Active Aerodynamic Tail Enhances Agile Locomotion of Legged Robots

2nd July, 2024

Brian Goodluck Mmari

(bmmari@purdue.edu, brnmmari@gmail.com)

Prof. Upinder Kaur, Dr. Jiajun An

Abstract

Terrestrial animals have evolved over millennia to effortlessly locomote across complex terrains, such as grasslands and forests. They exhibit a variety of gait patterns such as hopping, running, and even mid-air reorientation. A common feature among these animals is their tail which helps them in dynamic stabilization through the various gait patterns. While tails are being developed to improve the performance and adaptability of legged quadruped robots, they are modeled as passive mass which can only provide limited corrective torques to help in dynamic stabilization. In this work, we develop an Active Aerodynamic Tail (AA-Tail), with two propellers integrated at the tail end to enhance agile locomotion performances of legged robots. The proposed design weighs around 150g and enables highly precise and fast reaction capabilities, enhancing the robot's dynamic stabilization. For validation, we develop and deploy the tail on a 500g quadruped robot dog, Mini Pupper 2. The simulations and experiments demonstrate that our tail effectively enhanced the legged robot's agility across various motion patterns simultaneously, including running, turning, leaping, slope running, standing up, sidestepping, and forward and backward locomotion. This approach proves the effectiveness of the tail as an active stabilization component essential for robots to mimic natural gait patterns in complex terrains.

ACKNOWLEDGEMENTS

I would first like to thank Professor Upinder Kaur for welcoming me into her research group. Her mentorship has been truly inspirational, and her unwavering commitment to answering my questions has been invaluable. I am deeply grateful for her dedication to meeting with me and providing guidance. Her emphasis on collaboration with other lab members has allowed me to gain extensive knowledge, especially when working with another student in the lab, Juan Soto. I have also gained experience in some of her other research projects, conducted fieldwork at the farm, and met collaborators from other labs. I am incredibly fortunate to be part of this research group, and I look forward to continuing to learn from Professor Kaur and her team

I would also like to express my gratitude to my direct supervisor, Dr. Jiajun An, for his exceptional mentoring and teaching. He has always encouraged me to ask questions and introduced me to an online resource for exploring alternatives. Additionally, he has been instrumental in explaining some of his other remarkable research projects to me. I appreciate his feedback on this report, abstract, and the overall project. Working with him has been a wonderful experience, and I consider myself fortunate to have had the opportunity to learn from him.

INTRODUCTION

Agile maneuvers such as rapid acceleration, abrupt stops, running jumps and near-instantaneous turns seem effortless to animals [1, 2]. As a result, many terrestrial animals have exceptional agile locomotion capabilities. For instance Springboks gracefully leap over bushes while in motion, mountain goats effortlessly bounce over cliff faces as they climb and squirrels are able to climb trees and leap between branches. However, no animal epitomizes this more than the cheetah (*Acinonyx jubatus*)[3]. Consider a cheetah in high-speed pursuit of its prey: the animal will rapidly re-adjust its trajectory in response to the evasive maneuvers of the prey. It is this rapid, sensory driven feedback which future robots will need to harness for robust high speed locomotion[3].

Inspired by animals, tails have been proposed to enhance robotic agility across various motions, including hopping, rapid turning, jumping, mid-air reorientation, acceleration/deceleration, and flipping. As a result, the robotics community has recently seen a surge of interest into the notion of actuated tails[3]. For example, researchers at MIT have also investigated a Cheetah-inspired tail for attitude control during the airborne phase as well as disturbance rejection (at standstill)[3].Lastly, Patel and Braae [9] have demonstrated that an actuated tail can increase the acceleration capability of quadruped robots[3]. It is shown that by actuating the tail, one can increase the capture region, thus implying more aggressive maneuvers are possible[3]. Aggressive maneuvers means better agility which implies closer resemblance to actual animals.

Agility can be defined as the ability to perform a set of different but specific tasks executed in a fast and efficient manner[4].This definition is inspired by the analysis of natural role models, such as dogs and horses as well as robotic systems[4]. Different researchers have come up with different metrics to analyze agility in robotics. Well defined scores like the Froude number used by Alexander [5] or the normalization of the robots' speed to body-lengths per second (BL/s) in combina- tion with the cost of transport (CoT) [6] are accepted throughout the community. These metrics, in my opinion, offer ease of use. Other scientists proposed a framework for benchmarking versatility in comparison to the robots complexity[7]. Although this approach showed many exciting ideas, a big challenge is the complexity of the method itself as it includes comparisons of land, water, and aerial robots in one framework, which makes it easy to get confused[4]. Other methods such as one explored on the leaping quadruped Canid[8] of measuring agility scores produces non dimensionless quantities and thus scaling effects have to be taken into account when comparing different robots[[4]. As a result this paper has chosen a more standardized approach to measure agility metrics as elaborated by Peter Eckert and Auke Ijspeert[4], because all scores are normalized and dimensionless speeds, guaranteeing comparability between different robots and are only applicable to legged quadrupeds, hence easily transferable to our project.

ROBOT PLATFORM

The proposed robot platform used is the Mini Pupper 2, a quadruped robot manufactured by MangDang Robotics Company. This quadruped robot has a mass of 500g and is both lightweight and easily configurable. The robot is equipped with an IMU sensor, speaker, touch sensors, Raspberry Pi 4B, Compute Module 4, Arduino(ESP32) and offers servo feedback specifically position data.

Assembly of the robot was the first task and this was done per the guidelines posted on this link: https://minipupperdocs.readthedocs.io/en/latest/guide/AssembleMiniPupper2.html. Any issues/problems encountered during the process were easily resolved by getting in touch with the Discord community where similar questions could be searched or even brought forward.

Figure 1: Mini Pupper 2

The Mini Pupper 2 robot was already flashed with Ubuntu 22.04 and ROS2 Humble in its system. However, the robot also supports ROS1 as well as Ubuntu

20.04. However, the user can choose to flash the same images from their website at the following link: Mini Pupper pre-built images. In addition the Mini Pupper is equipped with an ESP32 which controls all the servos and as a result also came pre-flashed with its own image. https://github.com/mangdangroboticsclub/mini_pupper_2_bsp/tree/main/esp32. The easiest and most recommended way of accessing the robot was establishing an ssh connection to the robot, with the IP address of the robot displayed on its screen.

I was also tasked with enabling movement of the robot from the keyboard. This task involved cloning online the PupperKeyboardController repository found at the following link: https://github.com/stanfordroboticsclub/PupperKeyboardController.

The definitions for the movements setup by the repo for each key are also outlined in the repository.

EXPERIMENTAL RESULTS

Before constructing the metrics table, we first sought to gather the geometrical values for our Mini Pupper 2. All of the videos, experimental setups and data discussed in this paper are also included in the supplementary google drive folder submission Link .

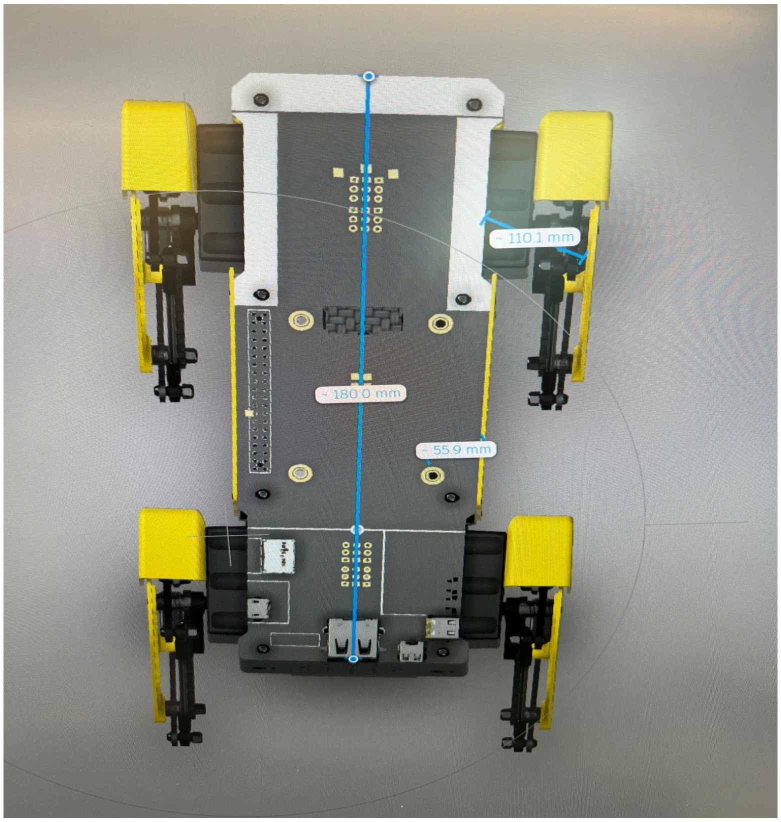


Figure 2: Length of robot, l_R = 0.18m

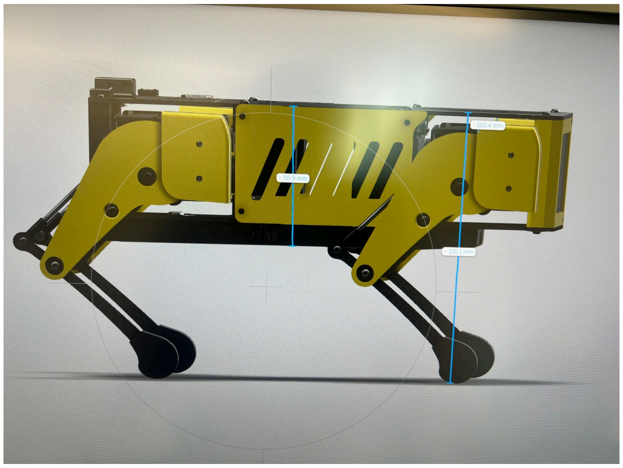


Figure 3: Height of robot, h_R = 0.1101m, h_{COM} = 0.02795m

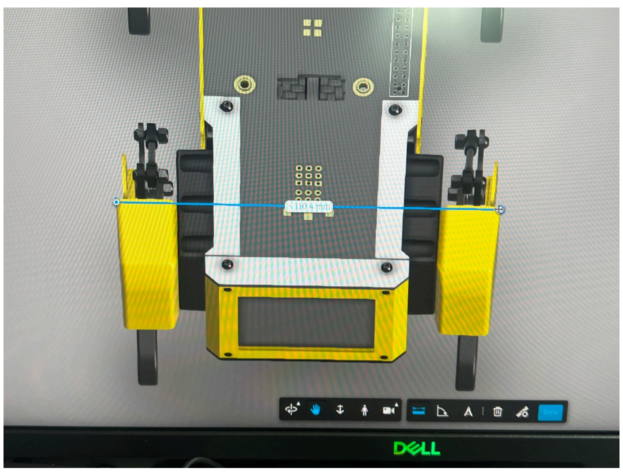


Figure 4: width of the robot, wR = 0.1104m

After obtaining the geometrical values for our Mini Pupper, the second task was establishing a metrics table to understand how agile our Mini Pupper 2 robot was without the tail. Based on the agility metrics that were discussed by Peter Eckert and Auke Ijspeert, we choose to consider 5 metrics based on easiness of setting up experiments and ability of our robot.

1) Forward and Backward Locomotion

This is the most known locomotion type in mobile robotics, straight forward and backward locomotion (A_{fl} /A_{bl})[4]. To calculate, we need the respective distance traveled forward l_{fl} [m] and backward l_{bl} [m] normalized with the robot height h_R [m] and the measured time of the respective movement t[s] in the dimensionless form. The variance is again the deviation from a straight path with respect to robot width wR [m] after a distance of *ten body-lengths*[4].

- $Afl = q_{fl} * \frac{lfl}{hR} * \frac{1}{t} * \sqrt{(\frac{hR}{g})}$
- $Abl = q_{bl} * \frac{lbl}{hR} * \frac{1}{t} * \sqrt{(\frac{hR}{g})}$

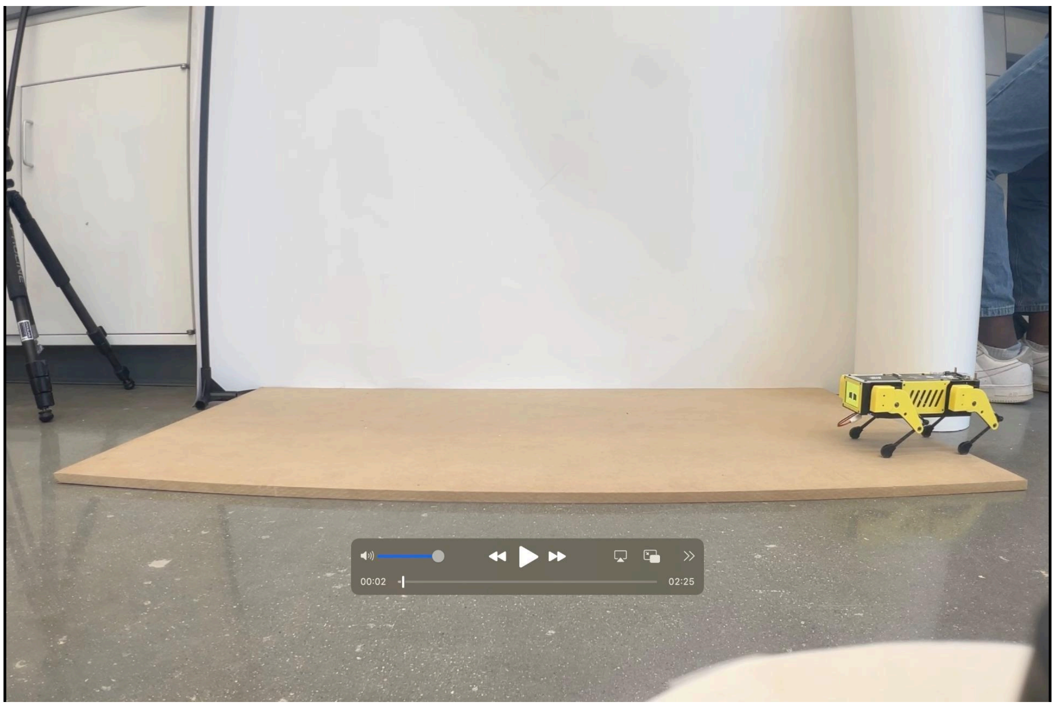


Figure 5: Setup for forward locomotion

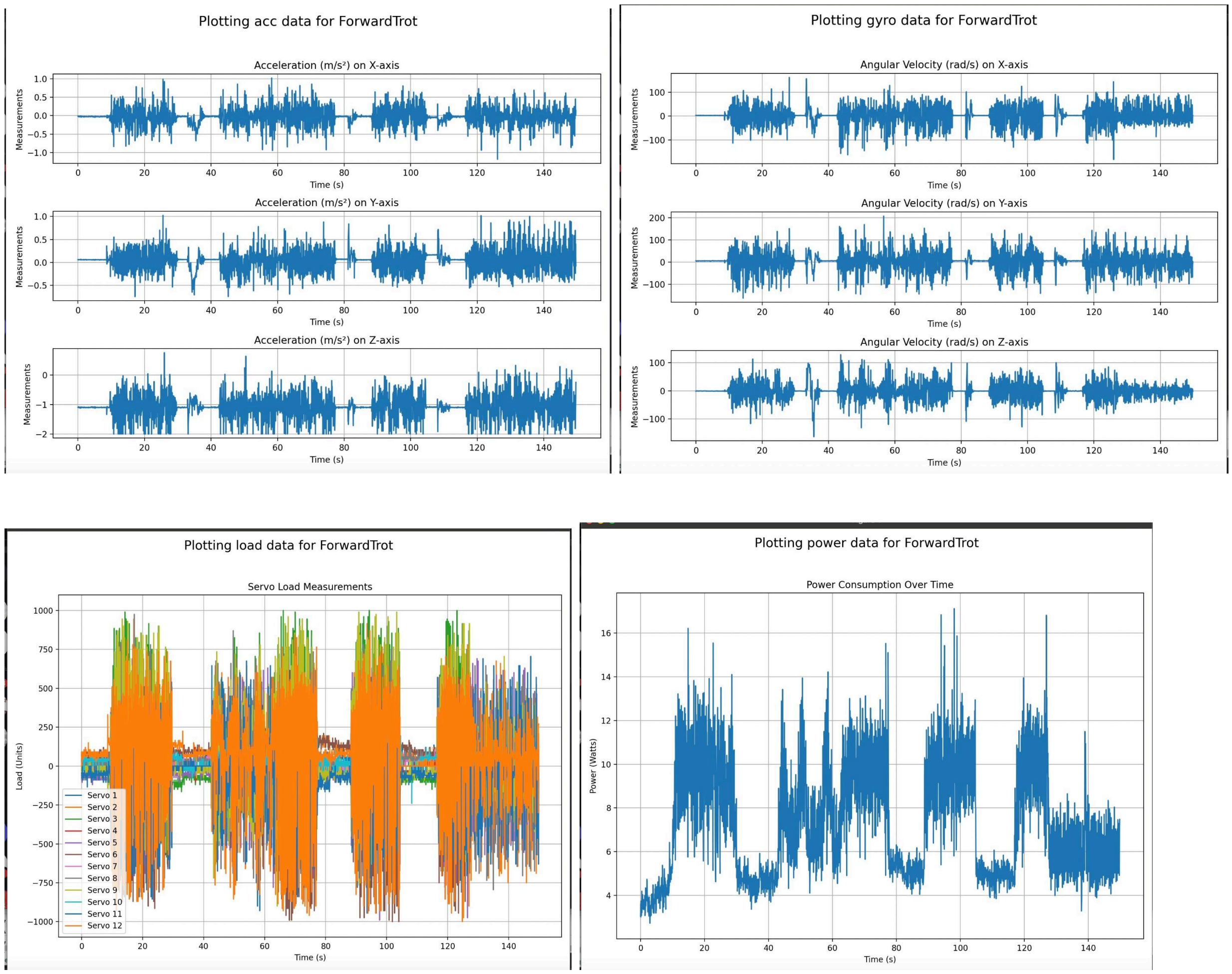


Figure 6: IMU data obtained from forward locomotion

Figure 7: Setup for backward locomotion

From the above equation and data that was collected from the experiments, the following results were obtained:

$$A_{fl} = 0.062,\ A_{bl} = 0.102, \text{ with } q = 1$$

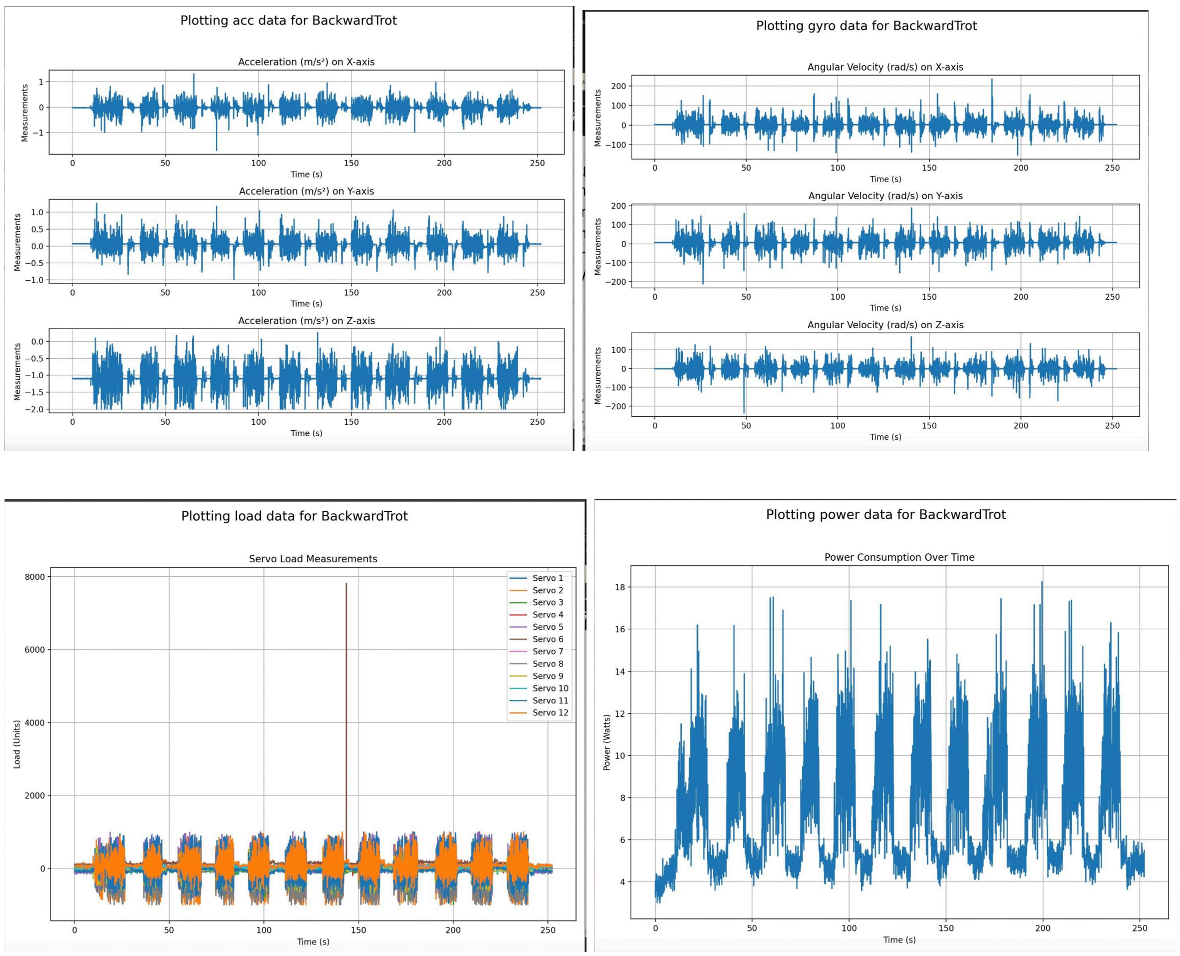


Figure 8: IMU data obtained from backward locomotion.

2) Slope Running $A_{s\,(1-3)}$

Slopes up- and downward, with the same calculation but one working with and one against gravity, as well as slopes inclined toward the sagittal plane of the robot and thus orthogonal to the movement direction should be considered[4].

The variance of the performance influences the measure with q_s where the percentile deviation from a straight path after a distance of *ten body-lengths* in respect to the robot width is calculated[4].

- $A_{s1} = qs * i_{s1} * \frac{hcom}{hR} * \frac{ls}{hR} * \frac{1}{t} * \sqrt{(\frac{hR}{g})}$.
- $A_{s2} = qs * (-i_{s2}) * \frac{hcom}{hR} * \frac{ls}{hR} * \frac{1}{t} * \sqrt{(\frac{hR}{g})}$
- $A_{s3} = qs * (i_{s3}) * \frac{hcom}{hR} * \frac{hR}{wR} * \frac{ls3}{hR} * \frac{1}{t} * \sqrt{(\frac{hR}{g})}$,
- $q_s = 1 - (\frac{\Delta ws}{wR})$

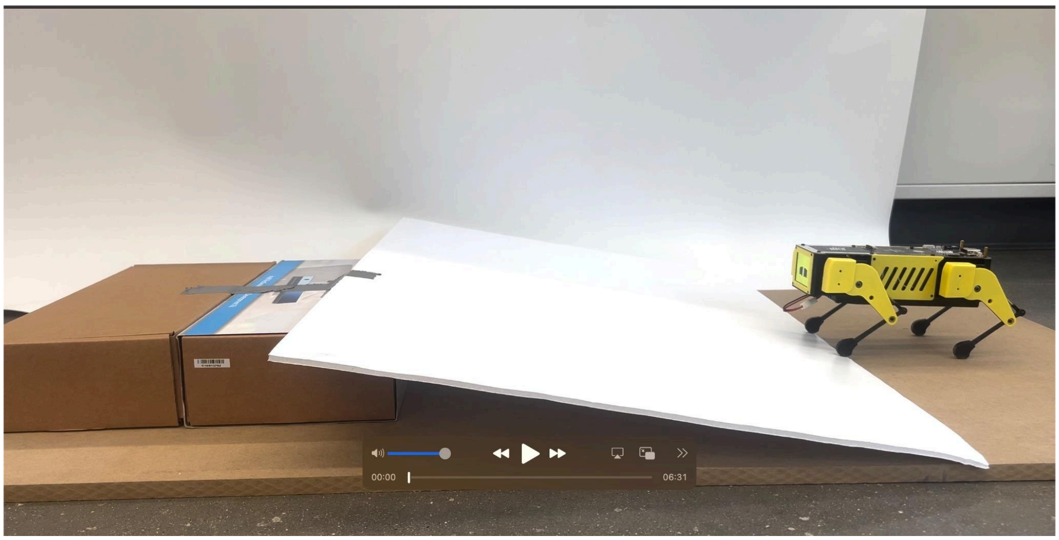
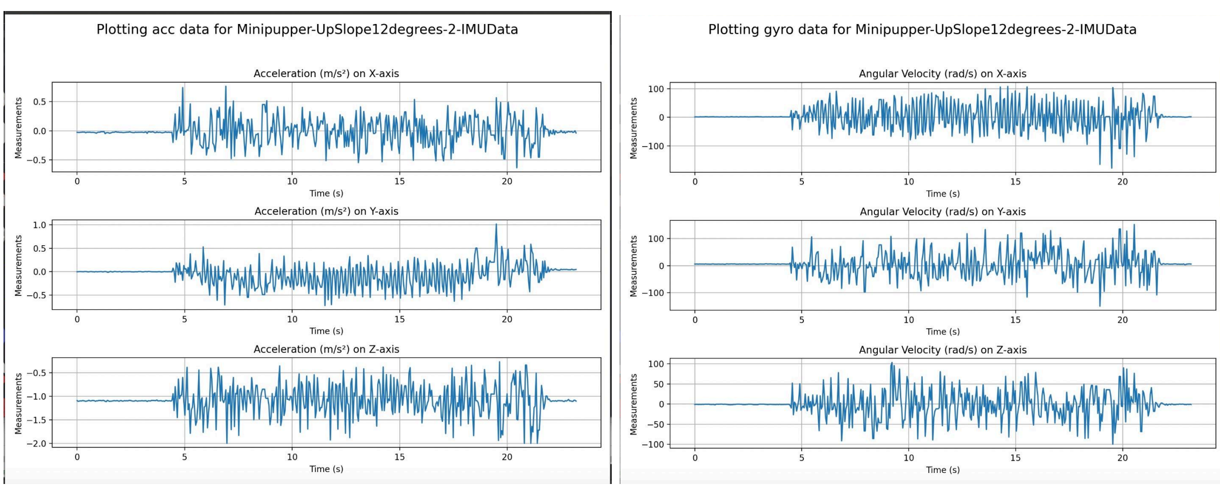


Figure 9: Setup for measuring A_{s2}, with plane inclined at 12°.



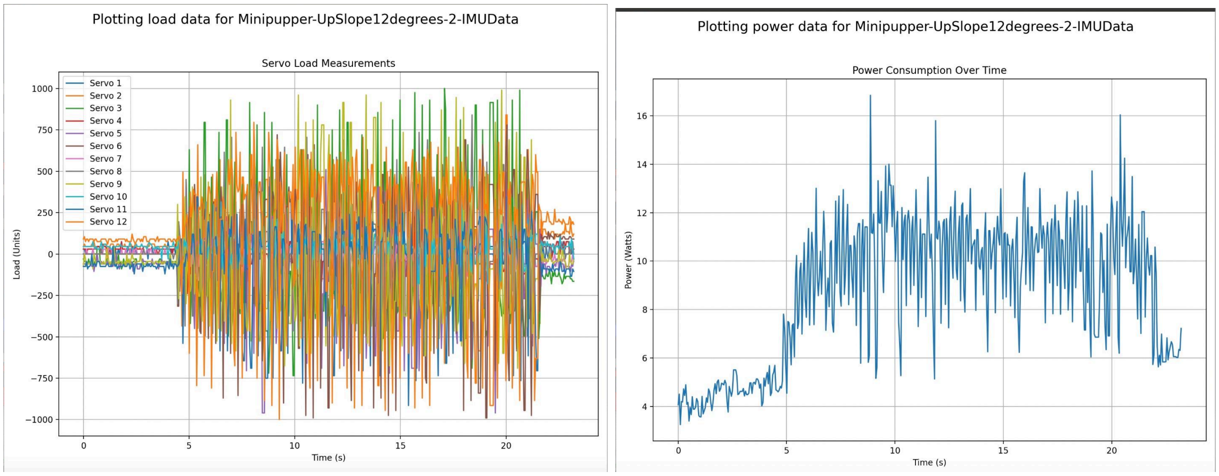


Figure 10: IMU data obtained from slope running against gravity locomotion.

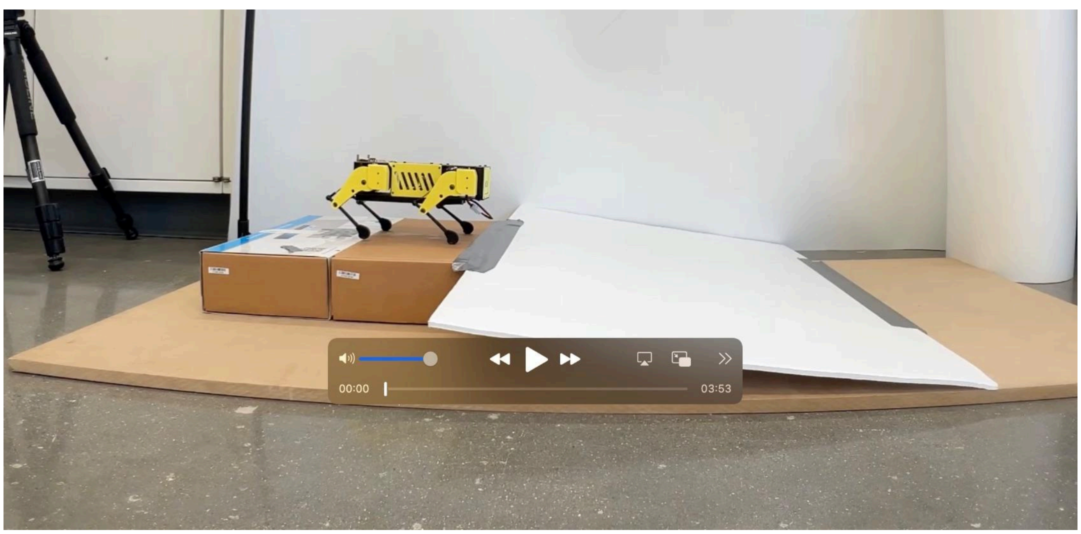


Figure 11: Setup for measuring A_{s1}, with plane inclined at 6°.

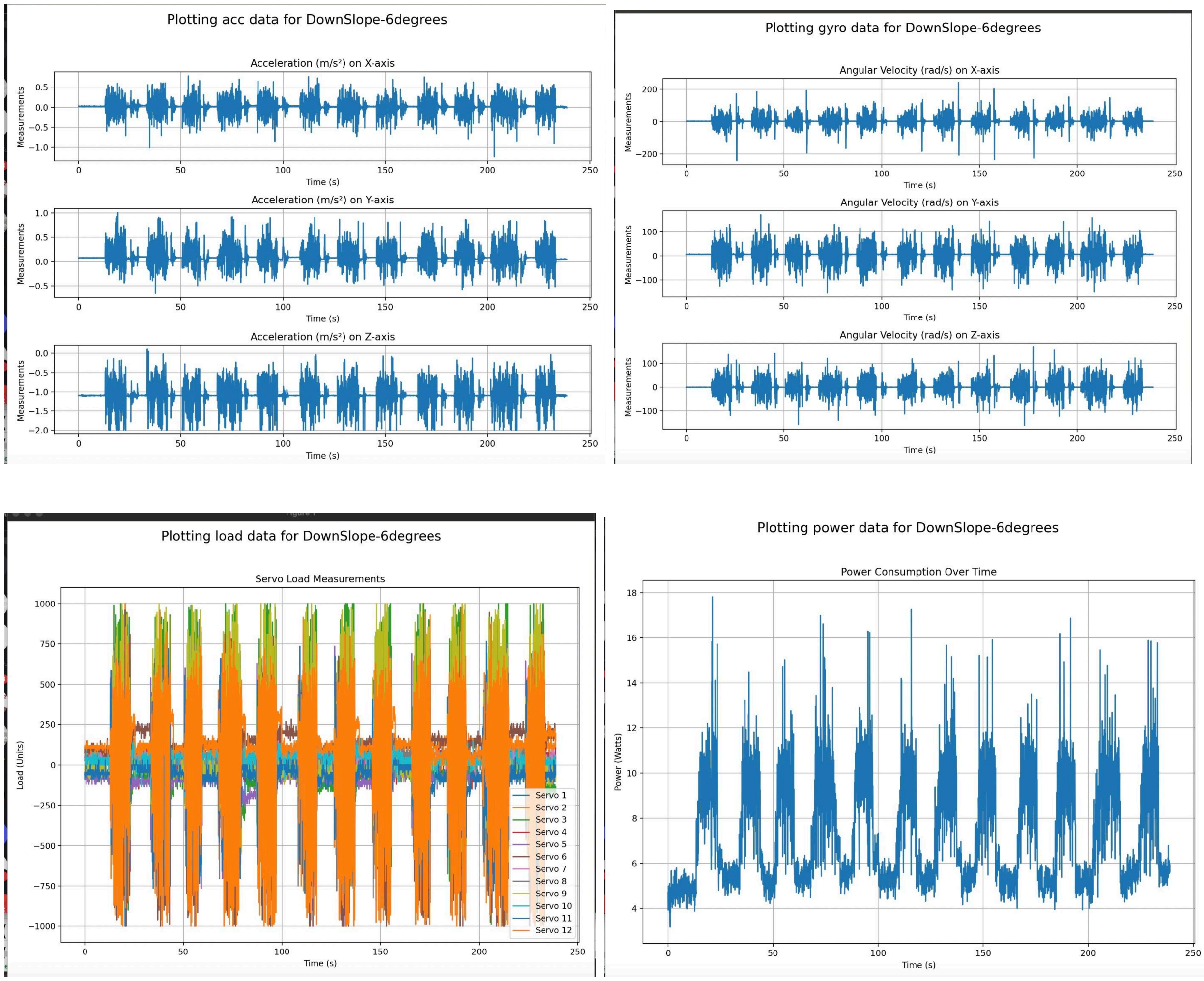


Figure 12: IMU data obtained from slope running with gravity locomotion.

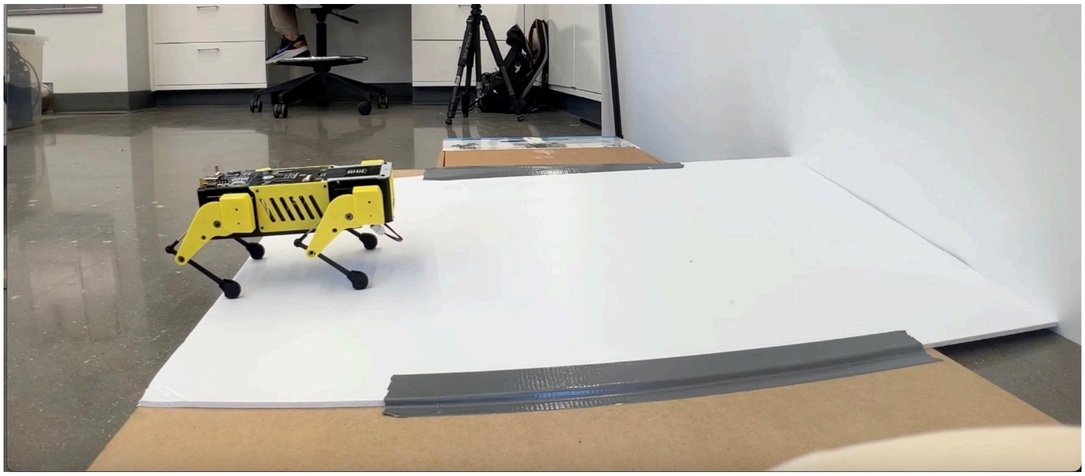


Figure 13: Setup for measuring A_{s3}, with plane inclined at 6°.

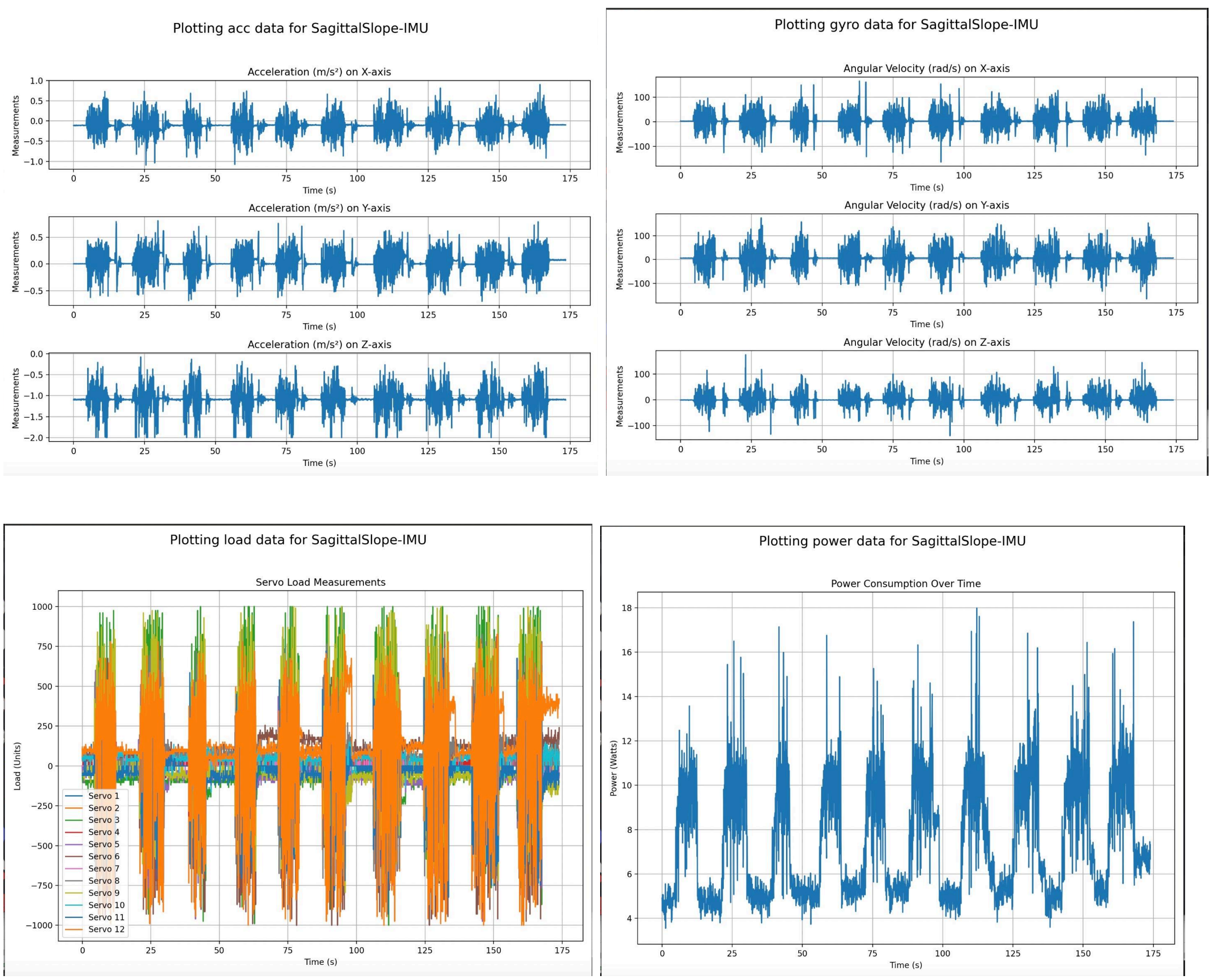


Figure 14: IMU data obtained from running a sagittal slope locomotion.

From the above equations and data that was collected from the experiments, the following results were obtained:

A_{s1} = 0.033 , A_{s2} = 0.030, and A_{s3} = 0.031, q = N/A

3) Turning on the Spot A_{ts}

A_{ts} is chosen if the robot's rotational axis is exactly in the geometric center, otherwise turning with a radius applies[4]. For our case, the Mini Pupper's rotational axis was also located at the geometric center, and hence this metric was also chosen.

- $A_{ts} = \frac{p}{t} * \sqrt{(\frac{hR}{g})}$

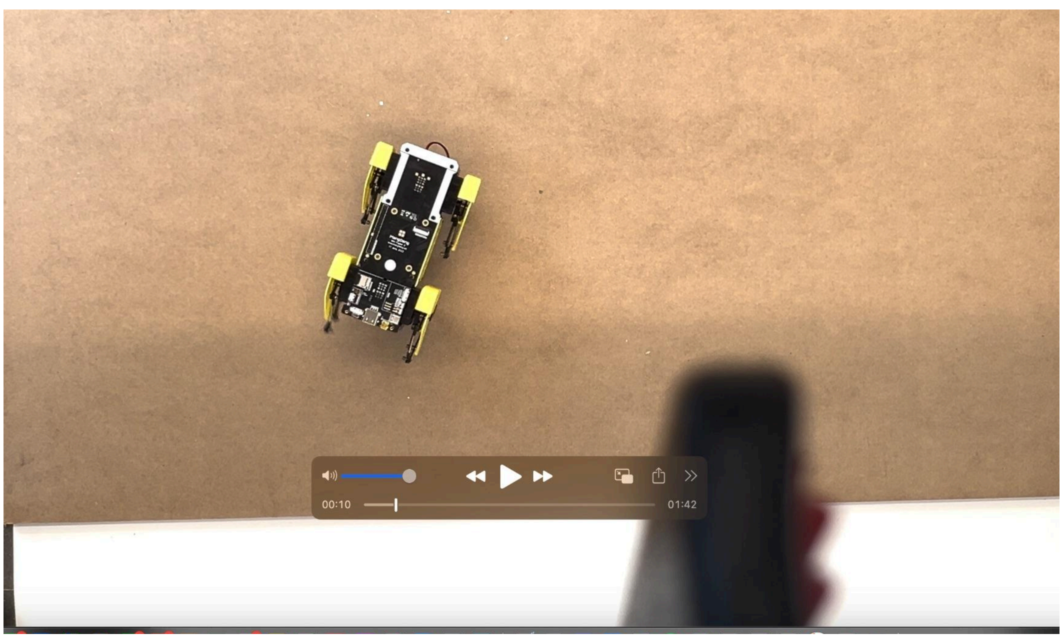


Figure 15: Setup for measuring A_{ts}

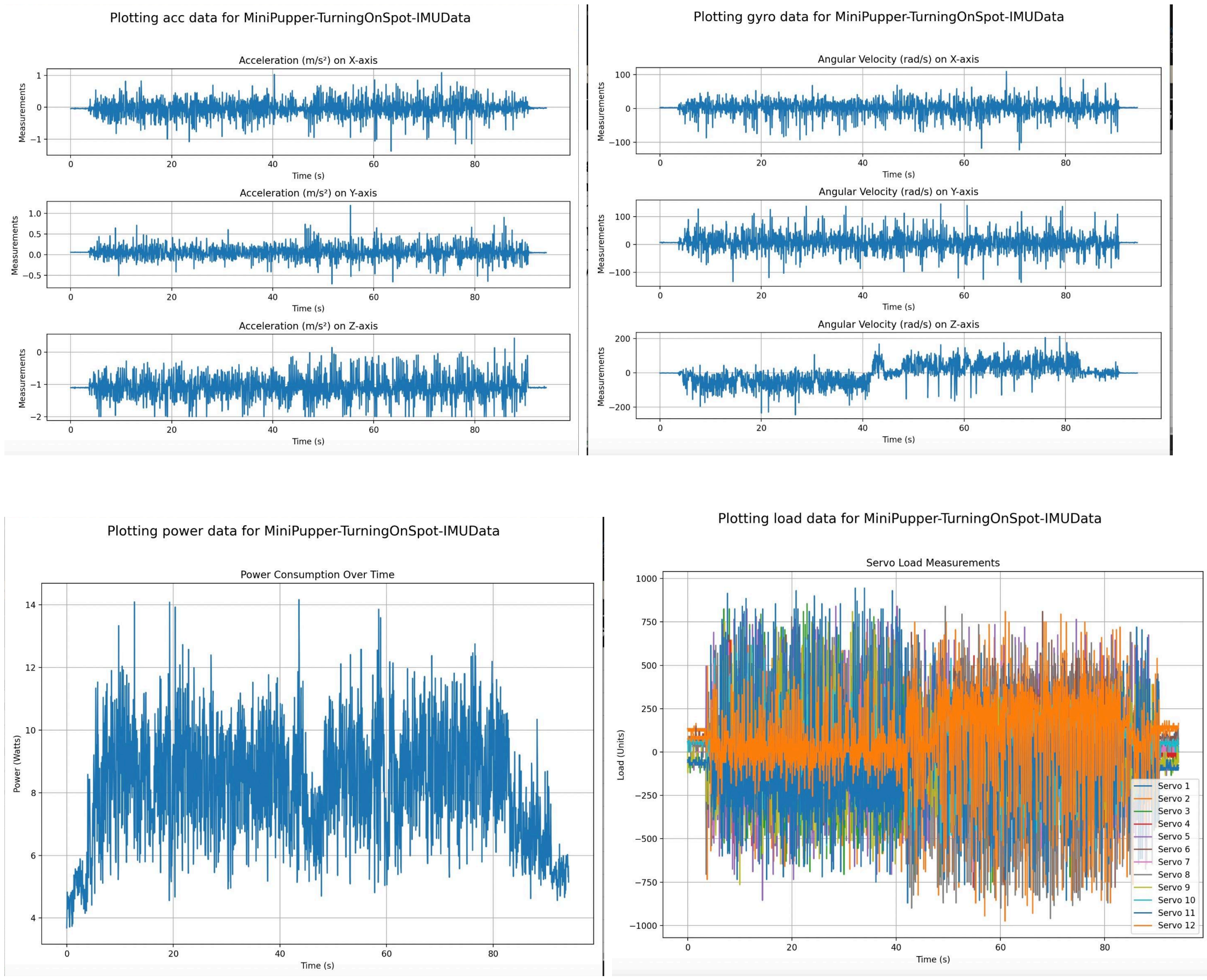


Figure 16: IMU data obtained from turning on spot locomotion

From the above equation and data that was collected from the experiments, the following results were obtained:

$A_{ts} = 0.015$

4) Jumping A_j and Leaping $A_{l(v)}$

Jumping has no or only little horizontal movement as it is describing how high the robot can jump, whereas one focuses on the horizontally traveled distance in air when talking about leaping[4]. Leaping was not a pre-set command in the minipupper and hence this was ignored for this setup. Jumping was also not as efficient as we would have thought but data was still recorded since adding a robotic tail would eventually enhance this actions' performance.

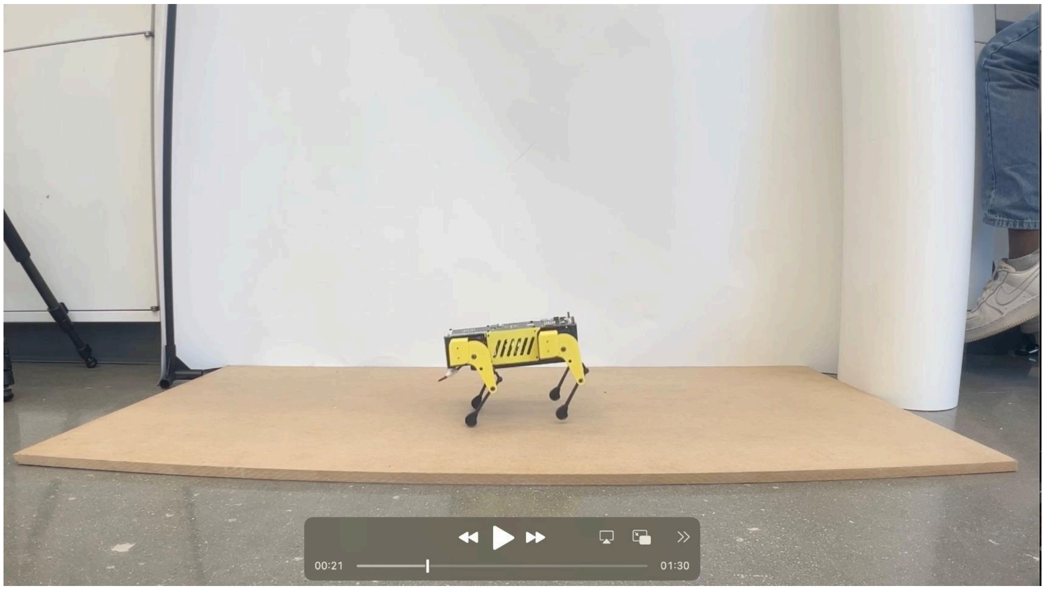


Figure 17: Setup for measuring A_j

Measurements for the agility metrics for this setup haven't been recorded yet because currently our robot doesn't jump as high enough. We're exploring a code script that uses pose estimates marked using DeepLabCut to calculate distances moved by points from one frame to another. The link to the code is here: https://github.com/farhanaugustine/DeepLabCut-Analysis-Jupyter-Scripts/blob/main/DLC_ROI_Distance_Velocity_Entries.ipynb

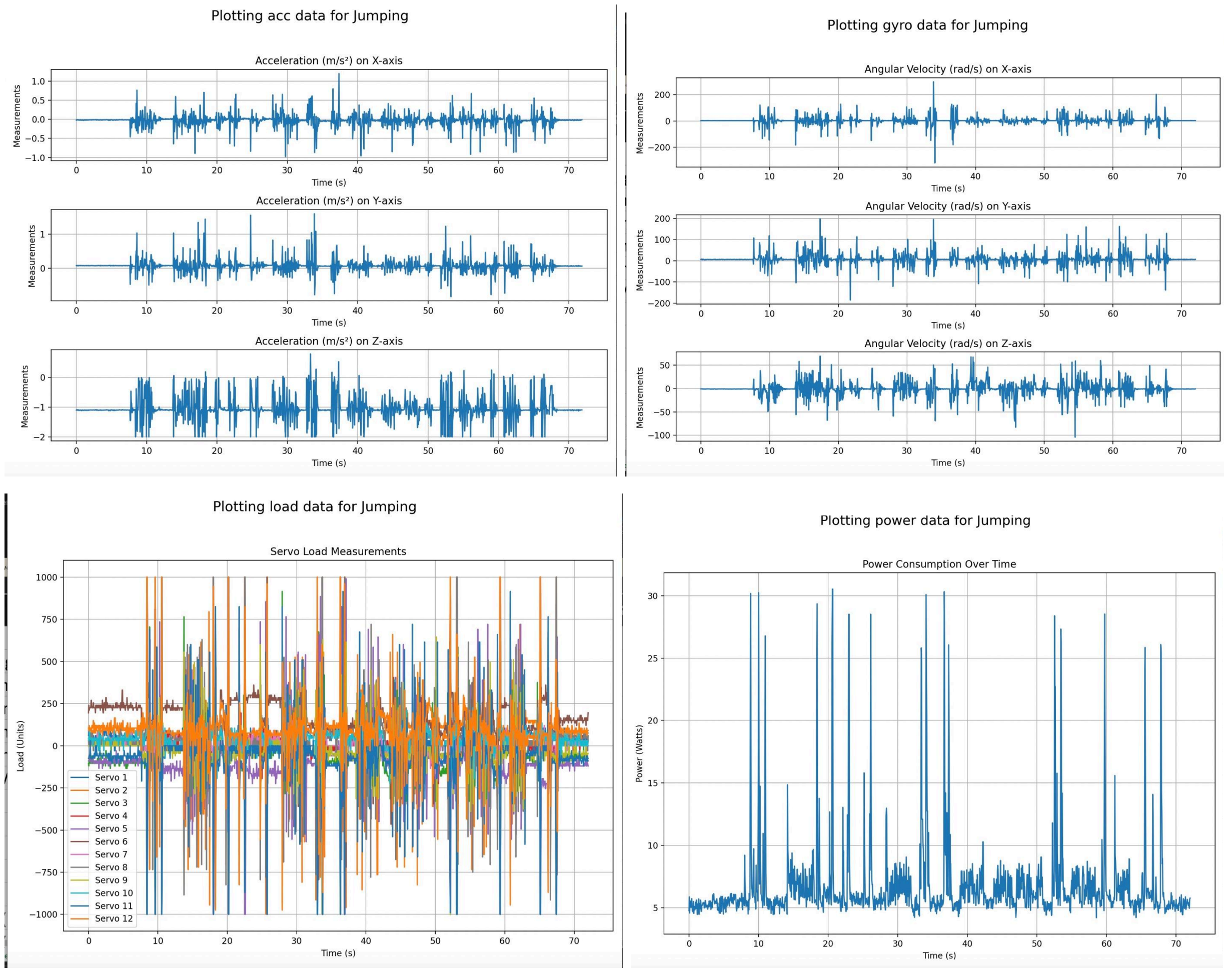


Figure 18: IMU data obtained from jumping locomotion

5) Sidestep (Nonholonomic) A_{sstep}

Moving sideward is defined through the width of one step w_S [m] normalized by the robot height hR [m] and the time needed to perform the maneuver t [s] in its dimensionless form, leading to the score A_{sstep}. q_{sstep} [%] describes the variance of the sidestep by relating the deviation from a straight path Δl_S [m] in terms of robot length l_R [m] after ten steps.

- $A_{sstep} = q_{sstep} * \frac{ws}{hR} * \frac{1}{t} * \sqrt{(\frac{hR}{g})}$

- $q_{sstep} = 1 - \left(\frac{\Delta ls}{0.25 * lR}\right)$

Measurements for the agility metrics have also not been recorded yet, as we're currently trying to analyze the best way to determine width of one step simply because the distances are so small. Our current approach is to look at the configuration files to determine the pre-set distance for such a movement.

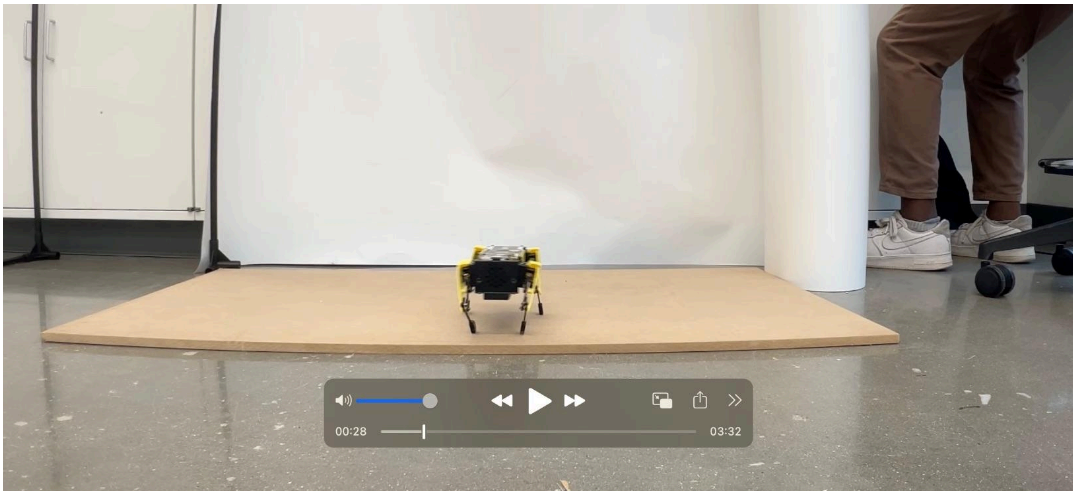


Figure 19: Setup for measuring A_{sstep}

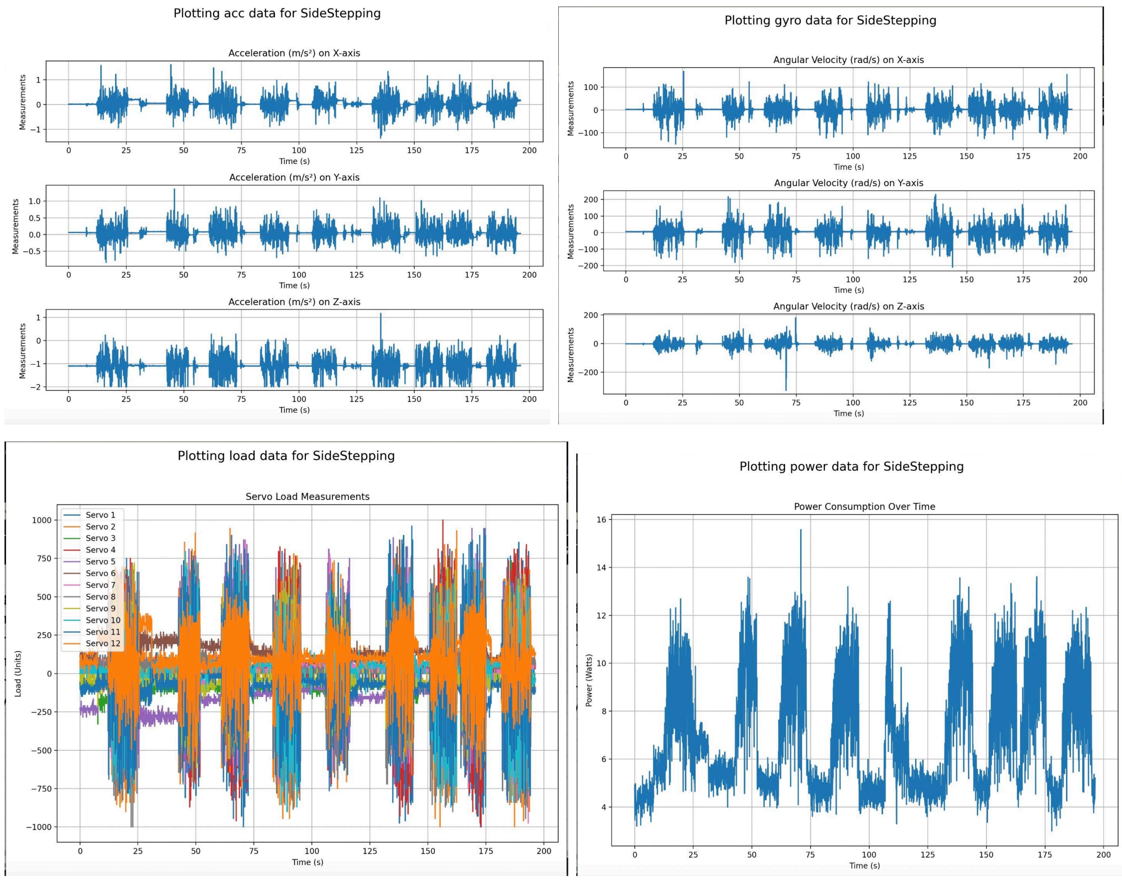


Figure 20: IMU data obtained from side stepping locomotion

In this article, we present an Active Aerodynamic Tail (AA-Tail) (Fig. 1A) to enhance agile locomotion of legged robots. By placing two propellers at the tail end, the AT-Tail can produce continuous large aerodynamic thrust without relying on the tail's swing range of motion. The long tail length allows the generated continuous thrust to produce sustained moments. With three actuators to adjust the position and direction of the propellers, we can manipulate the forces and moments in various directions. The thrust differential between the two propellers can create small torques and large forces, aiding in balance adjustments during movement. To demonstrate this, we integrated the AT-Tail into a 500 g quadruped robot (Mini Pupper).

Figure 21: Mini pupper 2 with Active-Aerodynamic Tail

We present a model of the AT-Tail and compare it with the aerodynamic drag tail and inertial adjustment tail based on robot dynamics. Using the agility metric proposed for multi-legged robots in [1], we intend to demonstrate that the AT-Tail effectively enhances robot agility across various motion patterns simultaneously, including running (Fig. 1B), turning (Fig. 1C), leaping (Fig. 1D), slope running (Fig. 1E), standing up (Fig. 1F), sidestepping (Fig. 1G), and forward and backward locomotion (Fig. 1H).

Figure 22: Active-Aerodynamic Tail

Our aerodynamic tail offers 3 degrees of freedom: yaw, pitch and roll. The intricate mechanism of the tail is offered in the diagram below:

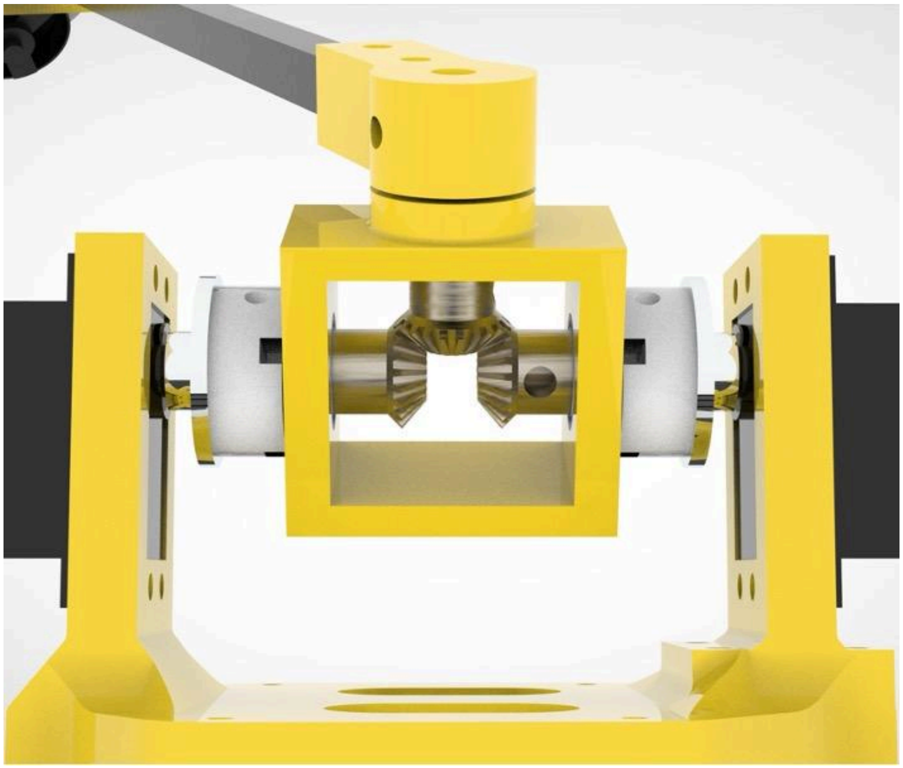


Figure 23: Zoomed image of the aerodynamic tail

To control and power our tail, including our propellor we decided to use some electronic components listed below:

- ESC controller, BLHeli_S

Link: https://www.getfpv.com/lumenier-36a-blheli-32-32bit-2-6s-w-telemetry.html?utm_source=google&utm_medium=cpc&utm_campaign=DM+-+NB+-+PMax+-+Shop+-+Under-index+-+SM+-+ALL+%7C+Full+Funnel&utm_content=pmax_x&utm_keyword=&utm_matchtype=&campaign_id=20800417087&network=x&device=c&gc_id=20800417087&gad_source=1&gbraid=0AAAAAD8cN5KXgB6sPyxibRhBxjkJ3HY8P&gclid=CjwKCAjw5Ky1BhAgEiwA5jGujkFMYdymiS06D3ovGxQZCO5oh1skvJ2EionM_2YahvGvFp0QvZyU2RoCJJ8QAvD_BwE

- 2 * ESP32 DEVKITV1

Link:https://www.amazon.com/ESP32-WROOM-32-Development-ESP-32S-Bluetooth-Arduino/dp/B084KWNMM4

- 2 * FeeTech Servos STS3032

Link:
https://www.aliexpress.us/item/3256805949137987.html?src=google&src=
google&albch=shopping&acnt=708-803-3821&slnk=&plac=&mtctp=&albbt
=Google_7_shopping&gclsrc=aw.ds&albagn=888888&isSmbAutoCall=fals
e&needSmbHouyi=false&src=google&albch=shopping&acnt=708-803-382
1&slnk=&plac=&mtctp=&albbt=Google_7_shopping&gclsrc=aw.ds&albagn
=888888&ds_e_adid=&ds_e_matchtype=&ds_e_device=c&ds_e_network=
x&ds_e_product_group_id=&ds_e_product_id=en3256805949137987&ds_
e_product_merchant_id=106594673&ds_e_product_country=US&ds_e_pr
oduct_language=en&ds_e_product_channel=online&ds_e_product_store_i
d=&ds_url_v=2&albcp=19108282527&albag=&isSmbAutoCall=false&need
SmbHouyi=false&gad_source=1&gbraid=0AAAAD6I-hGhkjoLOGmZ6Tdt
KYvy7WI9_&gclid=CjwKCAjw5Ky1BhAgEiwA5jGujmOwsddILMpMClmg6Q
Cwx7r2CIRTX9Bb_uWZpir4aQKZqMjTTPdW7hoCZrYQAvD_BwE&aff_fcid
=eeaf1b85a0224167aae499279025e405-1722536345250-09396-UneMJZ
Vf&aff_fsk=UneMJZVf&aff_platform=aaf&sk=UneMJZVf&aff_trace_key=ee
af1b85a0224167aae499279025e405-1722536345250-09396-UneMJZVf&t
erminal_id=a12d3e76b33e4db0b379c235a363eb0a&afSmartRedirect=n&g
atewayAdapt=glo2usa

- Matech Mini Power Hub Power Distribution Board

Link:https://www.amazon.com/Matek-Distribution-PDB-XT60-Quadcopter-
QAV210/dp/B0D5RFXT6Z/ref=sr_1_1?crid=2ZBOKW0U0PLC3&dib=eyJ2I
joiMSJ9.gwogkL-zoU_IqjVuN61mnDQitMRG_urzh0Cf-p_e4ddZXeiiMyC4S
POSU_IDRK3e9jBCJKUIs4YMxB3vU97nB89AU8JZA_dqvmjE8fhSpYVwn
_0KOgwjCq-i_6nbBQ-9DK9LreUbLoSAUxvEiSgrwD7pleueiRuMpaPl4OGd
WD69kGMvhtdIDY0hMwnlJR6pN1sYkLokJdtwr2SU3Snf-9UGoBLeGEXc0
sEkNgT2INk.ZMCLyPBXSylWn5XtUcyEHv2FYVPQvaJ3urFWoCA4CDU&
dib_tag=se&keywords=mini+power+distribution+board&qid=1722536309&
sprefix=mini+power+distribut%2Caps%2C119&sr=8-1

- N20 Motor

- 7.6V(2S) LiPo Battery

Link:
https://www.amazon.com/OVONIC-Connector-Airplane-Helicopter-Quadcopter/dp/B07L6BVRDG/ref=asc_df_B07L6BVRDG/?tag=hyprod-20&linkCode=df0&hvadid=692875362841&hvpos=&hvnetw=g&hvrand=694216092192169716&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=1022361&hvtargid=pla-2281435178818&psc=1&mcid=a791b7b71d1d38eb8885a1bd4b9015ae&hvocijid=694216092192169716-B07L6BVRDG-&hvexpln=73&gad_source=1

- QX- Brushless Fan Motor

Link:
https://www.qx-motor.net/products/qx-motor-64mm-edf-3800kv-brushless-motor-with-12-blades-ducted-fan-for-rc-airplane-parts

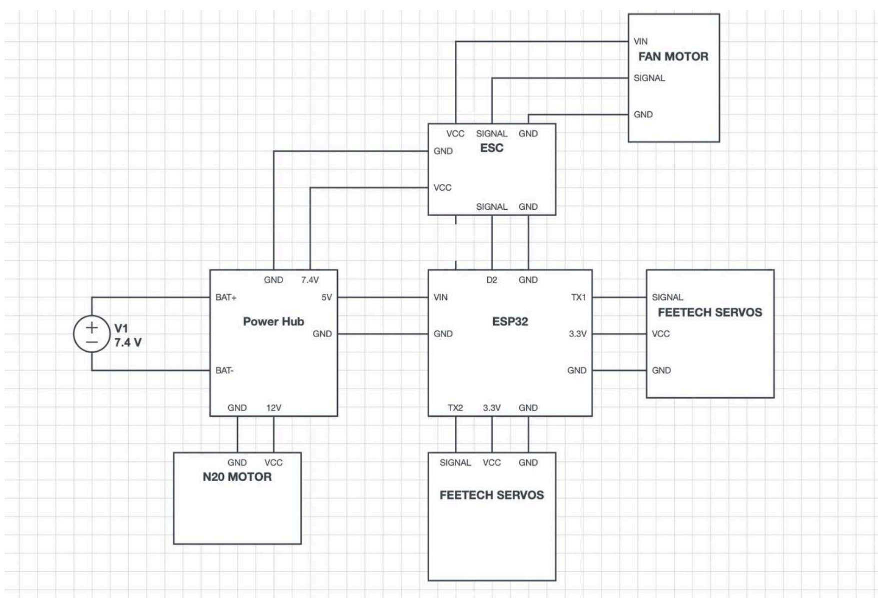


Figure 24: Circuit schematics

Our tail was controlled by employing the ESP_NOW communication protocol. To achieve this we used 2 ESP32's, one connected to the PC and the ESP32 connected to the tail robot. From there, we wrote an Arduino script that could send in commands from the Serial Monitor of my PC to the Serial Monitor of the Tail.

The arduino codes along with all the necessary files including the videos of the tail moving are found in the **Google Drive** listed below.

Transmitter code: MiniPupper_Trans.ino

Receiver code: MiniPupper_Rec.ino

From the experiments we did, it was found out that our tail can generate a force of 0.86N.

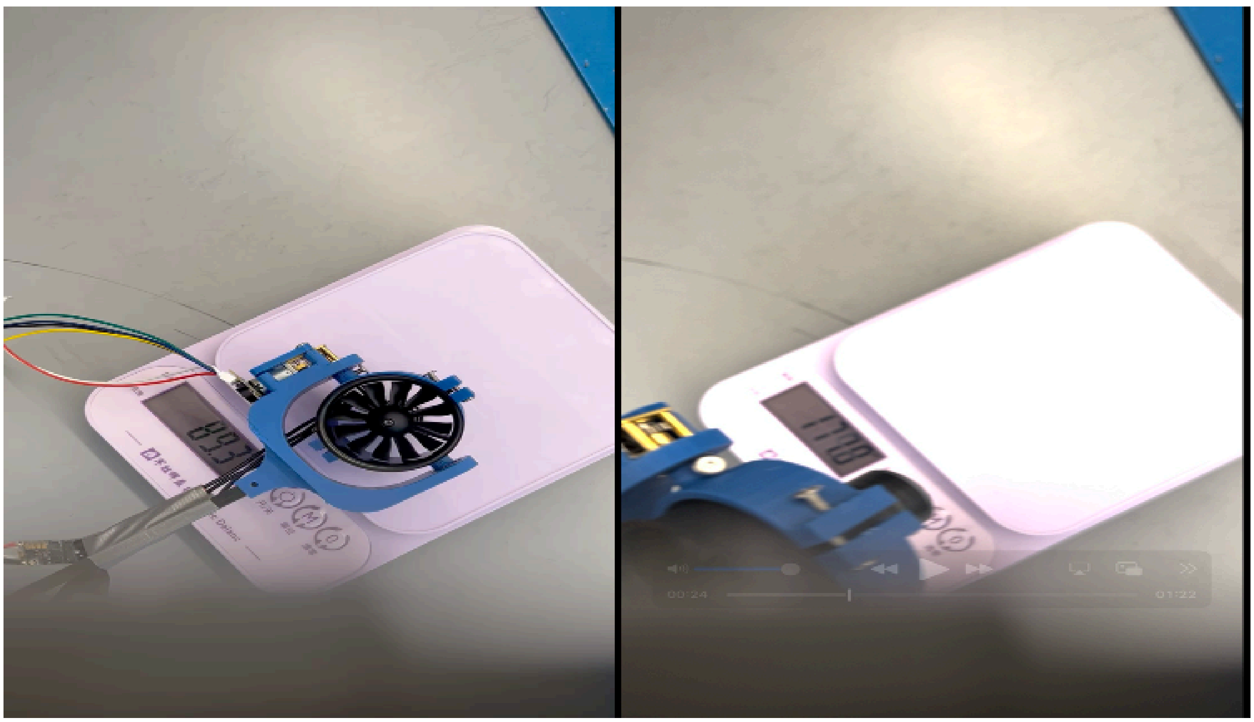


Figure 25: Diagram showing mass of propeller before and after in g.

Our tail is able to move left, right and even swing with the help of the fan motor.

FUTURE DIRECTIONS

We are currently setting up simulations for the MiniPupper 2 robot on Gazebo Simulator, a simulation software so as to understand how much force the MiniPupper can handle. This requires importing the step/stl files of the robot dog together with the urdf files of the robotic tail, and using physics of the tails and incorporating them to the robot dog. After that, we intend to develop our RL approach by training it from real-life cheetah videos and transferring the policy into simulations, then real-life robot dog. To do this, we require each servo joint and understand how each servo moves before we can transfer any policy into the robot. As it can be already inferred, the joints of a real-life cheetah have different limits to that of our legged robot. Because of this, we've currently been understanding pose estimates for our robot dog. Using a software called, DeepLabCut, we've been able to train the algorithm to our agility videos with the hope of understanding joints movements

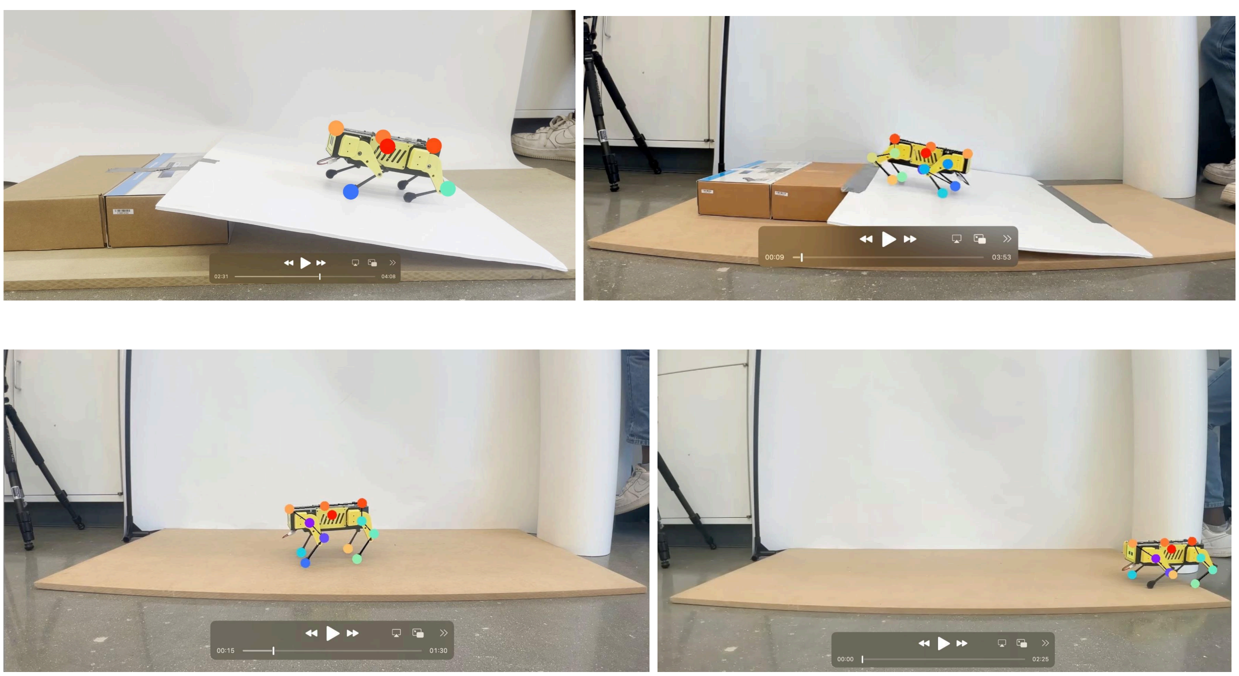


Figure 22: Labeled and trained videos with pose estimations

Further details for the package software and how to set it up are documented in the following link: https://deeplabcut.github.io/DeepLabCut/docs/installation.html

Further details for installing Nvidia Omniverse as well as Isaac Sim can be found at the following link: Omniverse Platform for OpenUSD Development ...NVIDIAhttps://www.nvidia.com › en-us › omniverse

This Discord channel has also been especially helpful for asking questions specifically directed at the developers of the mini pupper robot.

Supplementary materials: Google Drive

References.

[1] R. M. Alexander, Principles of Animal Locomotion, Princeton University Press, 2006.

[2] R. Full, T. Kubow, J. Schmitt, P. Holmes and D. Koditschek, “Quantifying Dynamic Stability and Maneuverability in Legged Locomotion,” Integrated and Comparative Biology, vol. 42, no. 1, pp. 149-157, 2002.

[3] Amir Patel and M. Braae,“Rapid Turning at High-Speed: Inspirations from the Cheetah’s Tail,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), November 3-7, 2013. Tokyo, Japan.

[4] Peter Eckert and Auke J. Ijspeert,”Benchmarking Agility For Multilegged Terrestrial Robots,” IEEE Transactions on Robotics, Vol. 35, No. 2, April 2019

[5] R. M. Alexander, “The gaits of bipedal and quadrupedal animals,” *Int. J. Robot. Res.*, vol. 3, no. 2, pp. 49–59, Jun. 1984. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/027836498400300205

[6] A. D. Kuo, “Choosing your steps carefully,” *IEEE Robot. Automat. Mag.*, vol. 14, no. 2, pp. 18–29, Jun. 2007.

[7] C. Nie, X. Pacheco Corcho, and M. Spenko, “Robots on the move: Ver- satility and complexity in mobile robot locomotion,” *IEEE Robot & Automation. Mag.*, vol.

20, no. 4, pp. 72–82, Dec. 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6582554/

[8] J. L. Pusey, J. M. Duperret, G. C. Haynes, R. Knopf, and D. E. Koditschek, “Free-standing leaping experiments with a power-autonomous elastic- spined quadruped,” *Proc. SPIE*, vol. 8741, 2013, Art. no. 87410W. [Online]. Available: https://doi.org/10.1117/12.2016073

[9] A. Patel and M. Braae, “An Actuated Tail Increases Rapid Acceleration Manoeuvres in Quadruped Robots,” in *International Joint Conferences on Computer, Information, Systems Sciences and Engineering*, 2012.