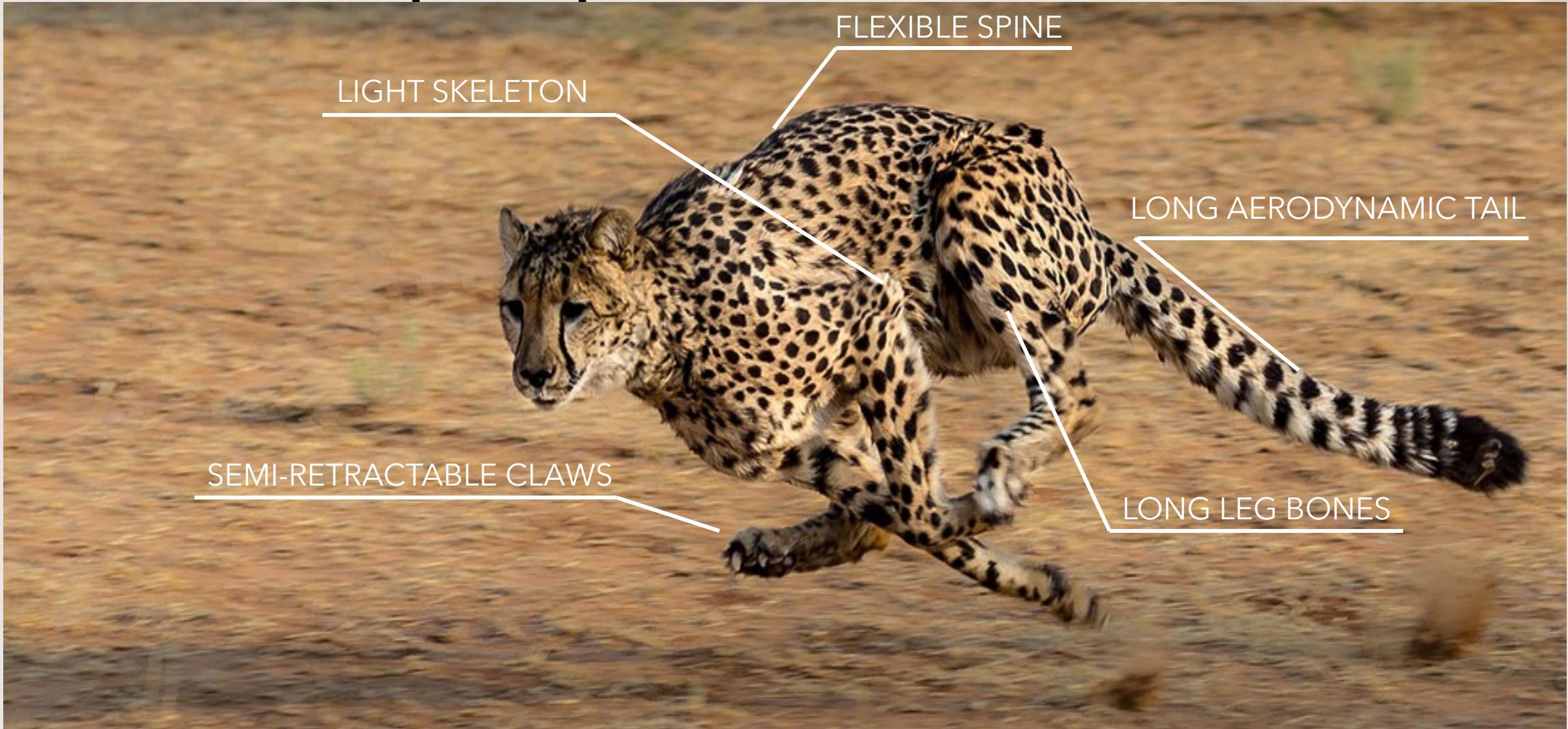


Dynamic Mobility

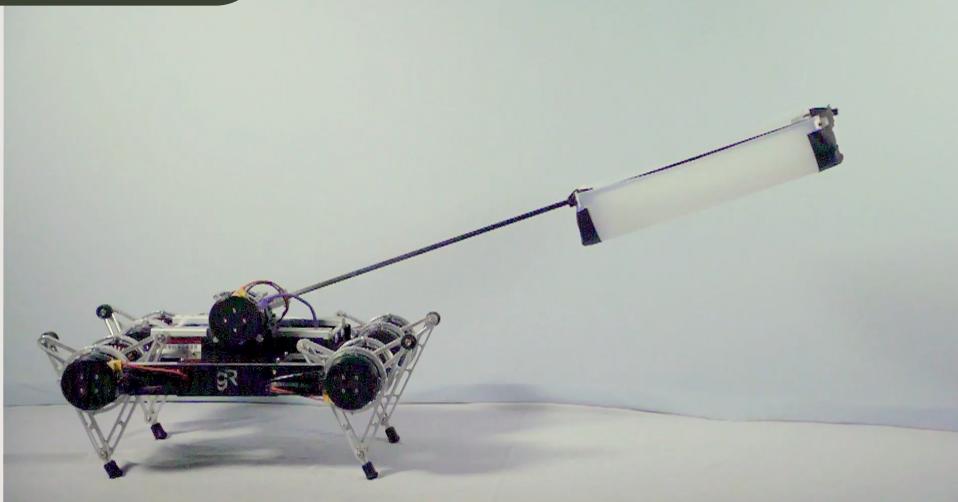
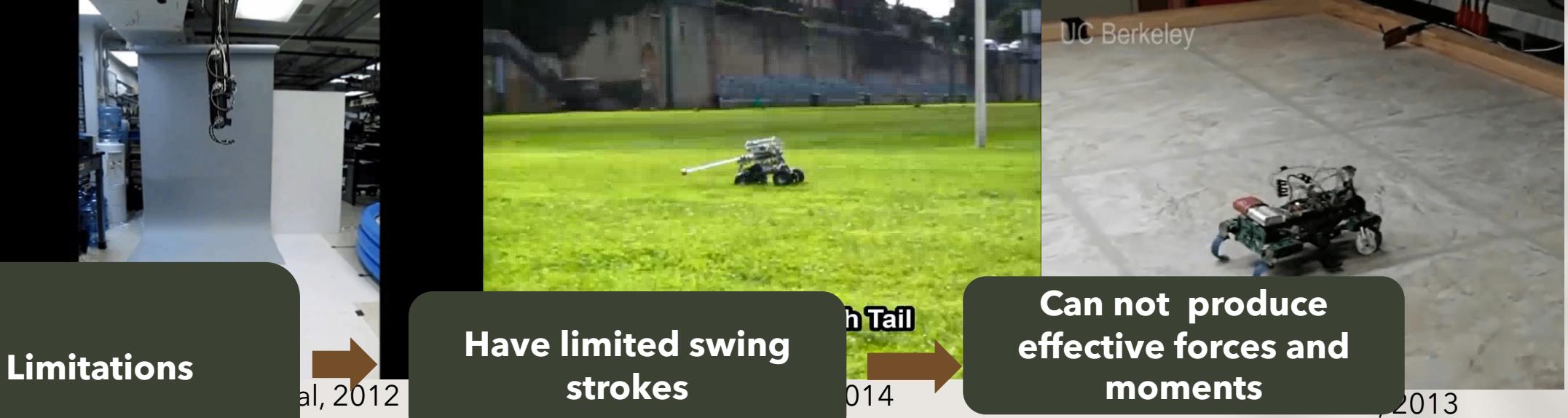


Speed

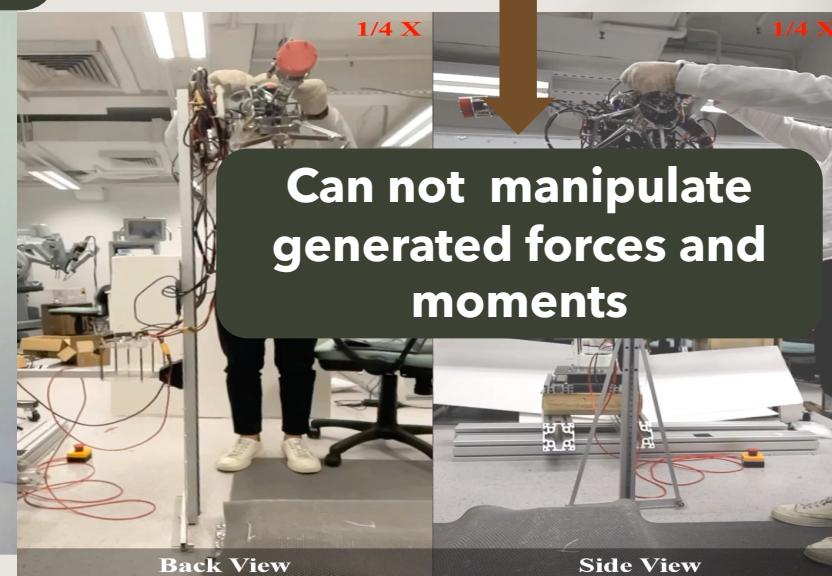
Unique Adaptation features in cheetahs



Investigate existing robotic tail designs



Joseph Norby et al, 2021

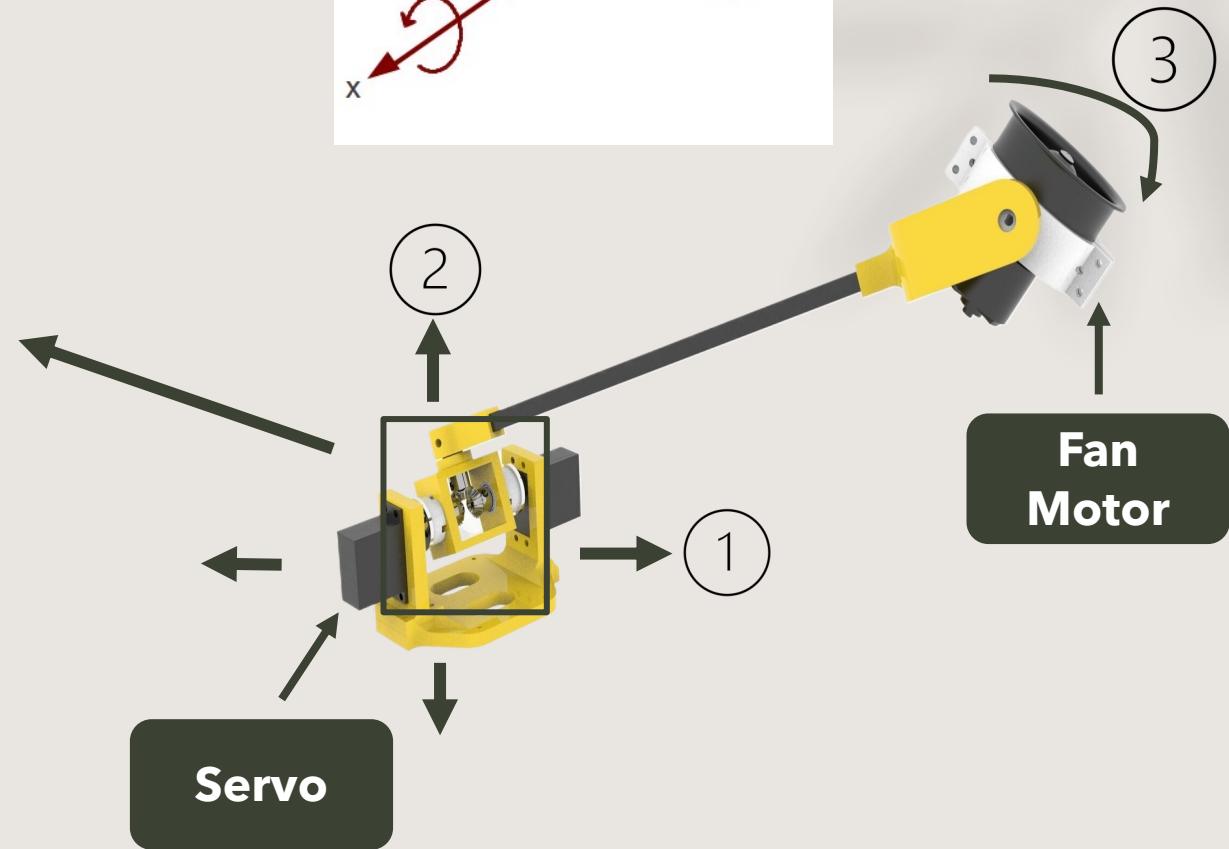
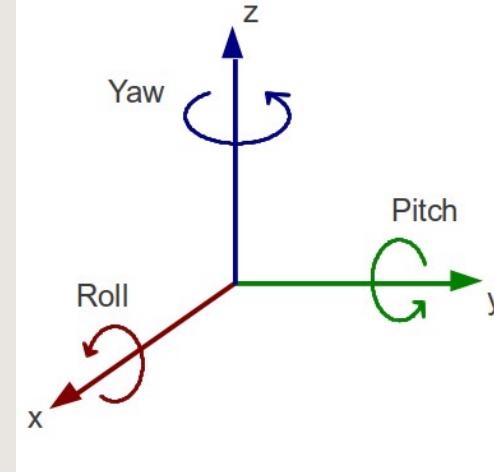
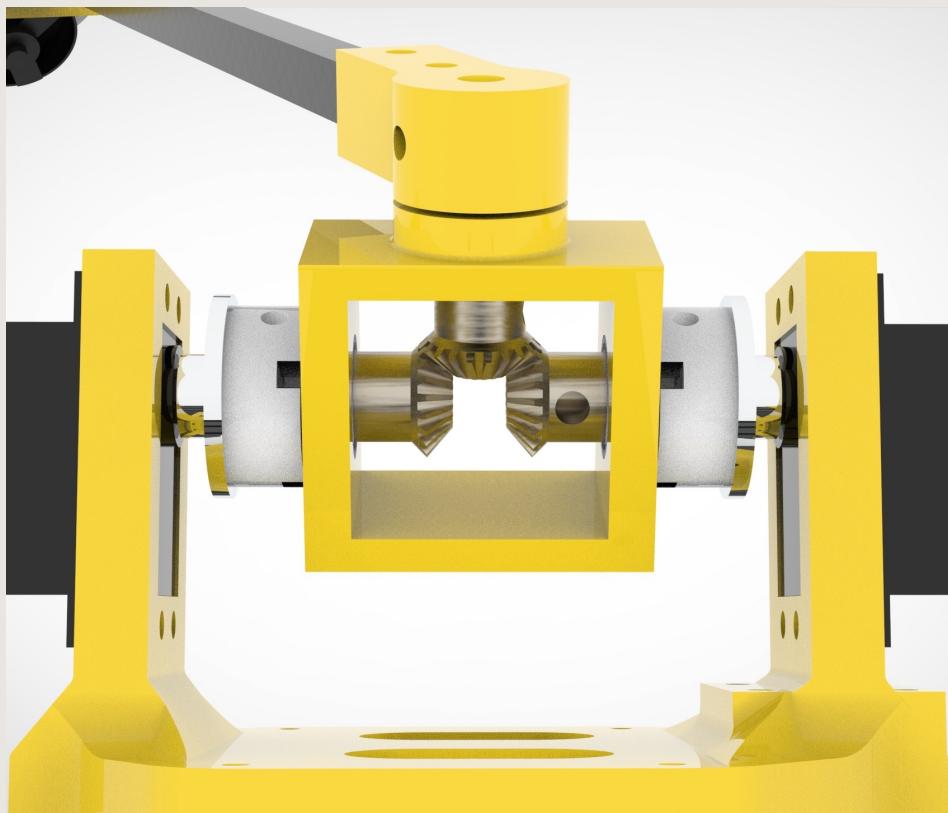


Jiajun An et al, 2023

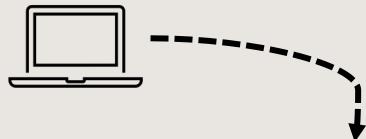
Goal: Design an active aerodynamic that generates continuous forces and moments, **without any stroke or direction limitations**, to enhance the agility of legged robots in complex movements.

Tail Design

3 Degrees of Freedom



Tail Control



ESP_NOW protocol

```
DOIT ESP32 DEVKIT V1
```

MiniPupper_Trans.ino

```
1 #include <esp_now.h>
2 #include <WiFi.h>
3
4 // MAC address of the receiver ESP32
5 uint8_t broadcastAddress[] = {0xE4, 0x65, 0xB8, 0x58, 0x20, 0xD8};
6 // Structure example to send data
7 // Must match the receiver structure
8
9 typedef struct ControlData {
10     int servoid;
11     int steps;
12     int degrees; // change position to degrees
13     int speed;
14     int motorSpeed;
15 } ControlData;
16
17 ControlData controlData;
18
19 esp_now_peer_info_t peerInfo;
20
21 // callback when data is sent
22 void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
23     Serial.print("\r\nLast Packet Send Status:");
24     Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fa
25 }
26
27 void setup() {
```

Output Serial Monitor >

Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on '... Both NL & CR 115200 baud

```
Ln 96, Col 44 DOIT ESP32 DEVKIT V1 on /dev/cu.usbserial-0001
```



```
DOIT ESP32 DEVKIT V1
```

MiniPupper_Rec.ino

```
1 #include <esp_now.h>
2 #include <WiFi.h>
3 #include <HardwareSerial.h>
4 #include <INST.h>
5 #include <SCS.h>
6 #include <SCSCL.h>
7 #include <SCSerial.h>
8 #include <SCServo.h>
9 #include <SMS_STS.h>
10
11 // Define the SCSCL objects for both servos
12 SCSCL sc1; // For the left-servo connected to GPIO10
13 SCSCL sc2; // For the right-servo connected to GPIO17
14
15 // Initialize HardwareSerial objects for both servos
16 HardwareSerial MySerial1(1); // Use UART1 for GPIO10
17 HardwareSerial MySerial2(2); // Use UART2 for GPIO17
18
19 // Define the signal pin for the ESC
20 const int escPin = 2; // GPIO2
21
22 typedef struct ControlData {
23     int servoid;
24     int degrees;
25     int steps;
26     int speed;
27     int motorSpeed;
28 } ControlData;
29
30 ControlData controlData;
31
32 esp_now_peer_info_t peerInfo;
33
34 // callback when data is received
35 void OnDataRecv(const uint8_t *mac_addr, uint8_t *data, int len) {
36     if (len == 12) {
37         controlData.degrees = data[0];
38         controlData.steps = data[1];
39         controlData.servoid = data[2];
40         controlData.speed = data[3];
41         controlData.motorSpeed = data[4];
42     }
43 }
44
45 void setup() {
```

Output Serial Monitor >

Message (Enter to send message to 'DOIT ESP32 DEVKIT V1' on '... Both NL & CR 115200 baud

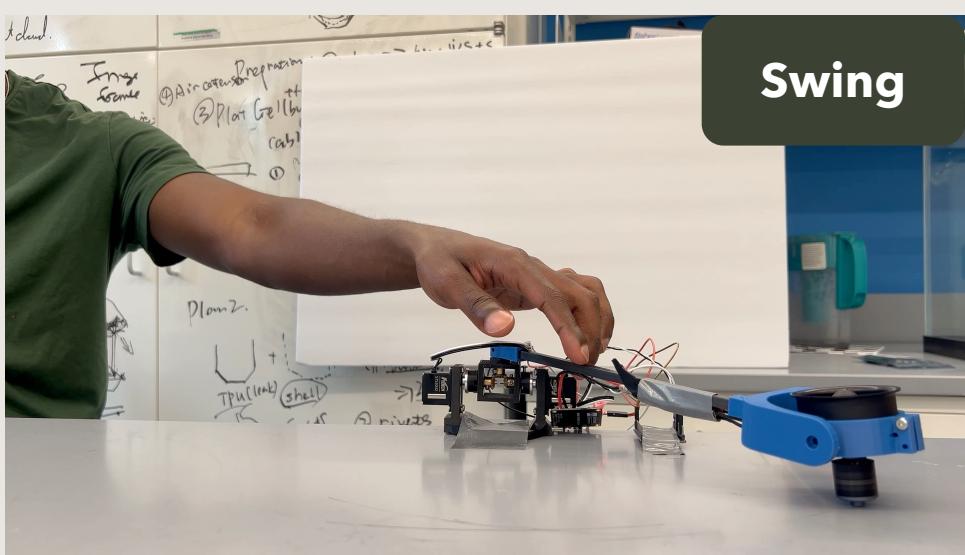
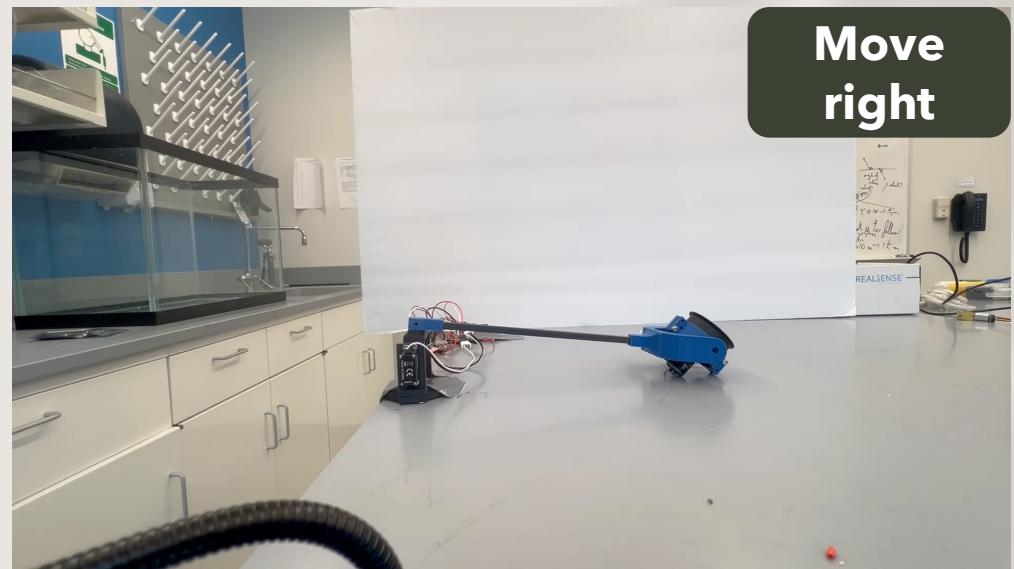
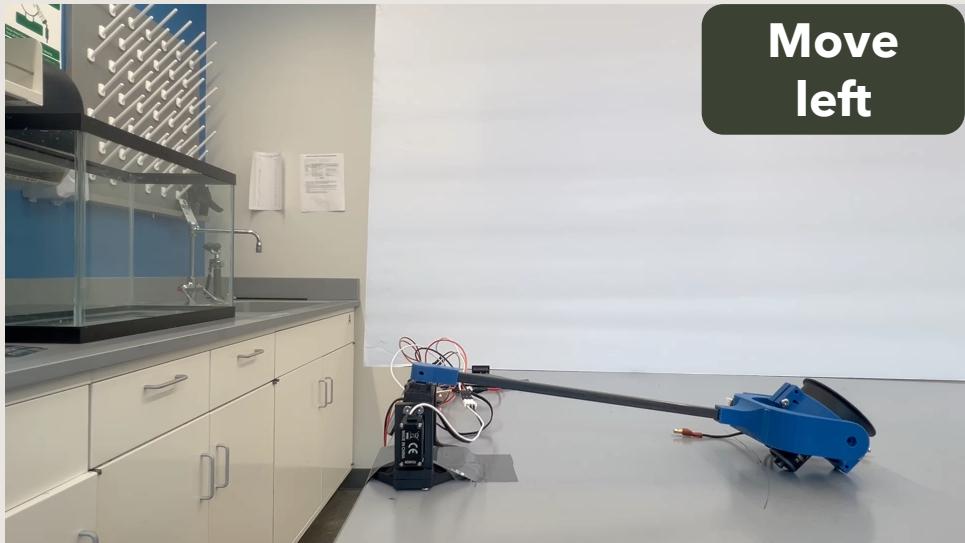
```
15:35:08.903 -> Last Packet Send Status: Delivery Success
15:35:16.723 -> Sent successfully
15:35:16.723 -> Sent Data - Servo ID: 1, Degrees: 14, Steps: 159, Speed: 0, Motor S
15:35:16.723 ->
15:35:16.760 -> Last Packet Send Status: Delivery Success
15:35:21.669 -> Sent successfully
15:35:21.669 -> Sent Data - Servo ID: 1, Degrees: 15, Steps: 170, Speed: 1, Motor S
15:35:21.669 ->
15:35:21.669 -> Last Packet Send Status: Delivery Success
15:35:24.802 -> Sent successfully
15:35:24.802 -> Sent Data - Servo ID: 1, Degrees: 14, Steps: 159, Speed: 1, Motor S
15:35:24.802 ->
15:35:24.802 -> Last Packet Send Status: Delivery Success
```

Ln 55, Col 3 DOIT ESP32 DEVKIT V1 on /dev/cu.usbserial-0001

One-way communication



Tail Experiments

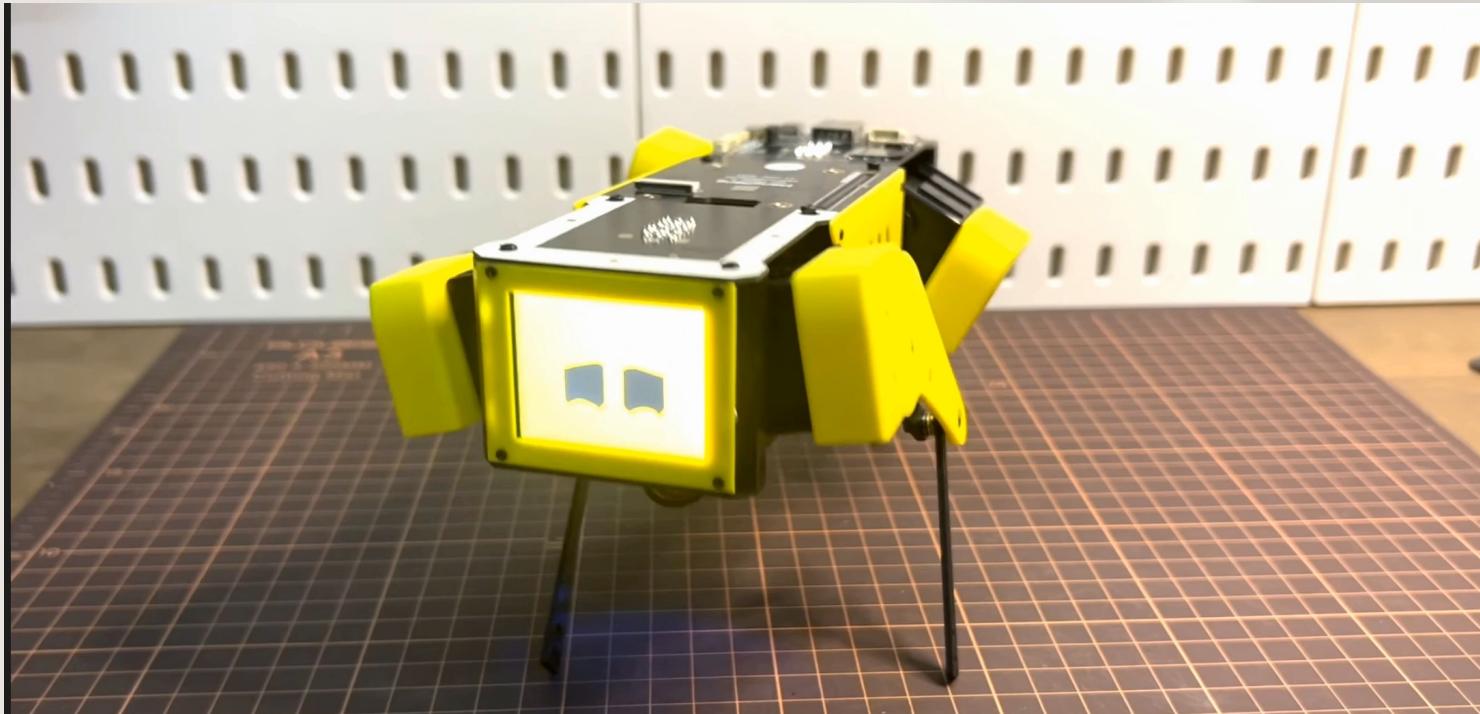


Robot platform

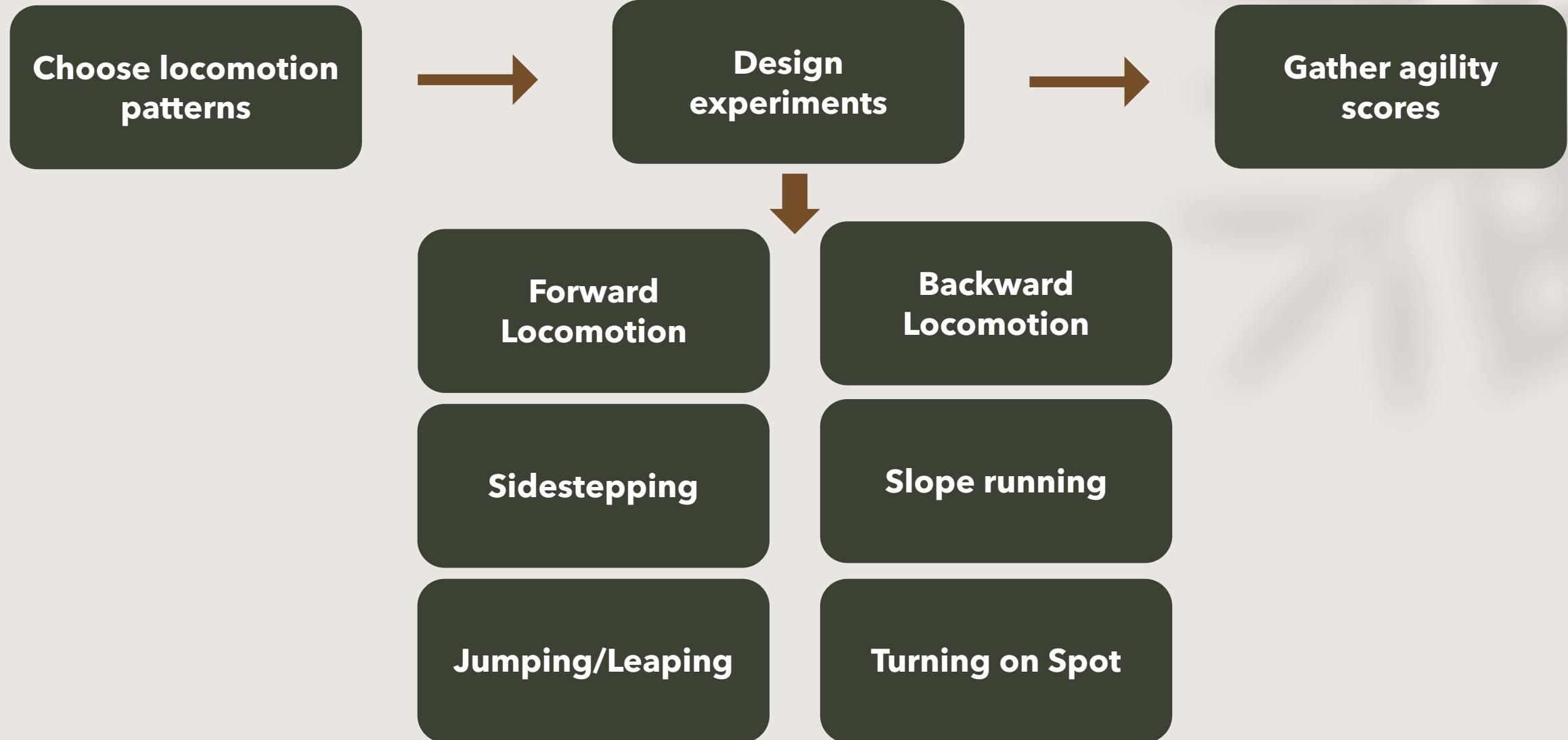
- Mini Pupper 2, quadruped robot
- Reasons for choice:
 - 12 Degrees of Freedom
 - Open source
 - Ability to mount attachments
 - High quality servos
 - Small



18cm x 11cm x 11cm



How do we measure agility?



Experiments



Results

Equations from
“Benchmark agility paper”



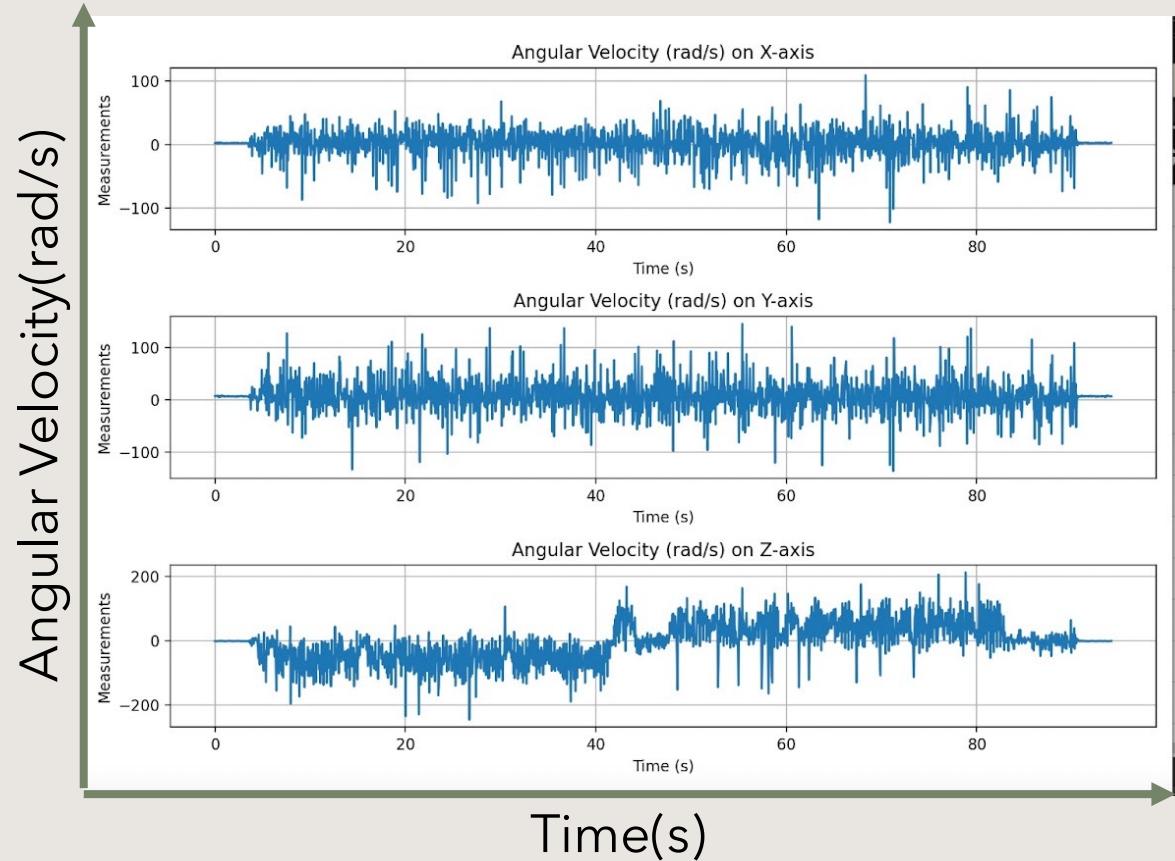
Calculate agility scores for Mini
Pupper 2

Calculation	Variance
$A_{ts} = \frac{p}{t} \cdot \sqrt{\frac{h_R}{g}}$	not needed
$A_{tr} = q_{tr} \cdot \frac{h_R}{r} \cdot \frac{p}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_{tr} = 1 - \left(\frac{\Delta r}{0.25 \cdot r} \right)$
$A_j = q_j \cdot \frac{h_j}{h_R} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_j = 1 - \left(\frac{\Delta h_j}{0.25 \cdot h_j} \right)$
$A_l = q_l \cdot \frac{l_l}{h_R} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_l = 1 - \left(\frac{\Delta l_l}{0.25 \cdot l_l} \right)$
$A_{lv} = q_{lv} \cdot \frac{l_{lv}}{h_R} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_{lv} = 1 - \left(\frac{\Delta l_{lv}}{0.25 \cdot l_{lv}} \right)$
$A_{s1} = q_s \cdot i_{s1} \cdot \frac{h_{com}}{h_R} \cdot \frac{l_s}{h_R} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_s = 1 - \left(\frac{\Delta w_s}{w_R} \right)$
$A_{s2} = q_s \cdot (-i_{s2}) \cdot \frac{h_{com}}{h_R} \cdot \frac{l_s}{h_R} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_s = 1 - \left(\frac{\Delta w_s}{w_R} \right)$
$A_{s3} = q_s \cdot i_{s3} \cdot \frac{h_{com}}{h_R} \cdot \frac{h_R}{w_R} \cdot \frac{l_{s3}}{h_R} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_s = 1 - \left(\frac{\Delta w_s}{w_R} \right)$
$A_{st1} = \phi \cdot q_{st} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_{st} = \frac{m_{success}}{10}$
$A_{st2} = q_{st} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_{st} = \frac{m_{success}}{10}$
$A_{sstep} = q_{sstep} \cdot \frac{w_s}{h_R} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_{sstep} = 1 - \frac{\Delta l_s}{0.25 \cdot l_R}$
$A_{fl} = q_{fl} \cdot \frac{l_{fl}}{h_R} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_{fl} = 1 - \frac{\Delta w_{fl}}{w_R}$
$A_{bl} = q_{bl} \cdot \frac{l_{bl}}{h_R} \cdot \frac{1}{t} \cdot \sqrt{\frac{h_R}{g}}$	$q_{bl} = 1 - \frac{\Delta w_{bl}}{w_R}$

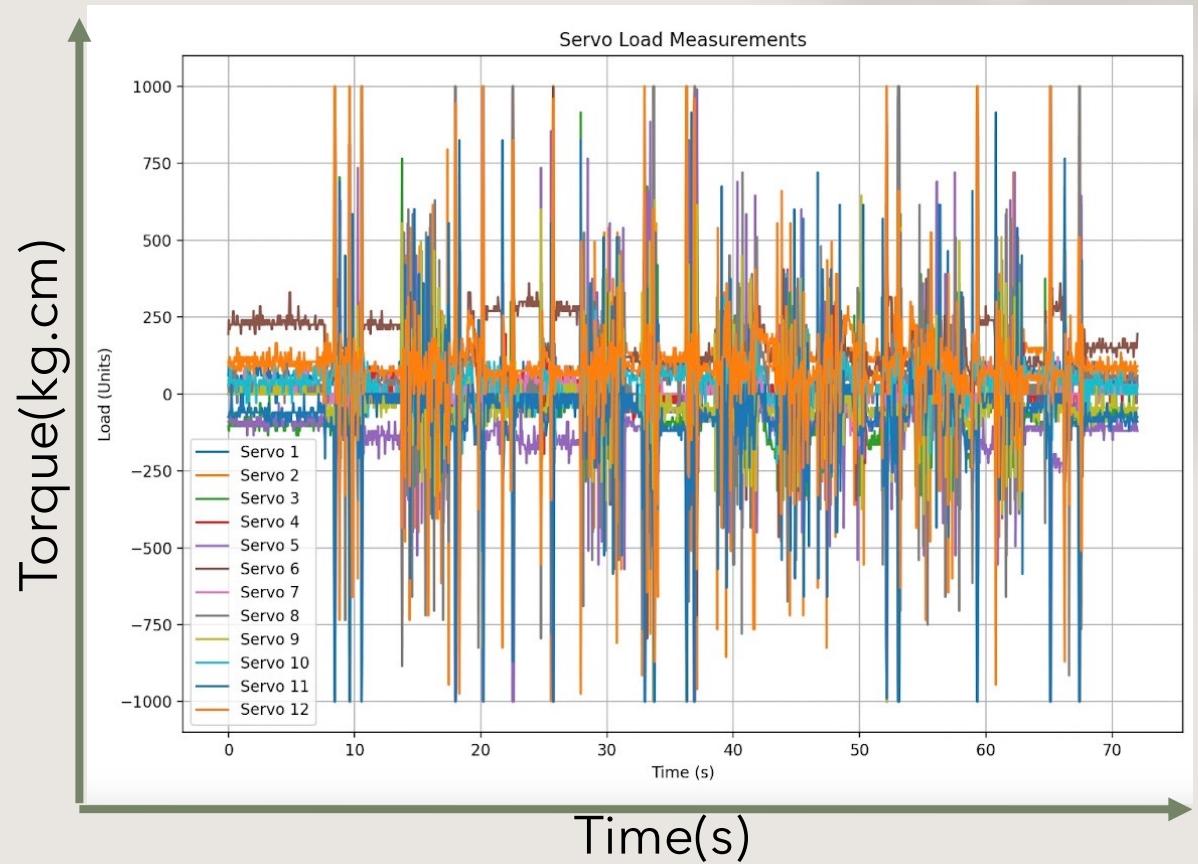
8	Ats	0.015	no q
9	Atr	x	
10	Aj		
11	Al		
12	Alv		
13	As1(downslope)	0.03274337	q = N/A
14	As2(upslope)	0.03016805	q = N/A
15	As3(sagittal plane)	0.03098841	q = N/A
16	Ast1		
17	Ast2		
18	Asstep		
19	Afl	0.062	q = N/A
20	Abl	0.102	q = N/A
21	Agav%		

Results

Gyro data for turning on spot

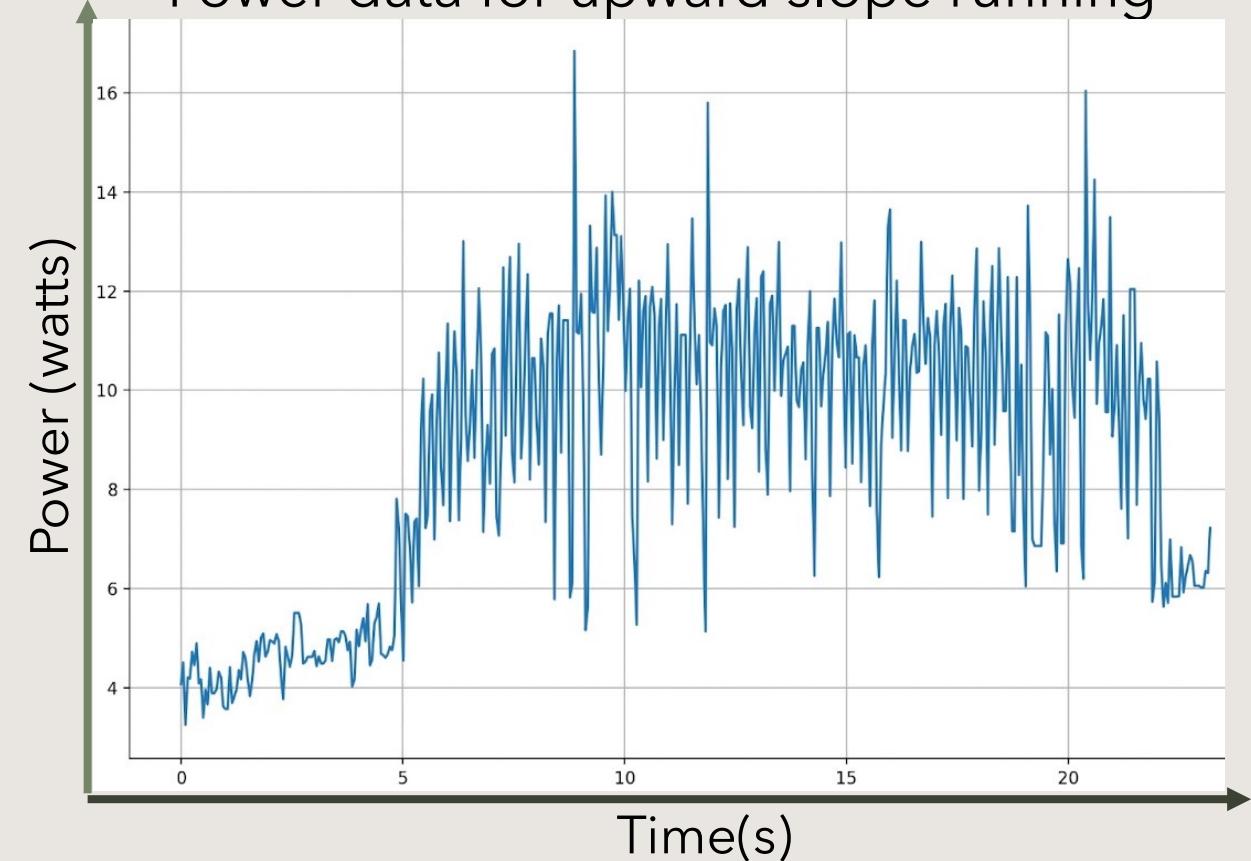


Load data for jumping

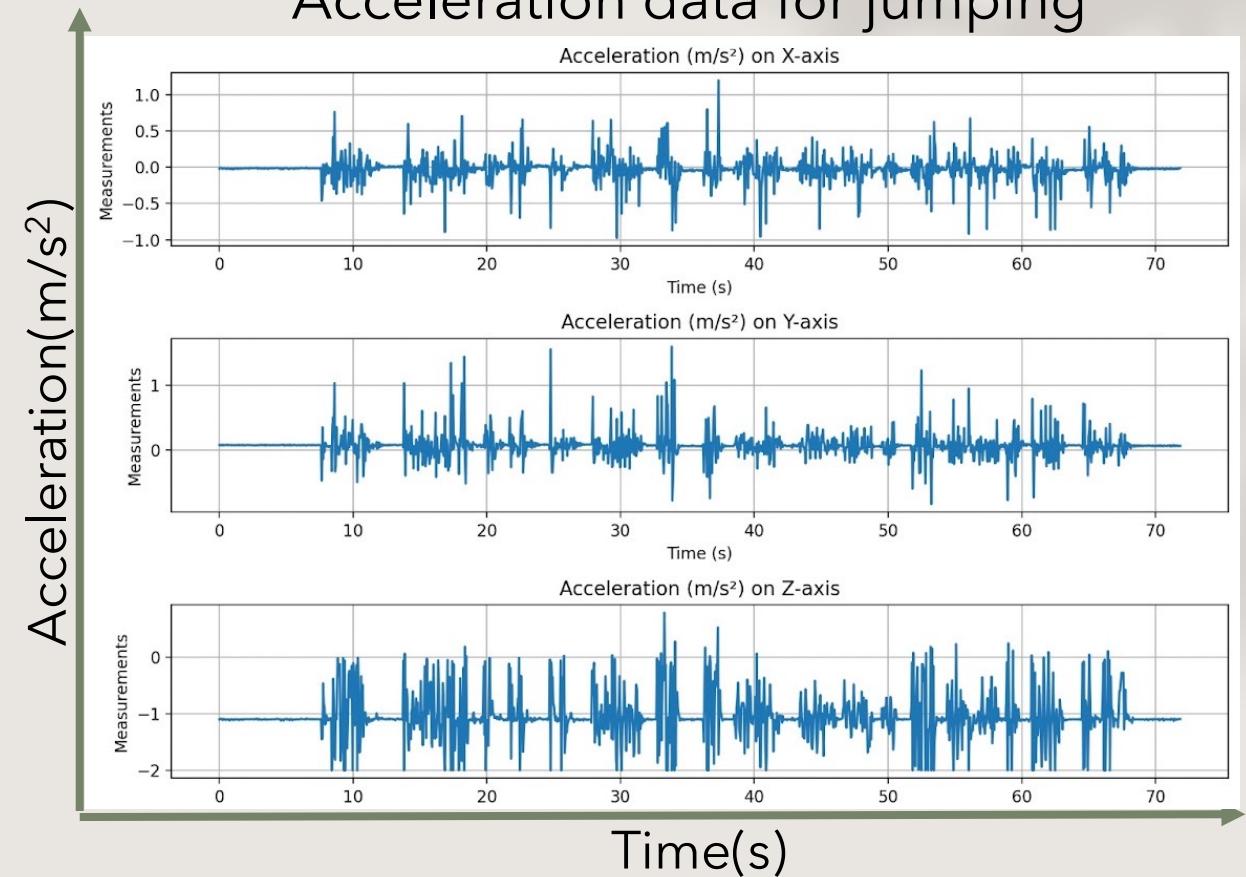


Results

Power data for upward slope running



Acceleration data for jumping



Conclusions & Future work

Conclusions:

- Designed an active aerodynamic tail
- Implemented tail control system
- Tested MiniPupper agility without tail

Future work:

- Simulate MiniPupper with tail
- Evaluate agility with tail
- Integrate Reinforcement Learning policy from cheetahs

