Department of Precision and Microsystems Engineering

Design and Development of a prototype for a Robot Monkey: Enabling Transition from Quadrupedal to Statically Stable Upright Standing

Tom Kuijlaars

Report no : MSD 2024.032
Coaches : Ir. P. Janssen & Ir. D. Syaifoel
Professor : Prof. dr. ir. J.L. Herder
Specialisation : Mechatronic System Design
Type of report : MSc Thesis
Date : 24 June 2024

TU Delft Delft University of Technology

Precision and Microsystems Engineering

Design and Development of a prototype for a Robot Monkey: Enabling Transition from Quadrupedal to Statically Stable Upright Standing

by

Tom Kuijlaars

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday June 24, 2024 at 11:00 AM.

Student number:	4557123	
Project duration:	May 1, 2023 – June 24, 2024	
Thesis committee:	Prof. dr. ir. J. Herder,	TU Delft, chair
	Dr. ir. S.H. Hossein Nia Kani,	TU Delft
	Ir. D. Syaifoel,	TU Delft, daily supervisor
	Ir. P. Janssen,	Avular, daily supervisor

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Acknowledgements

The exploration of mobile robotics represents not only the forefront of current technological advancements, but also deeply engaged my personal interest during my thesis project at Avular. I am grateful for the opportunity to contribute to this dynamic field, where innovation and practical applications appear limitless. Moreover, the supportive environment at Avular enabled me to explore innovative solutions and refine my skills in robotics engineering. From conceptualization to prototype testing, every phase of this project deepened my understanding of mobile robotics.

I am particularly grateful to Paul Janssen from Avular for his mentorship and guidance throughout this journey. He not only provided technical expertise but also encouraged me to explore new ideas and approaches. I also want to thank Alex Andriën for his assistance with the control aspects of my research. Furthermore, I would like to thank my supervisor Just Herder for giving me the chance to perform this research and providing me with valuable insights during our bi-weekly meetings. I want to thank Domas Syaifoel for his guidance and feedback during my thesis project.

Special thanks to my study buddies at P&E who adopted me in their study area. I also want to thank my roommates in Rotterdam for their help getting my mind of the study. Lastly, I want to thank my parents and my girlfriend for their support during my thesis and entire study time in Delft.

Tom Kuijlaars
Delft, June 2024

Contents

1 Introduction

Currently, there is significant interest in mobile robots. Legged robots offer significant advantages compared to conventional vehicles, as they enable locomotion in terrains that are inaccessible to wheeled and tracked vehicles [1] [2].

Existing quadruped robot such as SPOT [3], Mini Cheetah [4] and ANYmal [5] are very versatile. Mini Cheetah and Go2 [6] can do a somersault and the latter can stand on its two front legs. However, Go2 needs to constantly move to remain stable. Other quadruped robots are able to stand statically stable on their rear legs. In these cases though, either the robot is not completely stable during transition [7], or the robot uses very large feet which compromises the ability to maintain a dynamic quadruped gait [8].

The goal of this research is to design and develop a prototype quadruped robot capable of standing on its rear legs without compromising its ability to maintain a dynamic quadrupedal gait. This goal is divided into two sub goals: transitioning from quadrupedal standing to rear leg standing and remaining statically stable on rear legs. A tail is used so the robot can have a quasi-static transition and remain statically stable on its rear legs. Reflecting its core functionality, the prototype is named ”TT-Bot”, an acronym for Tail-assisted Transition Robot, highlighting its unique design and capabilities. TT-Bot is tested for stability, reach height and its capability to carry a payload. The research is done in cooperation with Avular [9].

This thesis report is structured as follows. First, in Section 2 the literature review that was executed is presented. Secondly, section 3 presents the research paper that was written for this research. Then, in Section 4 a final conclusion is given. Finally, the appendices at the end give the interested reader more detailed information about this research, **visual material** covering the different experiments is also available.

2

Literature Review

A Comparative Analysis of Quadruped Walking Robots: Gait, Mechanical Design and Actuation

Tom Kuijlaars
MSc Mechanical Engineering
Delft, The Netherlands

***Abstract*—This paper presents a review and comparative analysis of existing quadruped walking robots, focusing on their gait, mechanical designs and actuation methods. The review includes an in-depth examination of the different quadruped gaits, such as trot, pace, canter, bound and gallop. Various leg designs commonly used by quadruped robots are explored, such as prismatic, articulated and redundant articulated legs. Actuation methods, such as electrical, hydraulic and pneumatic, are discussed, highlighting their strengths and applications. Existing robots are classified into subcategories based on actuation and mechanical design. The study compares these subcategories based on speed, payload capacity and efficiency. Hydraulic actuated robots are linked to larger masses and higher payload capacities. Electrically actuated robots show better energy efficiency compared to hydraulically actuated ones. This analysis serves as a reference for researchers and engineers, guiding future developments in quadruped robot design and optimization.**

***Index Terms*—Quadruped robots, Gait, Leg design, Actuation, Speed, Payload, Efficiency**

I. Introduction

Currently, there is significant interest in mobile robots. These robots can be categorized into three different groups based on the way they move: legged robots, wheeled robots and tracked robots.[1] Legged robots offer significant advantages compared to conventional vehicles, as they enable locomotion in terrains that are inaccessible to wheeled and tracked vehicles. [2] [3].

Previous reviews on walking robots focused on heavy-duty legged robots [4], quadruped robot's legs [5], mammal-type quadrupeds [6], environment perception [1], anatomy of animals and robots [7] and development of quadruped robots [8].

This paper provides an analysis and comparison of existing quadruped walking robots focusing on their gait, mechanical design and actuation method. It classifies the robots in different categories based on their actuation method and makes subcategories based on mechanical design.

The paper is organized as follows. First, the different quadruped gaits are explained in section II. Second, multiple mechanical designs are elaborated upon in section III. After this the different actuation methods are discussed in section IV. In section V multiple existing quadruped are mentioned. A comparison of the different subcategories is done in section VI. The conclusions that are drawn are given in section VIII and a discussion in section VII.

II. Quadruped gait

In regular gaits, each foot is placed on the ground once per stride and consecutive strides are identical. The stride frequency is the number of strides per unit time. The stride length is the distance covered within a single stride. This distance is measured from one footstep to the next, made by the same foot.

The duty factor of a foot indicates the portion of time during which that foot remains in contact with the ground during one stride. Gaits can be classified on the basis of their duty factor. Gaits with duty factors greater than 0.5 are typically referred to as "walks". Gaits with duty factors below 0.5 are classified as "runs"[9].

The relative phase of a foot represents the timing of its placement within the stride, expressed as a fraction of the overall stride duration [9]. One foot is assigned as the first step, so this foot has a relative phase of zero.

Gaits can also be classified in symmetric and asymmetric gaits. A symmetric gait pattern refers to a gait where the left and right feet within each pair, so both front legs or both hind legs, have the same duty factors and a relative phase difference of 0.5 [10].

Within the group of walks there exist statically stable gaits. These gaits have a duty factor between 0.75 and 1. For these gaits there are always three feet on the ground. The center of gravity stays within the support polygon formed by it's feet, ensuring stability. These so called creeping gaits are particularly suitable for slow-speed movements.

Only three different creeping gaits exist where the placement of the feet ensures static stability at all times. One of these is the unique optimum gait that maximizes stability, the walking gait. [11]. This is the gait which is preferred by most natural quadrupeds.

Quadrupeds employ a variety of dynamically stable gaits, each characterized by distinct patterns of limb movement and coordination. These gaits include trot, pace, canter, bound and gallop. Each gait involves specific combinations of leg movements and timing, enabling quadrupeds to achieve different speeds and adapt to various locomotion requirements.

An overview of the different gait patterns can be seen in Figure 1 and they are explained further. The numbers represent phase differences relative to the front left leg.

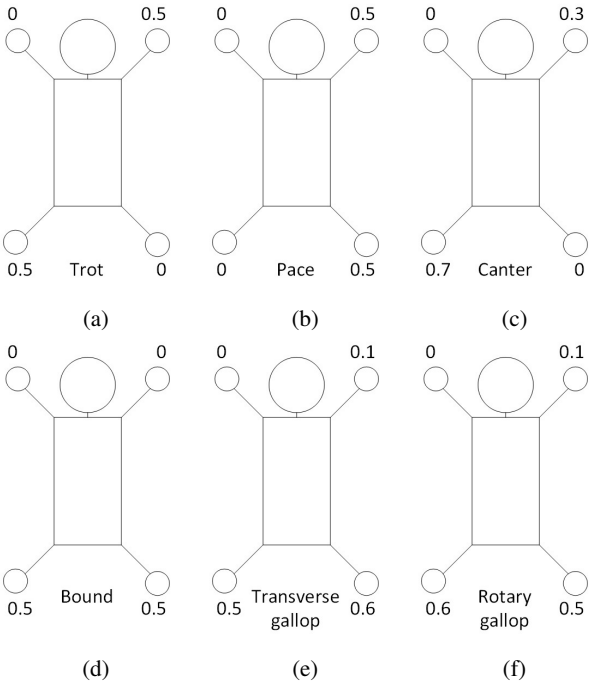


Fig. 1: Gait patterns of quadrupeds with phase difference [9]

A. Trot

The trot gait is one of the most common gaits observed in quadrupeds [10]. This gait is a symmetrical, two-beat diagonal gait. The legs on diagonally opposite sides move in pairs, so front left leg and hind right leg move simultaneously [12]. The gait pattern with its relative phase can be seen in Figure 1a. This gait is used by quadrupeds at medium speeds.

B. Pace

The pace gait is also a symmetrical, two-beat gait. However this gait is not diagonal. The legs on the same side move in pairs, so the front left and the hind left leg move simultaneously. The gait pattern with its relative phase can be seen in Figure 1b. Just as the trot gait, the pace gait is used at medium speeds.

C. Canter

The canter gait is an asymmetrical, three-beat gait. This particular gait is commonly seen when a horse transitions from trotting to galloping [13]. Therefore this gait is used between medium and high speeds. The gait pattern with its relative phase can be seen in Figure 1c.

D. Bound

The bound gait is a two-beat gait, the front and hind legs synchronize in pairs [14]. Although the gait looks symmetrical, it is paradoxically an asymmetric gait, because the relative phase of the pairs of legs is not 0.5. This can also be seen in Figure 1d. The bound gait is used by natural quadrupeds at high speeds.

E. Gallop

There are two different gallop gaits, transverse gallop (1e) and rotary gallop (1f). Both are four-beat and asymmetrical gaits. The difference between them can be seen in Figure 1. This gait is used by quadrupeds during high-speed locomotion.

III. Quadruped leg design

The mechanical design of a quadruped robot leg is crucial for achieving stable and efficient locomotion. There are three typical mechanical legs that are used by quadruped robots.

A. Prismatic leg

The prismatic leg is the most simple leg of the three. The two degrees-of-freedom (DOF) leg design replicates the spring-like movement found in animal legs using a straightforward structure [15]. It consists of a revolute joint and a prismatic joint that enable both rotational and linear motion. The topology of the prismatic leg can be seen in Figure 2.

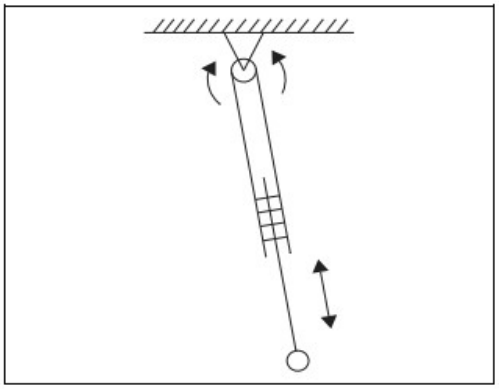


Fig. 2: Topology of prismatic leg [5]

B. Articulated leg

The difference between the prismatic and articulated leg is that articulated legs utilize a second revolute joint instead of a prismatic joint for controlling leg length. This type of leg design closely resembles the knee or elbow joint found in animal legs and exhibits excellent biomimetic properties. The topology of the articulated leg can be seen in Figure 3.

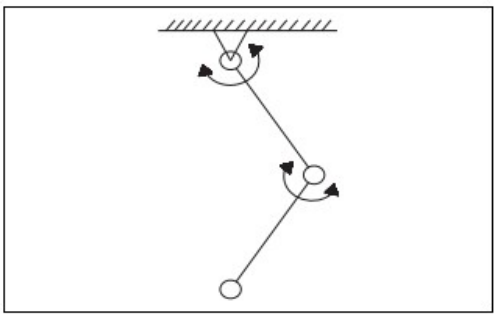


Fig. 3: Topology of articulated leg [5]

C. Redundant articulated leg

The last type of leg is the redundant articulated leg. This type is almost the same as the articulated leg. The difference is that this type has at least one more revolute joint. The most common type is the redundant articulated leg with three revolute joints. The topology of this redundant articulated leg can be seen in Figure 4.

This leg design closely resembles the legs commonly found in nature. The third revolute joint imitates the ankle joint of animals.

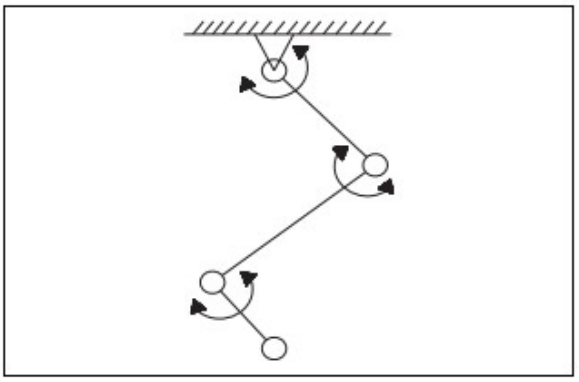
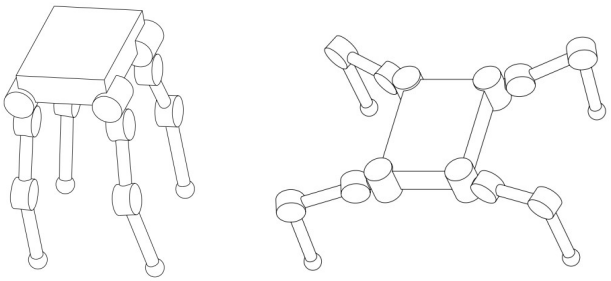


Fig. 4: Topology of redundant articulated leg[5]

D. Joint configuration

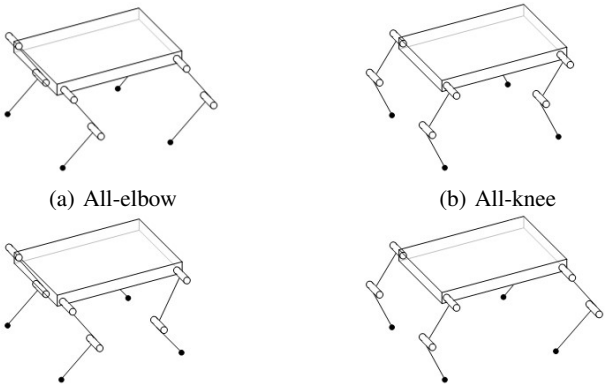
There are two types of (redundant) articulated legs, the mammal-type and the sprawling-type. The distinction lies in the positioning of the first leg segment, which can either be vertical or horizontal, corresponding to the mammal-type and sprawling-type configurations, respectively. These two different configurations can be seen in Figure 5.



(a) Mammal-type robot (b) Sprawling-type robot

Fig. 5: Configurations of legs [16]

The arrangement of leg joints in a quadruped robot plays a significant role in determining its performance in terms of kinematics and dynamics. There are four different joint configurations possible for the mammal-type quadruped robots which can be seen in Figure 6: all-elbow (6a), all-knee (6b), front elbow and back knee (6c), front knee and back elbow (6d) [17].



(a) All-elbow (b) All-knee

(c) Inward-pointing (d) Outward-pointing

Fig. 6: Configurations of mammal-type legs (forward direction is left)

IV. Actuation

Actuators play a crucial role in a quadruped robot by supplying it with the necessary power, torque and control for locomotion and determining its overall performance. The actuation of quadruped robots involves coordinating the movement of the joints in a synchronized manner. There are multiple power sources that can be used for the actuation of the joints; electrical, hydraulic and pneumatic.

A. Electrical

Electrical actuated quadrupeds can be divided into four groups with respect to their driving mode: series elastic drive, motor integrated gearbox drive, motor direct drive and servo motors.

Electric motors, due to their inherent limitations, often transmit relatively low torque considering their size and weight. To compensate for this limitation, high reduction ratio reducers are commonly used, however this limits the maximum joint speed. These systems encounter issues when moving on uneven terrain, as they tend to generate instantaneously high torque peaks that can lead to gear failure [17]. Series elasticity is introduced to greatly reduce these peak forces from happening. This is done by adding a spring to the system. A schematic overview can be seen in Figure 7. This turns the force control problem into a position control problem and therefore it significantly improves accuracy, since position is easier to control accurately than force. Another benefit of series elastic drive is that is has the possibility of energy storage [18], potentially making legged locomotion more efficient.

A motor integrated gearbox drive is utilized to amplify the torque produced by the motor in order to meet the motion requirements of the quadruped robot. This combination ensures satisfactory control accuracy and allows for a compact size of the robot's overall structure [17]. A gearbox ensures a high dynamic load capacity and stiffness, the latter is not necessarily desired.

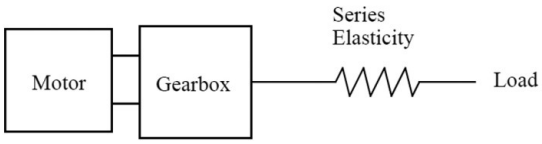


Fig. 7: Schematic overview of series-elastic actuator [18]

A motor direct-drive system in a quadruped robot does not use gears, belts, chains or other reducers to amplify motor torque. Instead it relies on a well-designed structure and carefully selected motor to achieve motion. This approach offers several advantages, including improved mechanical stability, enhanced mechanical efficiency and a more compact overall structure, as the gearbox is no longer required [17].

Quadruped robots designed for desktop applications and laboratory use typically rely on servo motors as their driving mechanism. The motors are compact in size, but possess the ability to generate significant torque. This system offers the advantage of adjustable power output, allowing for efficient energy consumption based on the load requirements at any given moment. This adaptability results in lower energy consumption when dealing with lighter loads [17].

B. Hydraulic

Hydraulic actuation offers faster response, higher output power, greater power density, and wider bandwidth compared to electrical actuation with respect to size and weight [19]. The power is supplied by a high-pressure fluid pump. The actuators are frequently found in the form of linear cylinders, rotary vane actuators and hydraulic motors [20]. It uses the pressure of the fluid on a surface to move it. These actuators are controlled using either a solenoid valve for on/off control or a servo-valve for proportional control. The solenoid valve or servo valve is operated electrically by a low-power electronic control circuit.

C. Pneumatic

The pneumatic actuators that are currently used in robotics are linear pneumatic cylinders and pneumatic artificial muscles (PAMs). Pneumatic cylinders work on the same principle as hydraulic cylinders, however they use a gas instead of a liquid.

PAMs are engines that use gas pressure to generate linear motion. They function by employing a flexible reinforced closed membrane, connected at both ends to fittings that transfer mechanical power to a load. When the membrane is inflated or deflated, it expands or contracts radially, respectively. Simultaneously, it contracts axially, resulting in a pulling force on the load. The resulting force and motion produced by this type of actuator are both linear and unidirectional. To achieve bi-directional motion, two actuators need to be coupled. One actuator is responsible for moving the load in one direction, while the other actuator acts as a brake to stop the load at its desired position. To move the load in the opposite direction, the function of the actuators change. This arrangement is commonly known as an antagonistic set-up. This antagonistic coupling can be used for both linear and rotational motion, as illustrated in Figure 8.

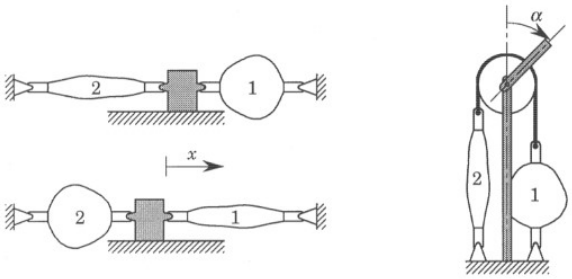


Fig. 8: Antagonistic set-up of PAMs[21]

V. State of the Art

In this section multiple existing quadruped walking robots will be discussed. Their gait, mechanical design and actuation is being looked at and some specifications are given. At first the electric robots are described, followed by both the hydraulically driven robots and the pneumatically actuated robots.

A. Electric robots

The Tokyo Institute of Technology have developed multiple quadruped walking robots better known as their TITAN-series. One of them, TITAN-XIII (Figure 9), is a sprawling-type robot and weighs around 5.65 kg. It has 3 DOFs per leg and it uses customized brushless DC motors as actuators for every joint. It consumes 135 W at a walking speed of 1.38 m/s [16].

Fig. 9: Quadruped robot TITAN-XIII [16]

The McGill University in Canada developed ScoutII. The robot consists of four compliant prismatic legs [22]. Only the hip joint of each leg is actuated by a series-elastic actuator. This is done with a brushed DC motor, gearbox and a belt and pulley pair. This enhances the torque and minimizes the impact forces experienced by the motor shaft, as explained before. It consumes between 430 and 440 W at a bounding

speed of 1.3 m/s. While galloping it can reach a speed of 1.4 m/s, however the power consumption at this gait is 515 W.

Zurich Federal Institute of Technology developed Star1ETH. Each leg has 3 DOFs and the joints are driven by 3 series elastic actuators [23]. This is done with brushless DC motors and harmonic drive gearbox. This is connected to a linear compression spring using a chain and a cable, thus reducing the impact loss and minimizing the inertia of the thighs and legs. The main configuration of the legs is all-inward, but it can be operated with different leg configurations. It can trot at a speed of 0.7 m/s and the robot requires in average less than 230W [24].

Stanford Doggo (Figure 10) is a quasi-direct-drive (QDD) quadruped robot developed by Stanford University. When using QDD the mass is increased. Therefore it is only beneficial if the output torque is increased high enough. Otherwise a larger (and heavier) motor will suffice. Each leg of Stanford Doggo has 2 DOFs and it can move forward in different gaits, such as walking, trotting and bounding. It has a total mass of 4.8 kg and can achieve a speed of 0.9 m/s [25].

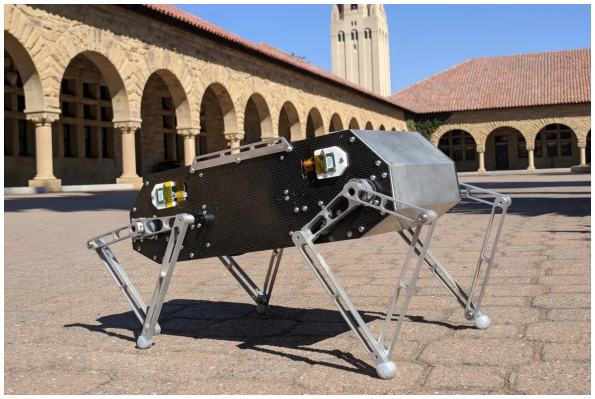


Fig. 10: Quadruped robot Stanford Doggo [25]

ANYmal from ETH Zürich is developed to operate on rough terrain. Hebei University of Technology has developed a flexible joint with series elasticity that is specifically designed for the leg configuration [17]. The joints are arranged in an offset manner, allowing for full rotation of all joints. This enables a wide range of motion [5]. It can walk, crawl, trot and even climb stairs at a 50 degree angle [26]. At standstill it's power consumption is 100 W and it is 290 W while trotting at a speed of 0.8 m/s [27]. It can reach a maximum speed of 1.3 m/s and has a weight of 50 kg [28].

KOLT is an electrically driven quadruped robot that also uses pneumatic springs. All the joints are driven by brushless motors. The knee electro-pneumatic actuation system has been developed with the goal of maximizing thrust and energy efficiency. In this system, the knee joint is driven by a brushless motor that generates leg flexion, while a pneumatic spring generates leg extension. The knee motor drives the knee joint using a cable mechanism and a system of pulleys. Meanwhile, the pneumatic spring is responsible for providing leg compliance during landing and allows for the storage of elastic energy. KOLT is able to trot at a speed of 1.1 m/s and does this with a power consumption of 2084 W [29].

HuboDog is a quadruped robot that weighs 42 kg including battery. Each leg has 3 DOFs and it uses brushless DC motors with harmonic drive as actuators. The estimation of trotting speed is 1.24 m/s. It can walk at a speed of 0.88 m/s and at 0.55 m/s with a payload of 24 kg [30].

The MIT Cheetah 1 (Figure 11) is a quadruped robot developed by MIT. Each leg of the MIT cheetah consists of three links. Through a pantographic leg design, the motions of the first and last links from the shoulder are kinematically connected in parallel to each other. As a result, this design creates 2 DOFs for these three links. Additionally a third DOF of the leg a-a is controlled by a low-power servo motor. The 2 DOFs links are actuated by the dual co-axial motor module, which contains high-torque electromagnetic motors [31]. The robot has a mass of 33 kg and has a power consumption of 377W at a speed of 2.3 m/s [32]. It can reach a maximum speed of 6 m/s with a trotting gait.

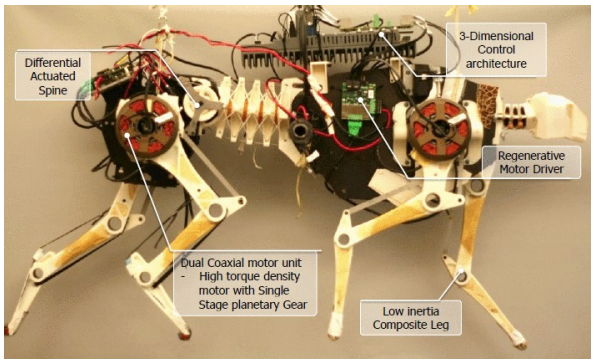


Fig. 11: Quadruped robot MIT Cheetah 1[32]

MIT developed a new quadruped robot called Cheetah 2. Just like MIT Cheetah 1, it uses the pantographic leg design, so each leg has 3 DOFs [33]. Instead of trotting it uses a bounding gait with which it reaches a maximum speed of 6.4 m/s [34].

MIT also developed the Cheetah 3. This quadruped robot only has 2 links for each leg, but each leg also has 3 DOFs. Each joint uses nearly identical actuators. These consist of a custom high torque density electric motor coupled to a single-stage planetary gear reduction [35]. It's range of motion is improved with respect to MIT Cheetah 2. The total weight of the robot is 45 kg and it uses a trotting gait.

Go2 is an electrically actuated quadruped robot developed by Unitree. It weighs around 22 kg and has 3 DOFs for each leg. It can reach a speed of 2.5 m/s [36].

AiDIN-VI (Figure 12) is a quadruped that uses modular actuator units, allowing for interactive torque/force control in different environments. Each leg has 3 DOFs and they are placed in an all-elbow configuration. AIDIN-VI has a total weight of 43.7 kg including battery. It is able to perform a walking, pacing and trotting gait with a maximum speed of 1.2 m/s [37].

Fig. 12: Quadruped robot AiDIN-VI [37]

Warp1 is developed by the Royal Institute of Technology in Stockholm. The robot has an all-knee configuration and each leg has 3 DOFs. The robot weighs around 60 kg and is able to walk at a speed of 1.2 m/s. All the joints are actuated with a DC motor via a harmonic drive connected to a cable and pulley system [38].

One of the most well-known quadruped robots is SPOT, developed by Boston Dynamics. It is able to perform omni-directional walking and trotting gaits [8]. SPOT has a total weight of around 30 kg, is able to reach a speed up to 1.6 m/s and has a payload of 14 kg [39]. It is electrically actuated and powered by an onboard battery. Each leg has 3 DOFs and the legs are pointed in an all-elbow configuration.

B. Hydraulic robots

During the 1960's, the General Electric Company in the United States created the walking truck (Figure 13), which became the first hydraulic-drive quadruped robot [17]. The robot has 3 DOFs in each leg with an all-knee configuration. It has a mass of around 1300 kg and it can walk at a speed of 2.2 m/s. It is controlled by a human operator's arms and legs [4].

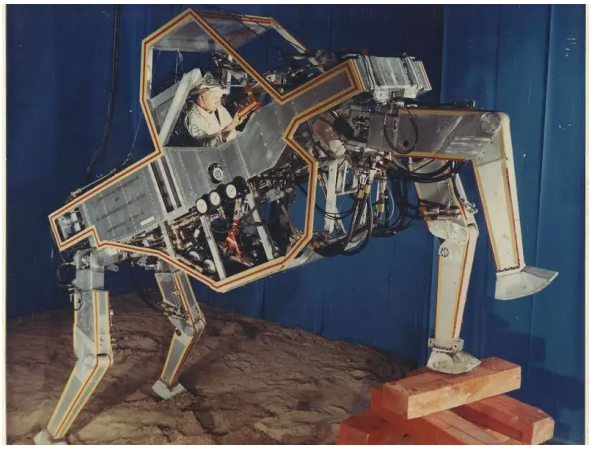


Fig. 13: Quadruped robot GE Walking Truck [40]

Boston Dynamics developed the hydraulically driven quadruped BigDog. It is powered by an internal combustion engine that delivers about 11 kW. This engine drives a hydraulic pump. Each leg has 4 DOFs, driven by 4 hydraulic actuators. The legs are in an inward-pointing configuration. Bigdog has a weight of approximately 109 kg and is able to perform multiple gaits: walking at 0.2 m/s, trotting at 2 m/s and bounding at 3.1 m/s [41].

Raibert M. developed a quadruped robot, named Raibert's Quadruped for convenience, that can trot and bound. It uses prismatic legs with each 3 DOFs. The legs are actuated with hydraulic cylinders. The robot has a mass of 38 kg, it can trot with 2.2 m/s and bound with 2.9 m/s [14].

HyQ has been developed by the Italian Institute of Technology (IIT) to serve as a platform to study highly dynamic motions such as running and jumping and careful navigation on rough terrain [42]. Each leg has 3 DOFs, which are partially hydraulically and electrically driven. An additional DOF is added with an integrated passive prismatic joint in the lowest leg segment. The hip flexion-extension (f-e) and knee f-e joints are driven by the hydraulic actuators. These joints require high velocity, high power-to-weight ratio and robustness against torque peaks. The hip abduction-adduction (a-a) joint is driven by a brushless DC motor with harmonic drive, because the compactness of this joint is more important. [43]. The total weight of HyQ is around 90 kg, it has the inward-pointing configurations of its legs, with which it can trot at a speed of 2 m/s.

HyQ2Max (Figure 14) is an enhanced version of the agile and versatile robot HyQ. The upgraded robot has increased durability, greater strength and an added ability to self-right, expanding its existing locomotion capabilities. It still has the same number of DOFs, the main difference lies in the actuators. It consists only of hydraulic actuators. The actuators for the hip a-a, hip f-e and knee f-e are a double-vane rotary, a single-vane rotary and an asymmetric hydraulic cylinder. The robot weighs 80 kg without onboard power supply. Simulations were run for 0 kg payload and for 40 kg payload, which represents a future extension with onboard power supply and a payload[44].

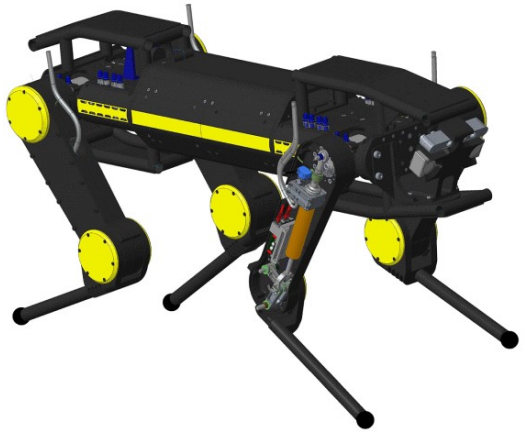


Fig. 14: Quadruped robot HyQ2MAX [44]

Most hydraulically actuated quadruped robots are heavy. The IIT developed a lightweight hydraulic quadruped robot MiniHyQ, based on the previously developed robot HyQ. MiniHyQ is somewhat smaller than it's predecessor, but above all it has a much lower weight of 35 kg. Each leg has 3 DOFs and these joints are fully hydraulically actuated by 2 linear and 1 rotary actuator [45]. The on-board power pack is specially developed for this robot [46].

Scalf is a hydraulically actuated quadruped robot. Each leg consists of three rotary joints. These are all actuated by linear hydraulic servo cylinders. The legs are placed in an all-inward configuration. It has a weight of 65 kg without power pack. It can reach a speed of 1.8 m/s with a trot gait [47].

C. Pneumatic robots

VU Quadruped (Figure 15) is a sprawling-type quadruped robot based on the typical structure of an insect. Each leg has 3 DOFs and the joints are actuated using pneumatic servoactuators. These servoactuators comprise of a custom pneumatic servovalve attached to a pneumatic cylinder. The robot has a mass of 6.9 kg without onboard power source and controller and it can trot at a maximum speed of 0,46 m/s [48].

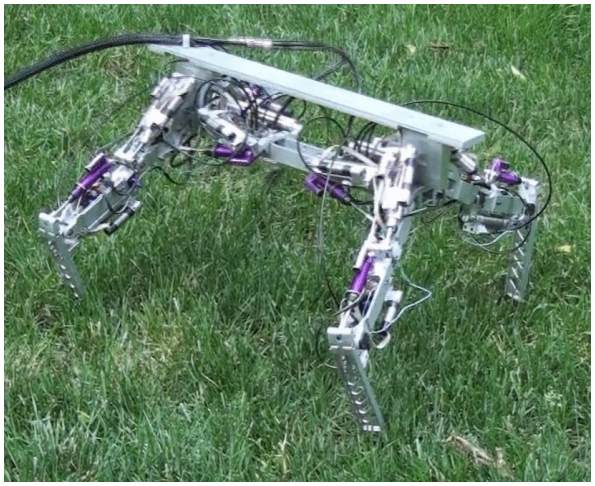


Fig. 15: Quadruped robot VU Quadruped [48]

Puppy (Figure 16) is a quadruped robot driven by PAMs. It's design is based on a full scale adult greyhound. It has redundant articulated legs with each 3 DOFs. An additional passive DOF is provided by it's flexible feet. Each joint has an antagonistic pair of PAMs made by FESTO, so it uses 24 in total [49]. The robot weighs 6.8 kg without power supply and it can walk at a speed of 1 m/s [50].

The Cheetah Robot is developed based on the dimensions of a cheetah. Each leg consists of four joints, three active joints and one passive compliant joint with a torsional spring. The three active joints are actuated with seven PAMs for each leg. Five of them act over single joints. Two of them act as biarticular muscles, which are muscles that cross two joints instead of one [51]. Just like the Puppy robot these PAMs

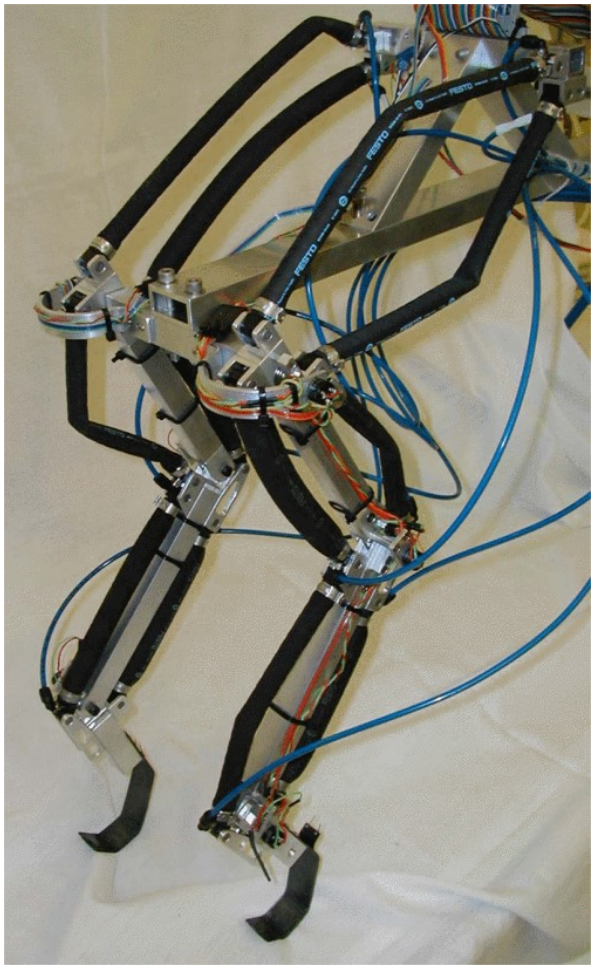


Fig. 16: Quadruped robot Puppy [49]

are made by FESTO. To achieve optimal performance, the actuators must possess sufficient power. Two PAMs are used in parallel as if it is one muscle to meet the joint requirements. This is done for the hip flexor, knee flexor, knee extensor and ankle extensor [52]. The total mass of the Cheetah Robot is 70 kg. In the simulations it can bound with a speed of 2.7 m/s.

Another quadruped robot driven by PAMs is developed by Fukuoka Y. et al. This robot is called PMA Quadruped for convenience. Each redundant articulated leg consists of 3 DOFs and uses 6 PAMs to rotate the joints [53]. The total mass of the robot is 7.4 kg, it is able to pace at a speed of 2.5 m/s [54].

VI. Comparison of robots

The robots that are described in section V are categorized on their actuation method. Within these three groups subcategories are made for robots with similar designs, which can be seen in Table I. For the electric actuation there are four different subcategories, for both hydraulic and pneumatic there are two subcategories.

The first electric subcategory is the mammal-type articulated leg, since all these robots have articulated legs with 3 DOFs, namely hip a-a, hip f-e and knee f-e. KOLT uses a pneumatic spring for leg extension, yet it still classifies as electric. The second electric subcategory is sprawling-type. The third electric subcategory is mammal-type pantographic leg. In this group belong the MIT Cheetah 1 and 2. They are similar to the first subcategory, however as said before they use three links for each leg, using a pantographic design. The robots in the last subcategory cannot be specified. Stanford Doggo and Scout II both have 2 DOFs for each leg, but their designs differ.

The first hydraulic subcategory is the mammal-type. All the robots in this category have 3 DOFs for each leg, except for one. BigDog has got 4 DOFs, since it has redundant articulated legs. The other robots cannot be specified. GE Walking truck has a human operator and Raibert's Quadruped uses prismatic legs.

The first pneumatic subcategory is PAMs, all these robots use pneumatic artificial muscles. The second pneumatic subcategory is cylinder. This robot uses pneumatic cylinders for actuation.

TABLE I: Subcategories of Robots

Electric			
Mammal-type articulated leg	Sprawling-type	Mammal-type pantographic leg	Other
AiDIN-VI ANYmal Go2 Hubodog KOLT MIT Cheetah 3 SPOT StarIETH Warp1	TITAN XIII	MIT Cheetah 1 MIT Cheetah 2	Stanford Doggo Scout II
Hydraulic		**Pneumatic**	
Mammal-type	Other	PAM's	Cylinder
BigDog HyQ HyQ2Max MiniHyQ Scalf	GE Walking truck Raibert's Quadruped	Cheetah Robot PMA Quadruped Puppy	VU quadruped

Quadruped walking robots can be compared on multiple criteria, such as speed, payload and efficiency. All the robots are compared on these three criteria, a complete overview can be seen in the table in Appendix A. Below, the various criteria are elaborated upon and the subsequent tables present the results. The values of the various criteria in the subcategories are based on the average scoring of the robots in that subcategory.

A. Speed

As described in section II different gait patterns are used at different speeds. The Froude number (Fr) is a dimensionless speed parameter and is written as:

$$Fr = v^2/gL \tag{1}$$

where v, g an L are the speed, gravitational constant and robot's leg length, respectively.

The choice of walking gaits is preferable for low Froude numbers, while running gaits are more suitable for high Froude numbers [55].

The speed of a robot can also be expressed in normalized speed (NS) [8], which takes the length of the robot into account.

$$NS = v/BL \tag{2}$$

where, v and BL are the speed and body length of the robot, respectively.

The different speed parameters of the subcategories can be seen in Table II. From this table it can be seen that a high NS corresponds to a high Fr. The exception is the sprawling-type. The MIT robots where specially developed to reach high speeds, as can be seen. There seems to be no direct correlation between actuation method and normalized speed.

TABLE II: Speed of Robots

Actuation	Subcategory	Speed (m/s)	Fr (-)	NS (s^{-1})
Electric	Mammal-type (art. leg)	1.31	0.42	1.65
	Sprawling-type	1.38	0.65	6.47
	Mammal-type (pan. leg)	6.2	7.22	7.57
	Stanford Doggo	0.9	0.59	2.14
	ScoutII	1.3	0.53	1.55
Hydraulic	Mammal-type	2.04	0.65	1.95
	GE Walking truck	2.2	0.2	0.55
	Raibert's Quadruped	2.9	1.53	2.76
Pneumatic	PAM's	1.48	0.5	2.2
	Cylinder	0.46	0.06	1

B. Payload

The payload (PL) of a quadruped walking robot refers to the maximum weight it can carry while remaining stable and functional. This varies a lot between small and large robots. To make a good comparison between small and large robots the payload capacity (PLC) is used. This is their payload with respect to their own weight:

$$PLC = PL/m \tag{3}$$

where PL and m are the payload and mass of the robot, respectively.

The different payloads of the subcategories can be seen in Table III. The payload is not known for all the robots, but it is also interesting to look at the masses. Generally robots with a large mass use hydraulic actuators and they have a high payload. On the other side, robots with small masses use pneumatic actuators. Electric actuators are used in a wide range of masses.

TABLE III: Payload of Robots

Actuation	Subcategory	Mass (kg)	PL (kg)	PLC (%)
Electric	Mammal-type (art. leg)	44.74	19.7	56.4
	Sprawling-type	5.65	5	88.5
	Mammal-type (pan. leg)	33	-	-
	Stanford Doggo	4.8	-	-
	ScoutII	24.8	-	-
Hydraulic	Mammal-type	76	60	76.3
	GE Walking Truck	1300	70	5.4
	Raibert's Quadruped	38	-	-
Pneumatic	PAM's	27.93	24.4	381.3
	Cylinder	6.9	-	-

C. *Efficiency*

Energy efficiency is a critical factor to minimize in dynamic locomotion. Gait pattern, stride and duty factor are closely related to energy efficiency. To evaluate the energy efficiency of animals and legged robots the total cost of transport (COT) is used in general and is written as:

$$COT = P/mgv \tag{4}$$

where P, m, g and v are the robot's power, mass, gravitational constant and speed, respectively [56]. The lower the COT, the more efficient a robot is.

The COT for the different subcategories can be seen in Table IV. For pneumatically actuated robots the COT is not known. There is a correlation between actuation and COT. Hydraulically actuated robots are way less efficient then electrically actuated robots. The MIT robots are also designed to be highly efficient, this can be clearly seen in the table.

TABLE IV: Cost of Transport of Robots

Actuation	Subcategory	Mass (kg)	Speed (m/s)	COT (-)
Electric	Mammal-type (art. leg)	44.74	1.31	1.45
	Sprawling-type	5.65	1.38	1.76
	Mammal-type (pan. leg)	33	6.2	0.49
	Stanford Doggo	4.8	0.9	3.2
	ScoutII	24.8	1.3	1.38
Hydraulic	Mammal-type	76	2.04	7.89
	GE Walking Truck	1300	2.2	-
	Raibert's Quadruped	38	2.9	-
Pneumatic	PAM's	27.93	1.48	-
	Cylinder	6.9	0.46	-

VII. Discussion

In various papers, the masses of the robots differ even when referring to the same robot, which may introduce discrepancies when compared to other studies. Additionally, variations in mass occur when considering the presence or absence of an onboard power supply; such data is presented in detail in Appendix A.

The COT remains undisclosed for several robots within the existing literature. Where feasible, I have computed the COT using available data. Consequently, the COT values for certain robots may be subject to potential inaccuracies. This applies similarly to the Fr for specific robots, for which I have also conducted my own calculations. In AppendixA it is stated which robots are involved.

VIII. Conclusion

In conclusion, quadruped walking robots come in various designs with different leg types, joint configurations and actuation methods. Electrically actuated robots generally achieve higher speeds and efficiency compared to hydraulically actuated ones. In general robots with large masses are hydraulically actuated. There are still very few robots that are pneumatically actuated. Further research on pneumatically actuated walking robots is essential to generate additional reference materials for a comprehensive comparison with electrically and hydraulically actuated walking robots.

References

[1] Meng Xiangrui et al. *A Review of Quadruped Robots and Environment Perception*. 2016.

[2] Manuel Fernando Silva and J. A.Tenreiro MacHado. *A literature review on the optimization of legged robots*. Oct. 2012, pp. 1753–1767. DOI: 10.1177/1077546311403180.

[3] G. Satheesh Kumar et al. "Literature Survey on Four-Legged Robots". In: Springer Science and Business Media Deutschland GmbH, 2021, pp. 691–702. ISBN: 9789811544873. DOI: 10.1007/978-981-15-4488-0_58.

[4] Hongchao Zhuang et al. *A review of heavy-duty legged robots*. 2014, pp. 298–314. DOI: 10.1007/s11431-013-5443-7.

[5] Yuhai Zhong et al. "Analysis and research of quadruped robot's legs: A comprehensive review". In: *International Journal of Advanced Robotic Systems* 16 (3 Mar. 2019). ISSN: 17298814. DOI: 10.1177/1729881419844148.

[6] Yibin Li et al. *Research of Mammal Bionic Quadruped Robots: a Review*. 2011, pp. 166–171. ISBN: 9781612842516.

[7] Akira Fukuhara, Megu Gunji, and Yoichi Masuda. "Comparative anatomy of quadruped robots and animals: a review". In: *Advanced Robotics* 36 (13 2022), pp. 612–630. ISSN: 15685535. DOI: 10.1080/01691864.2022.2086018.

[8] Priyaranjan Biswal and Prases K. Mohanty. "Development of quadruped walking robots: A review". In: *Ain Shams Engineering Journal* 12 (2 June 2021), pp. 2017–2031. ISSN: 20904479. DOI: 10.1016/j.asej.2020.11.005.

[9] R. McN. Alexander. *Locomotion of Animals*. 1982.

[10] R. McN. Alexander. "The Gaits of Bipedal and Quadrupedal Animals". In: *The International Journal of Robotics Research* 3 (1984), pp. 49–59. DOI: 10.1177/027836498400300205.

[11] R B Mcghee and A A Fkask. *On the Stability Properties of Quadruped Creeping Gaits**. 1968.

[12] Danpu Zhao et al. "Gait definition and successive gait-transition method based on energy consumption for a quadruped". In: *Chinese Journal of Mechanical Engineering (English Edition)* 25 (1 Jan. 2012), pp. 29–37. ISSN: 10009345. DOI: 10.3901/CJME.2012.01.029.

[13] R. McN. Alexander. *Principles of Animal Locomotion*. Princeton University Press, 2003.

[14] Marc H Raibert. "TROTTING, PACING AND BOUNDING BY A QUADRUPED ROBOT". In: *Journal of Biomechanics* 23 Suppl 1 (1990), pp. 79–98. DOI: 10.1016/0021-9290(90)90043-3.

[15] Marc H. Raibert et al. *Dynamically Stable Legged Locomotion*. Massachusetts Institute of Technology, Sept. 1989.

[16] Satoshi Kitano et al. "TITAN-XIII: sprawling-type quadruped robot with ability of fast and energy-efficient walking". In: *ROBOMECH Journal* 3 (1 Dec. 2016). ISSN: 21974225. DOI: 10.1186/s40648-016-0047-1.

[17] Ligang Yao, Hao Yu, and Zongxing Lu. "Design and driving model for the quadruped robot: An elucidating draft". In: *Advances in Mechanical Engineering* 13 (4 2021). ISSN: 16878140. DOI: 10.1177/16878140211009035.

[18] Gill A Pratt and Matthew M Williamson. *Series Elastic Actuators*. 1995, pp. 399–406.

[19] Daegyeong Kim et al. *Principal properties and experiments of hydraulic actuator for robot*. 2014, pp. 458–460.

[20] Victor Scheinman and J Michael Mccarthy. “Mechanisms and Actuation”. In: *Springer Handbook of Robotics*. 2008, pp. 67–86.

[21] Frank Daerden and Dirk Lefeber. *Pneumatic Artificial Muscles: actuators for robotics and automation*. 2002.

[22] Ioannis Poulakakis, James Andrew Smith, and Martin Buehler. “Modeling and experiments of untethered quadrupedal running with a bounding gait: The scout II robot”. In: *International Journal of Robotics Research* 24 (4 Apr. 2005), pp. 239–256. ISSN: 02783649. DOI: 10.1177/0278364904050917.

[23] Marco Hutter et al. “Starleth: A compliant quadrupedal robot for fast, efficient, and versatile locomotion”. In: *Adaptive Mobile Robotics - Proceedings of the 15th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, CLAWAR 2012* (2012), pp. 483–490. DOI: 10.1142/9789814415958_0062.

[24] Marco Hutter et al. “Toward combining speed, efficiency, versatility, and robustness in an autonomous quadruped”. In: *IEEE Transactions on Robotics* 30 (6 Dec. 2014), pp. 1427–1440. ISSN: 15523098. DOI: 10.1109/TRO.2014.2360493.

[25] Nathan Kau et al. *Stanford Doggo An Open-Source, Quasi-Direct-Drive Quadruped*. 2019.

[26] Marco Hutter et al. “ANYmal - A Highly Mobile and Dynamic Quadrupedal Robot”. In: (2016).

[27] M. Hutter et al. “ANYmal - toward legged robots for harsh environments”. In: *Advanced Robotics* 31 (17 Sept. 2017), pp. 918–931. ISSN: 15685535. DOI: 10.1080/01691864.2017.1378591.

[28] ANYBotics. *ANYmal Technical Specifications*. 2022. URL: https://www.anybotics.com/anymal-specifications-sheet/.

[29] Joaquin Estremera and Kenneth J. Waldron. “Thrust control, stabilization and energetics of a quadruped running robot”. In: *International Journal of Robotics Research* 27 (10 Oct. 2008), pp. 1135–1151. ISSN: 02783649. DOI: 10.1177/0278364908097063.

[30] Jae Wook Chung, Ill Woo Park, and Jun Ho Oh. “On the design and development of a quadruped robot platform”. In: *Advanced Robotics* 24 (1-2 Jan. 2010), pp. 277–298. ISSN: 01691864. DOI: 10.1163/016918609X12586214966992.

[31] Hae-Won Park and Sangbae Kim. *The MIT Cheetah, an Electrically-Powered Quadrupedal Robot for High-speed Running*. 2014, p. 323328.

[32] Sangok Seok et al. *Design Principles for Highly Efficient Quadrupeds and Implementationon the MIT Cheetah Robot*. 2013, pp. 3307–3312. ISBN: 9781467356435.

[33] Hae-Won Park, Sangin Park, and Sangbae Kim. “Variable-speed Quadrupedal Bounding Using Impulse Planning: Untethered High-speed 3D Running of MIT Cheetah 2”. In: (2015).

[34] Hae Won Park, Patrick M. Wensing, and Sangbae Kim. “High-speed bounding with the MIT Cheetah 2: Control design and experiments”. In: *International Journal of Robotics Research* 36 (2 Feb. 2017), pp. 167–192. ISSN: 17413176. DOI: 10.1177/0278364917694244.

[35] Gerardo Bledt et al. “MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot”. In: (2018).

[36] Xingxing Wang. *Go2, Unitree Robotics*. 2023. URL: https://m.unitree.com/en/go2/.

[37] Yoon Haeng Lee et al. “Development of a Quadruped Robot System with Torque-Controllable Modular Actuator Unit”. In: *IEEE Transactions on Industrial Electronics* 68 (8 Aug. 2021), pp. 7263–7273. ISSN: 15579948. DOI: 10.1109/TIE.2020.3007084.

[38] Christian Ridderström et al. *The basic design of the quadruped robot Warp1*. 2000. URL: https://www.researchgate.net/publication/255585366.

[39] Boston Dynamics. *ABOUT SPOT*. 2023. URL: https://dev.bostondynamics.com/docs/concepts/about_spot#about-spot.

[40] Tomas Kellner and Mike Keller. *AT-AT Boy! GE’s Walking Truck From The 1960s Mixed ’Star Wars’ With Jules Verne*. May 2020.

[41] Marc Raibert. “BigDog, the rough-terrain quadruped robot”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* 17 (1 PART 1 2008). ISSN: 14746670. DOI: 10.3182/20080706-5-KR-1001.4278.

[42] Claudio Semini. *HyQ - Design and Development of a Hydraulically Actuated Quadruped Robot*. 2010.

[43] C. Semini et al. “Design of HyQ -A hydraulically and electrically actuated quadruped robot”. In: *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering* 225 (6 2011), pp. 831–849. ISSN: 20413041. DOI: 10.1177/0959651811402275.

[44] Claudio Semini et al. “Design of the Hydraulically Actuated, Torque-Controlled Quadruped Robot HyQ2Max”. In: *IEEE/ASME Transactions on Mechatronics* 22 (2 Apr. 2017), pp. 635–646. ISSN: 10834435. DOI: 10.1109/TMECH.2016.2616284.

[45] Hamza Khan et al. *Development of the Lightweight Hydraulic Quadruped Robot - MiniHyQ*. 2015.

[46] Hamza Khan et al. *DEVELOPMENT OF A LIGHTWEIGHT ONBOARD HYDRAULIC SYSTEM FOR A QUADRUPED ROBOT*. 2015.

[47] Xuewen Rong et al. “Design and simulation for a hydraulic actuated quadruped robot”. In: *Journal of Mechanical Science and Technology* 26 (4 Apr. 2012), pp. 1171–1177. ISSN: 1738494X. DOI: 10.1007/s12206-012-0219-8.

[48] Keith W. Wait and Michael Goldfarb. “A pneumatically actuated quadrupedal walking robot”. In: *IEEE/ASME Transactions on Mechatronics* 19 (1 Feb. 2014), pp. 339–347. ISSN: 10834435. DOI: 10.1109/TMECH.2012.2235078.

[49] Kurt S Aschenbeck et al. *Design of a Quadruped Robot Driven by Air Muscles*. 2006. URL: http://biorobots.case.edu,.

[50] Alexander Hunt, Nicholas Szczecinski, and Roger Quinn. “Development and training of a neural controller for hind leg walking in a dog robot”. In: *Frontiers in Neurorobotics* 11 (APR Apr. 2017). ISSN: 16625218. DOI: 10.3389/fnbot.2017.00018.

[51] Wikipedia. *Biarticular muscle*. 2021. URL: https://en.wikipedia.org/wiki/Biarticular_muscle#:~:text=Biarticular%20muscles%20are%20muscles%20that,the%20hip%20and%20the%20knee..

[52] Xin Wang et al. “Design and Development of a Cheetah Robot under the Neural Mechanism Controlling the Leg’s Muscles”. In: (2012).

[53] Yasuhiro Fukuoka et al. “Pace Running of a Quadruped Robot Driven by Pneumatic Muscle Actuators: An Experimental Study”. In: *Applied Sciences (Switzerland)* 12 (9 May 2022). ISSN: 20763417. DOI: 10.3390/app12094146.

[54] Yasuhiro Fukuoka et al. “Autonomous speed adaptation by a muscle-driven hind leg robot modeled on a cat without intervention from brain”. In: *International Journal of Advanced Robotic Systems* 18 (5 2021). ISSN: 17298814. DOI: 10.1177/17298814211044936.

[55] R. McN. Alexander. *ENERGY-SAVING MECHANISMS IN WALKING AND RUNNING*. 1991, pp. 55–69.

[56] V.A. Tucker. *The Energetic Cost of Moving About*. 1975.

APPENDIX A
TABLE OF ROBOTS

Name	Year	Gait	DOF (/leg)	Leg configuration	Actuation	Mass (kg)	BL (m)	L (m)	Speed (m/s)	PL (kg)	Fr (-)	NS (s^{-1})	PLC (%)	COT (-)
AiDIN-VI[37]	2019	T, P	3	All-elbow	Electric	43.7 (on) 39.7 (off)	0.762	0.68	1.2	25	0.22*	1.57	57.21	1.19 (T) 2.02 (P)
ANYMAL[26][27][28]	2016	T	3	Inward	Electric	50 (on) 46.5 (off)	0.93	0.56	1.3	23	0.31*	1.4	46	1.23
BigDog[41]	2008	T, B	4	-	Hydraulic	109 (on)	1.1	1	3.1 (B) 2 (T)	50	0.98	2.82	45.87	15
Cheetah Robot[52]	2012	B	3	-	Pneumatic	70 (off)	0.8	0.7	2.7	-	1.06*	3.38	-	-
GE Walking Truck[4]	1968	W	3	All-knee	Hydraulic	1300 (off)	4	2.5	2.2	70	0.2*	0.55	5.38	-
Go2[36]	2023	T	3	All-elbow	Electric	25 (on)	0.7	0.4	2.5	7	1.59*	3.57	31.82	-
HuboDog[30]	2010	T	3	Inward	Electric	42 (on)	0.8	0.6	0.55	24	0.05	0.69	57.14	-
HyQ[42][43]	2010	T	3	Inward	Hydraulic*	91 (on) 70 (off)	1	0.68	2	50	0.6	2	55.56	0.78*
HyQ2Max[44]	2017	T	3	Inward	Hydraulic	80 (off)	1.306	0.74	1.5	40	0.31*	1.15	50	-
KOLT[29]	2008	T	3	All-elbow	Electric*	80 (on)	1.75	0.7	1.1	-	0.18	0.63	-	2.57
MiniHyQ[45]	2015	-	3	Inward	Hydraulic	35 (on) 24 (off)	0.85	0.4	1.8	-	0.83*	2.12	-	-
MIT Cheetah 1[31][32]	2013	T	3	-	Electric	33 (on)	1	0.5	6	-	7.34	6	-	0.51
MIT Cheetah 2[33][34]	2015	B	3	-	Electric	33 (on)	0.7	0.59	6.4	-	7.1	9.14	-	0.47
MIT Cheetah 3[35]	2018	T	3	All-elbow	Electric	45 (on)	0.6	0.5	1.6	-	0.5*	2.67	-	0.45
PMA quadruped[53][54]	2022	P	3	-	Pneumatic	7.4 (off)	0.485	0.24	0.75	-	0.24*	1.55	-	-
Puppy[49][50]	2006	W	3	Inward	Pneumatic	6.4 (off)	0.6	0.51	1	24.4	0.2*	1.67	381.25	-
Raibert's Quadruped[14]	1986	T, P, B	3	-	Hydraulic	38 (-)	1.05	0.56	2.9 (B) 2.2 (T)	-	1.53	2.76	-	-
Scalf[47]	2011	T	3	Inward	Hydraulic	65 (off)	1.1	0.6	1.8	100	0.55*	1.64	153.85	-
Scout II[22]	2005	B	2	-	Electric	24.8 (on)	0.84	0.32	1.3	-	0.53	1.55	-	1.38*
SPOT[39]	2016	T	3	All-elbow	Electric	32 (on) 28 (off)	1.1	0.6	1.6	14	0.43*	1.45	46.67	0.8*
Stanford Doggo[25]	2019	T, B	2	-	Electric	4.8 (on)	0.42	0.14	0.9	-	0.59*	2.14	-	3.2
StarIETH[23][24]	2014	W, T	3	Inward	Electric	25 (on)	0.505	0.4	0.7	25	0.21	1.39	100	3.5 (W) 1.7 (T)
Titan XIII[16]	2016	W	3	Sprawling	Electric	5.65 (on) 5.29 (off)	0.2134	0.297	1.38	5	0.65*	6.47	88.5	1.76
VU quadruped[48]	2010	W	3	Sprawling	Pneumatic	6.9 (off)	0.46	0.38	0.46	-	0.06*	1	-	-
WarpI[38]	1998	W	3	All-knee	Electric	60 (on)	0.8	0.59	1.2	-	0.25*	1.5	-	0.55*

Gait: B = Bound, P = Pace, T = Trot, W = Walk
Mass: On/off-board power supply
Fr and COT with a * were not found in literature but are computed by myself with Equation 1 and Equation 4, respectively
HyQ actuation: The hip a/a is driven electrically
KOLT actuation: A pneumatic spring is used for leg extension

3

Research Paper

Design and Development of a prototype for a Robot Monkey: Enabling Transition from Quadrupedal to Statically Stable Upright Standing

Tom Kuijlaars
MSc Mechanical Engineering
Precision and Microsystems Engineering
Delft University of Technology

***Abstract*—This paper presents the development and analysis of a quadruped robot prototype designed to achieve statically stable rear leg standing. Existing quadruped robots such as Mini Cheetah and Go2 are versatile, but rely on dynamic movement to remain stable. We present a robot prototype that can transition from quadrupedal to an upright stance while remaining statically stable throughout using a tail. This Tail-assisted Transition Robot (further: TT-Bot) utilizes electric actuators and combines aluminum and 3D printed nylon components to form a lightweight yet robust structure. It aims for a functional approach through strategic motor placement and a belt-pulley system for knee joint actuation. Kinematic analyses guide the control strategies implemented to manage the transition. Initial results from simulations and physical tests indicate that TT-bot can effectively maintain an upright stance under controlled conditions. Stability tests indicate successful static stability across slopes from -10° to 5°. A payload test showed that it was able to transition with a payload of 3 kg. Challenges during transition remain with steeper inclines and higher payloads.**

I. Introduction

Currently, there is significant interest in mobile robots. Legged robots offer significant advantages compared to conventional vehicles, as they enable locomotion in terrains that are inaccessible to wheeled and tracked vehicles [1] [2].

Existing quadruped robot such as SPOT [3], Mini Cheetah [4] and ANYmal [5] are very versatile. Mini Cheetah and Go2 [6] can do a somersault and the latter can stand on its two front legs. However, Go2 needs to constantly move to remain stable. Other quadruped robots are able to stand statically stable on their rear legs. In these cases though, either the robot is not completely stable during transition [7], or the robot uses very large feet which compromises the ability to maintain a dynamic quadruped gait [8].

The goal of this research is to design and develop a prototype quadruped robot capable of standing on its rear legs without compromising its ability to maintain a dynamic quadrupedal gait. This goal is divided into two sub goals: transitioning from quadrupedal standing to rear leg standing and remaining statically stable on rear legs. A tail is used so the robot can have a quasi-static transition and remain statically stable on its rear legs. Reflecting its core functionality, the prototype is named "TT-Bot", an acronym for Tail-assisted Transition Robot, highlighting its unique design and capabilities. TT-Bot is tested for stability, reach height and its capability to carry a payload. The research is done in cooperation with Avular [9].

This paper is structured as follows. In Section 2 the methodology is explained. The design is shown, analyses are done and the experiments are described. Section 3 presents the results of the simulations and experiments. In Section 4 the results are discussed and future research is suggested. Section 5 covers the conclusion of this paper.

II. Method

This section presents the design requirements, final design, a kinematic analysis and finally the control setup and test setup.

A. Requirements & Constraints

Being a proof-of-concept prototype, TT-Bot, has been developed to operate within a 2D plane. While not being a full quadruped, it remains effective in validating the sub-goals. When quadrupedal standing is mentioned, it refers to standing on one rear leg and one front leg, since TT-Bot is not a full quadruped. The functional requirements and constraints are mentioned in Table I

TABLE I: Requirements & Constraints of TT-Bot with the reason

Requirement/Constraint	**Rationale**
Quasi-static transition	Research gap & concept study
Statically stable upright standing	Research gap & concept study
Use of maximum 6 RMD motors [10]	Motor availability at Avular
Inward pointing configuration	Aesthetics
Carry payload of at least 2 kg	Expected payload for the final robot
Reach a minimum height of 120 cm	Maximum height of a door handle
Able to transition on slope of minimal $\pm 5^\circ$	Operate in industrial area
Maximum 175 mm extending tail in retracted state	Prevent collision with environment during quadruped gait

B. Design

A schematic overview of the design of TT-Bot is shown in Figure 1. The main components are the body, front leg, rear leg and tail. Both legs and the tail consist of two links that are connected via a revolute "knee" joint. The upper link is connected to the body via a revolute "hip" joint. This results in a total of six revolute joints that are actuated by the motors mentioned in Table I. The legs and tail are all made of nylon

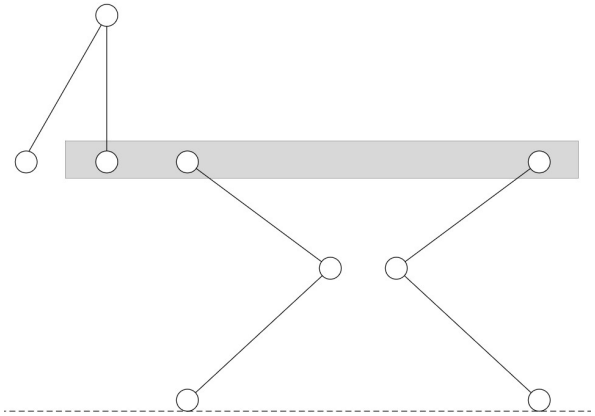
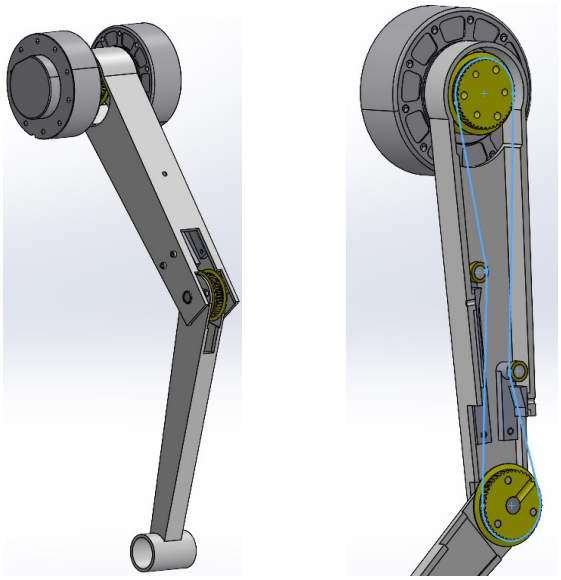


Fig. 1: Schematic overview of the design



(a) Model of front leg (b) Belt-pulley system

Fig. 2: SolidWorks models of one leg, the blue line represents the belt

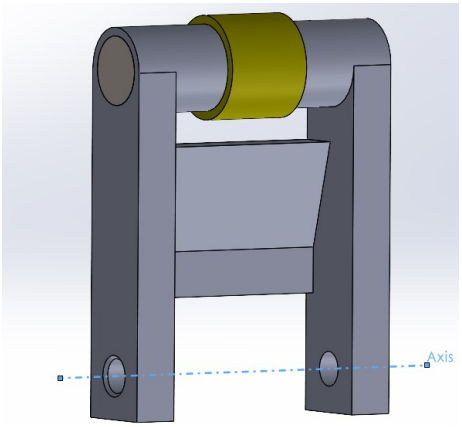


Fig. 3: SolidWorks model of the tensioner with a steel rod and roller bearing, the axis shown is the pivot point

and 3D printed with Selective Laser Sintering-3D printing. The body is made of aluminum sheet metal.

Each motor has a mass of 710 gram, an outer diameter of 98.0 mm and an axial length of 51.5 mm. They cover more than 60% of the total weight. Therefore it is important where the actuators are positioned to influence the robot Center of Mass (CoM).

The motors for both the hip and knee joints are placed in parallel in the hip area, as is shown in Figure 2a. The upper leg is directly attached to the right motor. This motor is responsible for the rotation of the hip joint. A belt-pulley system is used to rotate the knee joint. The belt transmits the rotation of the knee motor to the knee hinge, as shown in Figure 2b. This layout reduces the mass and inertia of the legs.

The hip pulley and knee pulley are both 3D printed and have the same diameter and number of teeth. Therefore the gear/motion ratio between motor and joint is one. In order to properly tension the belt, two tensioners are attached to the inside of the upper leg. The tensioners are pushed inwards and hinge around the axis shown in Figure 3. These tensioners were specifically designed to fit in the upper leg and are also 3D printed. The rubber belt travels over the roller bearing to reduce friction and wear. The knee pulley is rigidliy connected to the lower leg. A steel rod is clamped inside the knee pulley with a set screw. This rod, and therefore the whole lower leg, is able to rotate with a set of flanged plain bearings that are present in the upper leg.

The rear leg is identical to the front leg except for the foot. Due to the planar configuration of TT-Bot, it faces a challenge in maintaining lateral stability. To address this issue, the width of the foot of the rear leg has been increased to match that of the body. This design feature effectively prevents lateral tipping.

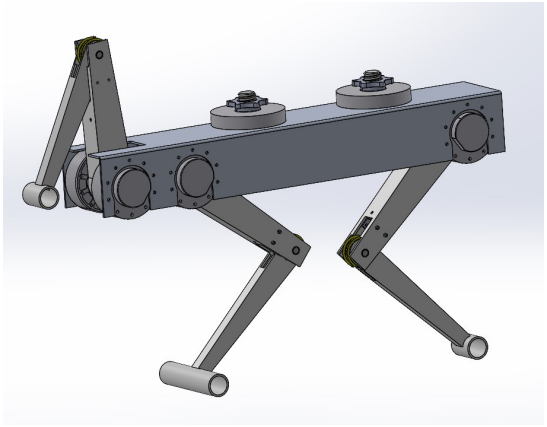
The tail of TT-Bot is slightly shorter than its legs, because it was specified to extend at most 175 mm while in retracted state. However, the working principle of the tail is identical to the legs.

All the motors are directly mounted to the body. With this configuration the upper and lower leg are parallel linked. This means the upper and lower leg both rotate with respect to the orientation of the body. This principle is used for the inverse kinematics in subsection II-C. A favorable side effect is that the legs are positioned in the middle of the body, which also effectively prevents lateral tipping.

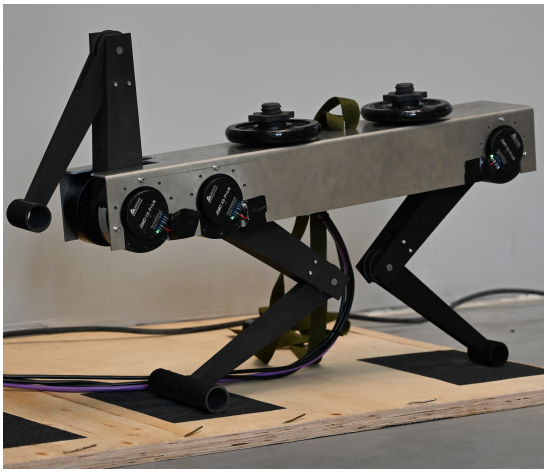
A SolidWorks model and physical model of the design are shown in Figure 4.

C. Kinematic Analysis

Each leg of TT-Bot can be modelled as an open loop kinematic chain. The foot is modelled as a revolute joint connected to the fixed world. The lower leg L_1 connects the foot to the knee joint. The knee is a movable revolute joint.



(a) SolidWorks model



(b) Physical model

Fig. 4: Representation of TT-Bot

The upper leg L_2 connects the knee joint to the hip joint. For one leg the hip joint is modelled as the end-effector.

With Grübler's formula the mobility of the end-effector can be calculated [11].

$$m = 3(l - n - 1) + \sum_{i=1}^{n} d_i \tag{1}$$

where l represents the total number of rigid bodies, including the base, n is the total number of joints, and d_i the number of degrees of freedom (DoF) for joint i.

As described above, one leg consists of 3 rigid bodies including base and 2 revolute joints with 1 DoF, this yields the following for Equation 1:

$$m = 3(3 - 2 - 1) + 2 = 2$$

This results in a 2 DoF system. The end-effector is able to translate in x and y direction, it also has a coupled angle of orientation with respect to the other legs. A schematic view of the system can be seen in Figure 5.

The x and y position of the end-effector can be calculated using the joint variables, this is called forward kinematics [12].

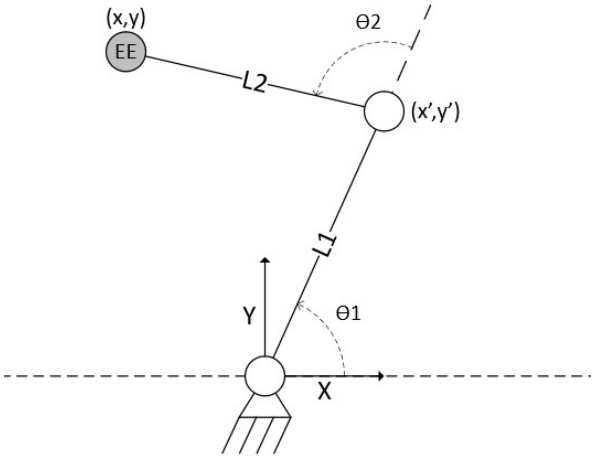


Fig. 5: Forward kinematics of one leg

First the position of the knee joint is calculated using trigonometry:

$$\begin{aligned} x' &= L_1 \cos\theta_1 \\ y' &= L_1 \sin\theta_1 \end{aligned}$$

The position of the end-effector relative to the knee position:

$$\begin{aligned} x'' &= L_2 \cos(\theta_1 + \theta_2) \\ y'' &= L_2 \sin(\theta_1 + \theta_2) \end{aligned}$$

Combining these results gives the position of the end-effector:

$$\begin{aligned} x &= L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) \\ y &= L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) \end{aligned} \tag{2}$$

With forward kinematics it is possible to calculate the position of the end-effector given the joint angles. To control TT-Bot the position of the end-effector is needed as input. With these given coordinates the position of the knee joint and the joint angles can be calculated using inverse kinematics.

The position of the foot (x_{base}, y_{base}) and hip (x, y) are both prescribed. First the distance r between them is calculated with Pythagorean theorem:

$$\begin{aligned} dx &= x - x_{base} \\ dy &= y - y_{base} \\ r &= \sqrt{dx^2 + dy^2} \end{aligned} \tag{3}$$

Using the law of cosines, the coordinates of point P along line r are calculated and with Pythagorean theorem the height h:

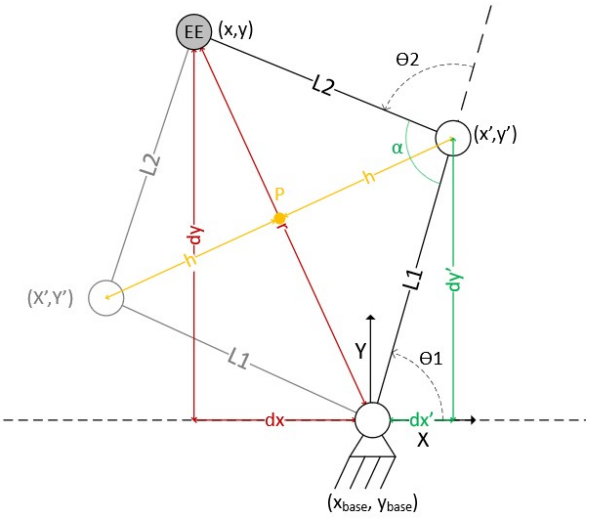


Fig. 6: Inverse kinematics of one leg

$$
\begin{aligned}
a &= \frac{{L_1}^2 - {L_2}^2 + r^2}{2r} \\
x_p &= x_{base} + \frac{a \cdot dx}{r} \\
y_p &= y_{base} + \frac{a \cdot dy}{r} \\
h &= \sqrt{{L_1}^2 - a^2}
\end{aligned}
\tag{4}
$$

With the coordinates of point P and distance h, the position of the knee joint can be calculated. For a given position of the end-effector there are two different solutions possible (inward and outward orientation) as can be seen in Figure 6.

$$
\begin{aligned}
x' = x_p + h \cdot \frac{dy}{r} \quad or \quad X' = x_p - h \cdot \frac{dy}{r} \\
y' = y_p - h \cdot \frac{dx}{r} \quad or \quad Y' = y_p + h \cdot \frac{dx}{r}
\end{aligned}
\tag{5}
$$

Once the position for the knee is chosen, the angle θ_1 can be easily calculated using Pythagorean theorem. The angle θ_2 can be computed using the law of cosines.

$$
\begin{aligned}
dx' &= x' - x_{base} \\
dy' &= y' - y_{base} \\
\alpha &= \cos^{-1}\left(\frac{{L_1}^2 + L_2^2 - r^2}{2 \cdot L_1 \cdot L_2}\right)
\end{aligned}
$$

dx' and dy' are used to calculate θ_1 and α is used to calculate θ_2:

$$
\begin{aligned}
\theta_1 &= \tan^{-1}\left(\frac{dy'}{dx'}\right) \\
\theta_2 &= \pi - \alpha
\end{aligned}
\tag{6}
$$

The kinematic analysis of one leg is used for the kinematic analysis of the entire robot.

For the quasi-static transition method, the CoM of the entire robot shifts backwards. Initially, TT-Bot adopts a “Quadruped” stance where both the front and rear legs are in contact with the ground, see Figure 7a. Subsequently, as the transition progresses, the tail also makes contact with the ground, leading to the “Transition” stance, see Figure 7b. As the CoM shifts from the front and rear leg support polygon to the rear leg and tail support polygon, the front leg can be lifted from the ground, resulting in the “Upright” stance where only the tail and rear leg serve as supports, see Figure 7c. This sequential process yields three distinct instances for conducting kinematic analysis.

For all instances TT-Bot is a parallel manipulator, for a generalized parallel manipulator is a closed-loop kinematic chain mechanism whose end-effector is linked to the base by several independent kinematic chains [13].

During the ”Quadruped” stance, the rear and front leg form a closed loop kinematic chain, since the rear and front foot contact points are modelled as revolute joints connected to the fixed world. During the ”Upright” stance, the tail and rear leg form a closed loop kinematic chain, since the tail and rear foot are modelled as revolute joints connected to the fixed world. During the ”Transition” stance, both closed loop kinematic chains apply. All three contact points are modelled as revolute joints connected to the fixed world.

The mobility of the mechanism is calculated using Grübler’s formula. For both the ”Quadruped” and ”Upright” stance the closed loop chain consists of six bodies and six single DoF joints. For the ”Transition” stance the closed loop chain consists of eight bodies and nine 1DoF joints. Filling this in Equation 1 results in:

$$
\begin{aligned}
m &= 3(6 - 6 - 1) + 6 = 3 \\
m &= 3(8 - 9 - 1) + 9 = 3
\end{aligned}
$$

In this study, TT-Bot is identified as a fully parallel planar manipulator, possessing three DoF: two translations and one rotation [13]. Despite having more than three actuated joints, it is not redundantly actuated. Each actuator is essential for precise control over the DoF of the end-effector, specifically tailored to meet the unique kinematic requirements of TT-Bot.

The body (end-effector) of TT-Bot is able to translate in x and y direction and rotate about the z-axis. The position and orientation of the end-effector are described as x_{com}, y_{com} and ϕ, where (x_{com}, y_{com}) is the coordinate of the CoM of TT-Bot and ϕ is the angle of the body relative to horizontal. The combined mass of the body including motors is 80% of the total mass. Therefore we assume that the CoM of the body is at the same position as the CoM of the whole robot. In Figure 7 it can be seen that the CoM stays almost exactly at the same place in the body for the different poses.

The positions of the feet, i.e. the base joints depend on the angle (γ) of the slope the robot is standing on, which can be seen in Figure 8. These positions are calculated in Equation 7.

(a) "Quadruped" stance, CoM is positioned between front and rear leg.

(b) "Transition" stance, CoM is positioned above the rear leg.

(c) "Upright" stance, CoM is positioned between rear leg and tail.

Fig. 7: Three different stances with their CoM shown

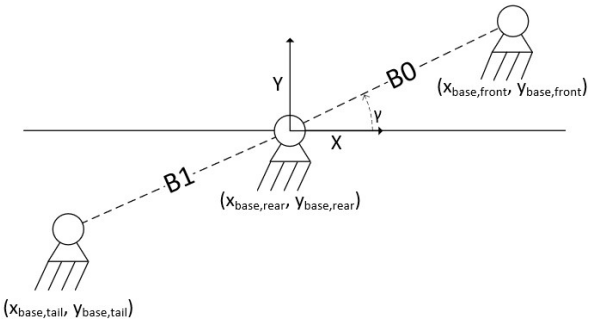


Fig. 8: Position of the base joints on a slope

$$
\begin{aligned}
x_{base,rear} &= 0 \\
y_{base,rear} &= 0 \\
x_{base,front} &= B_0 \cos\gamma \\
y_{base,front} &= B_0 \sin\gamma \\
x_{base,tail} &= -B_1 \cos\gamma \\
y_{base,tail} &= -B_1 \sin\gamma
\end{aligned} \tag{7}
$$

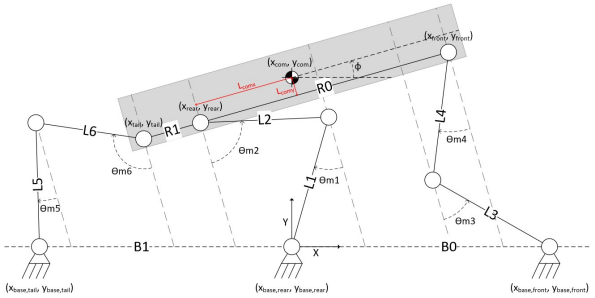


Fig. 9: Inverse Kinematics of TT-Bot

The position of the hip joints for the tail (x_{tail}, y_{tail}), rear leg (x_{rear}, y_{rear}) and front leg (x_{front}, y_{front}) can be seen in Figure 9 and are computed on the basis of the position and orientation of the CoM in Equation 8.

$$
\begin{aligned}
x_{com} &= x \\
y_{com} &= y \\
x_{tail} &= x - (L_{comx} + R_1)\cos\phi + L_{comy}\sin\phi \\
y_{tail} &= y - (L_{comx} + R_1)\sin\phi - L_{comy}\cos\phi \\
x_{rear} &= x - L_{comx}\cos\phi + L_{comy}\sin\phi \\
y_{rear} &= y - L_{comx}\sin\phi - L_{comy}\cos\phi \\
x_{front} &= x + (R_0 - L_{comx})\cos\phi + L_{comy}\sin\phi \\
y_{front} &= y + (R_0 - L_{comx})\sin\phi - L_{comy}\cos\phi
\end{aligned} \tag{8}
$$

With the positions of the hip joints and base joints known, the positions of the knee joints are calculated using Equation 3, Equation 4 and Equation 5 in the same way as done for one leg. This results in two possible configurations for the knee joint. For the tail and front leg the position is chosen for x_{min} and for the rear leg for x_{max}. The hip and knee joints are the actuated joints for TT-Bot, so these need to be calculated. As mentioned before, the upper and lower leg move independently from each other. So the angles of the joints are calculated with respect to the body orientation. The motor angles for the knee joints and hip joint from Figure 9 can be calculated using Pythagorean theorem.

$$
\begin{aligned}
rear: \quad & \\
dx_1 &= x_{knee} - x_{base} \\
dy_1 &= y_{knee} - y_{base} \\
dx_2 &= x_{knee} - x_{rear} \\
dy_2 &= y_{rear} - y_{knee}
\end{aligned}
\qquad
\begin{aligned}
front/tail: \quad & \\
dx1 &= x_{base} - x_{knee} \\
dy1 &= y_{knee} - y_{base} \\
dx2 &= x_{rear} - x_{knee} \\
dy2 &= y_{rear} - y_{knee}
\end{aligned}
$$

$$
\begin{aligned}
& rear: && front/tail: \\
\theta_{m1} &= \tan^{-1}\frac{dx1}{dy1} + \phi & \theta_{m3/m5} &= \tan^{-1}\frac{dx1}{dy1} - \phi \\
\theta_{m2} &= \tan^{-1}\frac{dx2}{dy2} - \phi & \theta_{m4/m6} &= \tan^{-1}\frac{dx2}{dy2} + \phi
\end{aligned} \tag{9}
$$

TABLE II: Dimensions of parameters from Figure 9

Part	Length
L_1	265 mm
L_2	239 mm
L_3	265 mm
L_4	239 mm
L_5	245 mm
L_6	203 mm
B_0	480 mm
B_1	470 mm
R_0	480 mm
R_1	110 mm

The values for L_1, L_3 and L_5 are measured from the bottom of the foot to axis of rotation of the knee joint. The values for L_2, L_4 and L_6 are measured from the axis of rotation of the knee joint to the axis of rotation of the hip joint. As mentioned in subsection II-B the legs of TT-Bot are 3D printed. The maximum possible length to 3D print is 300 mm. The length of the upper leg is dependent on both the 3D printer's capabilities and the specific length of the belt, which is 612 mm. The total length of the 3D printed upper leg is 289 mm. The length of the lower leg is chosen to achieve a mammal-like ratio between the upper and lower legs. The distance between the base joints B_0 is chosen to be the same value as R_0, so the hip joints are both located above the feet in the initial position. The value of R_1 is chosen so the actuators for the tail can fit directly behind the actuators of the rear leg. The values for the tail depend on the retracted state constraint from Table I. The upper part of the tail it is also dependent on the belt, which is 540 mm long.

D. Control

The motors utilized operate under velocity input control. Although position control is preferred for TT-Bot. The objective is to designate a specific angle to the motors and ensure the motors rotate to achieve this angle. Consequently, a PID controller is designed. The error pos_error_rad between the desired motor position pos_set and the real-time position $motor\_pos\_rad$ is given as input in the PID controller. This results in a desired output velocity in rpm for the motors. MATLAB Simulink is used to control the motors. A simplified schematic control diagram is shown in Figure 10. At the start, both legs and the tail are set in a straight vertical position downwards. In this initial position all the motor angles are set to zero.

For TT-Bot to stand up, it has to traverse a specified path. With SolidWorks, a few positions for the body are chosen where the CoM is in the correct spot to maintain stability. The first position is the initial position with both legs and

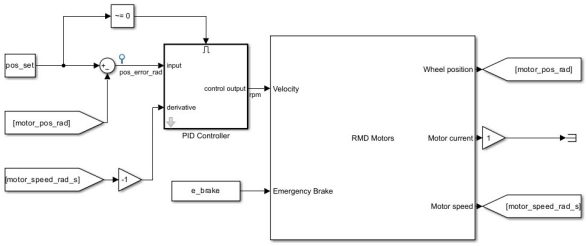


Fig. 10: Simplified controller for position control

tail in a vertical line down. The second position can be seen in Figure 7a. This is the position that would be used for quadruped walking. The third and fourth positions look like Figure 7b, but in the third position, the projected CoM lies within the front and rear leg support polygon. In the fourth position the projected CoM lies within the tail and rear leg support polygon. The fifth position can be seen in Figure 7c. In its final position, TT-Bot is completely upright.

For every position mentioned above, x_{com}, y_{com} and ϕ are known. Using MATLAB, a path trajectory is made for the body, which is done using the function *cubicpolytraj*. This function generates a third-order polynomial trajectory of the end-effector position and orientation. Since the CoM of the whole robot is not perfectly calculated, the transition stance is executed from $x_{com} = 50$ mm until $x_{com} = -50$ mm, which can be seen in Figure 11. This ensures that all feet are on the ground around the tipping point.

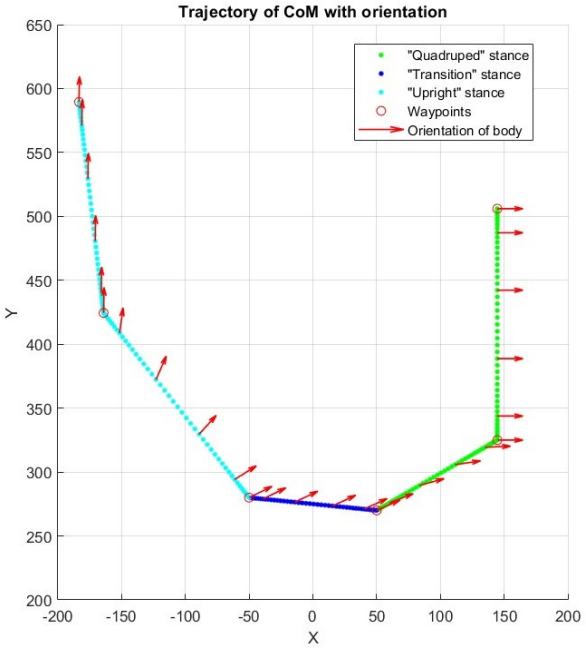


Fig. 11: Path trajectory of CoM with orientation

For every trajectory point on the path the knee positions and motor angles are calculated using the inverse kinematics from

subsection II-C. For the ”quadruped” stance, the position for the tail joints cannot be calculated with inverse kinematics, since the tail foot is not able to touch the ground. An extra trajectory is made by selecting values for θ_{m5} and θ_{m6} at the first three waypoints in Figure 11. The angles of the third waypoint are the angles that were calculated with inverse kinematics. A path for the motor angles is made, and with these motor angles, the positions of the knee and foot are calculated using forward kinematics.

For the ”upright” stance, the position for the front leg cannot be calculated with inverse kinematics. Therefore an extra trajectory is also made for this leg. Values for θ_{m3} and θ_{m4} are chosen for the last three waypoints in Figure 11. The angles of the first waypoint are the ones that were calculated with inverse kinematics. For the final position the lower front leg is positioned in a vertical line above the rear leg. The other angles are chosen such that no collisions occur. For these angles, a path is made, and the positions of the knee and foot are calculated using forward kinematics.

A simulation of the whole path of TT-Bot is made to ensure all joint angles are possible and a smooth transition is made between the angles calculated with inverse kinematics and the angles chosen for the extra trajectories. In Figure 12, multiple snapshots of the simulation are shown.

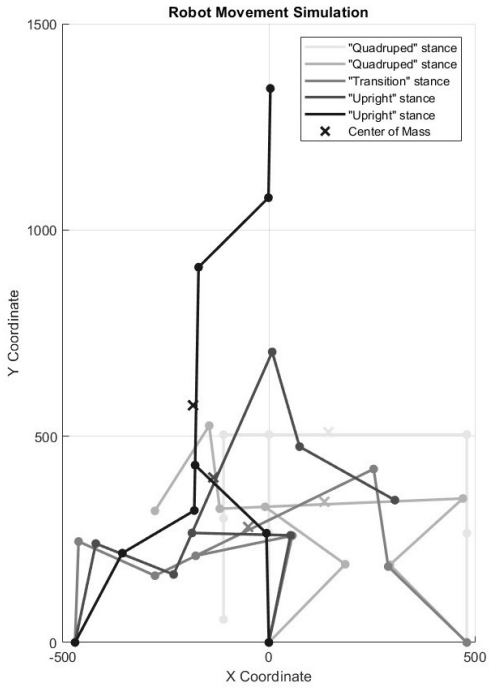


Fig. 12: Snapshots of the simulation

All the angles for the motors calculated by the different path trajectories are given as input pos_set in the controller from Figure 10.

E. Testing

TT-Bot is tested to see if it meets the criteria mentioned in subsection II-A. For every single test a different path is generated and visualized with simulation to make sure there are no collisions and the CoM is positioned in the correct places. A picture of the test setup can be seen in Figure 13. Anti-slip tape is attached to the feet and grip tape is put on the wooden board to give the feet more grip on the ground. A rope is connected to the body to hold it in the air in case it falls.

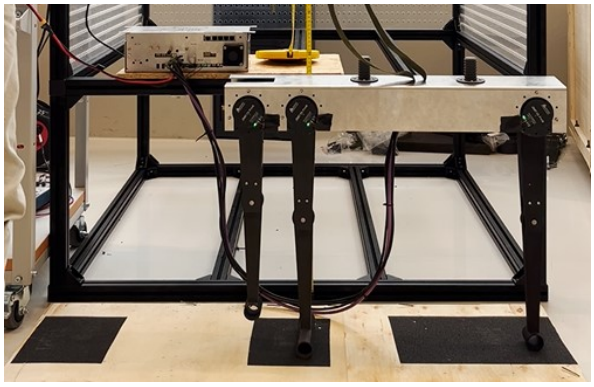


Fig. 13: Test setup, TT-Bot is in its initial position to reset the motor angles

1) Stability: The stability of TT-Bot is tested with two different tests. For both tests the payload is 2 kg, since this is one of the requirements. The reach height is measured for both tests. This is the distance from the rear foot to the front foot in the final position.

The first method to test the stability is to change the slope when TT-Bot is already in its final upright position. Starting from an angle of $0°$, the slope is slowly increased until TT-Bot tips over its tail. This is repeated, while the angle is slowly decreased until TT-Bot tips over its rear leg. At these points the projected CoM is positioned outside of the support polygon. The slope for the tipping points is also calculated with MATLAB and compared to the experiment. This test is done for different locations of the base of the tail, so with different values for B_1.

The second stability test is the ability to stand up on different slope angles $\gamma = [-10°, -5°, 0°, 5°, 10°]$, as seen in Figure 14. The position of the feet change according to Equation 7. The maximum achievable reach height is measured at these different angles and compared to the simulation done.

2) Payload: On the back of TT-bot's body, weights can be mounted to imitate a payload. With this payload the position of the CoM changes. With SolidWorks the CoM of the body is calculated for the different payloads. The different values for L_{comx} and L_{comy} can be seen in Table III. These values are used in the MATLAB script for the path trajectory. In the last column the body weight with respect to the total weight is shown, $f = \frac{m_{body}}{m_{total}}$. For higher payloads the position of the

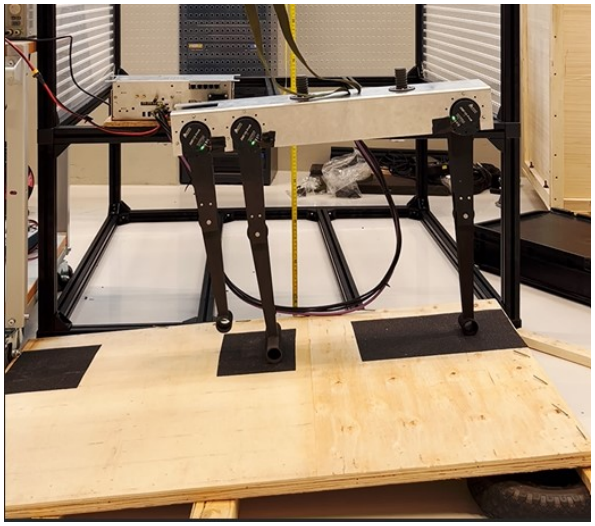


Fig. 14: Test setup, TT-Bot is in its initial position to reset the motor angles on a slope of 5 degrees

CoM is more accurately calculated. TT-Bot is tested on the ability to stand up with the payload on a horizontal surface.

TABLE III: Position of CoM for different payloads

Payload	L_{comx}	L_{comy}	f
0 kg	145 mm	7 mm	80%
1 kg	176 mm	16 mm	83%
2 kg	170 mm	23 mm	85%
3 kg	192 mm	28 mm	86%
4 kg	185 mm	32 mm	88%

III. Results

In this section the results of the different experiments and simulations are presented.

A. Stability

In Figure 15, the stability angle is plotted against the different positions for the base of the tail. For these different base positions, the reach height also changes, as can be seen in Figure 16. Since both the reach height and stability angle are plotted against the base distance, it is possible to make a new plot where the stability angle is plotted against the reach height, as seen in Figure 17.

For a base distance B_1 of 400 mm, TT-Bot was not able to stand up, as the red cross indicates in Figure 16, Figure 15 and Figure 17.

For the second stability test, the slope at which TT-Bot is able to stand up is measured. For different slopes TT-Bot has different reach heights. TT-Bot was not able to stand up at a slope of 10°, as the red cross indicates in Figure 18.

B. Payload

TT-Bot was able to stand up with a payload of 1, 2 and 3 kg. It was not able to stand up with a payload of 4 kg.

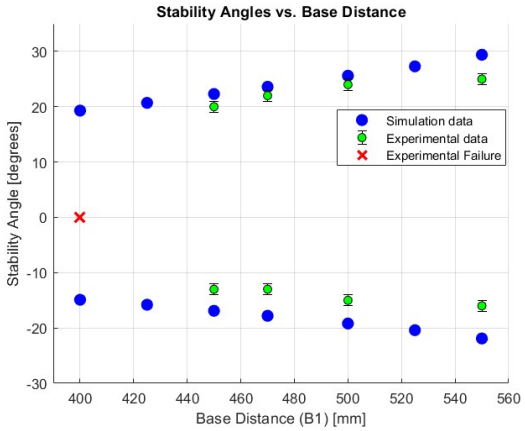


Fig. 15: Stability vs base width, the upper data points: tip over at tail, the lower data points: tip over at rear leg

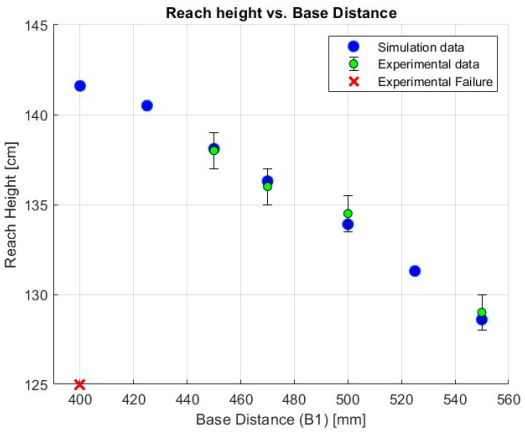


Fig. 16: Reach height vs base width

IV. Discussion

In this section the results will be discussed regarding the design, simulations and the experiments. Finally, some recommendations for future research are presented.

A. Design

The decision to use a tail as additional support ensures that the TT-Bot remains stable during both the transition and the final position. This is evident from both simulations and various experiments. The U-shape of the body makes it sensitive to torsion. This, combined with positioning the knee motor within the body frame, introduces some additional complications. The belt pulley system works to transmit the motor rotation to knee rotation. However, the tension of the belt also causes it to pull on one side of the body, causing the body to warp slightly. This particularly affected the lateral stability of the entire robot during the experiments.

The upper leg has a large opening at the end since the lower leg needs to be able to rotate within it. This causes the ends

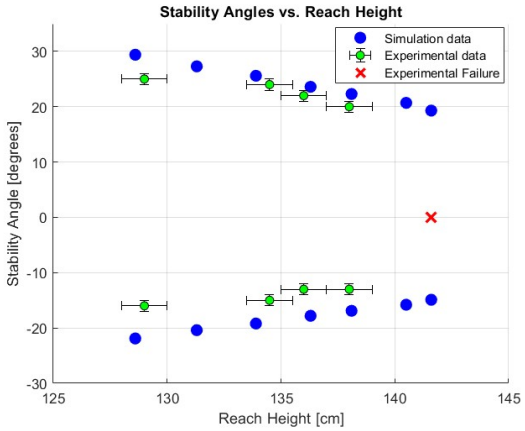


Fig. 17: Stability vs reach height

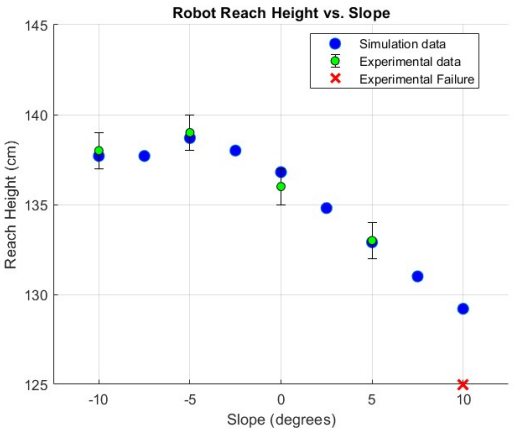


Fig. 18: Height vs slope

to bend outwards and allows the lower leg to move slightly from left to right in the sliding bearing of the upper leg.

B. Experiments

TT-bot's feet are modelled as fixed revolute joints. Anti-slip tape and grip tape are used to simulate this as closely as possible. However, the feet can still slide over the ground. Also, TT-Bot rotates on a foot and not at the pivot point as modeled by a revolute joint. This causes the lengths of L_1, L_3 and L_5 to change slightly.

For motor control, the angles of the motors are first reset to 0 degrees when both legs and the tail are set to their initial position. This is done manually and by eye, which can result in small deviations from their actual angles. This causes the results of the experiments to not precisely match the simulations. Additionally, this means that slightly different results may occur if an experiment is repeated multiple times.

C. Results

The results of the experiments and the differences with the simulations are discussed.

1) Stability test 1: In the first stability test the width of the base changes. For a base width of 400 mm TT-bot was not able to stand up, as indicated by the red cross that can be seen in Figure 16. However, it did manage to go through the transition phase. While standing up on its rear leg and tail, it wobbled forwards and backwards until it was about to fall. The results of the first stability test show the angle at which TT-Bot, in final upright position, fails to remain stable. In the simulation data, this is the point where the projected CoM is outside the support polygon. This should also be reflected in the experimental data. However, Figure 15 illustrates that the experimental and simulation data align within 70% for the decreasing angle and within 85% for the increasing angle. TT-Bot fails to remain stable at lower angles than predicted by the simulation data. This can have various explanations. First of all, stability test 1 was carried out by manually lifting the wooden board with TT-Bot in final upright position at one end to create a slope. Therefore, it is plausible that the effective slope is not perfect in the 2D plane, resulting in TT-Bot to fall earlier. Secondly, as mentioned before, the CoM of the whole robot is not modelled perfectly. Introducing the possibility, that the CoM is positioned higher, which would typically result in a lower stability angle.

2) Stability test 2: For the second stability test, TT-Bot's ability to transition on different slopes is tested. Afterwards, the reach height is measured and compared with the simulation. TT-bot was not able to stand up on a slope of $10°$, as indicated by the red cross that can be seen in Figure 18. Although, it could stand still on the slope in its initial position, when the test was started, it immediately fell backwards. In a subsequent test, it was able to go through the transition phase. However, because of the steepness of the angle and therefore the position of the CoM, most of the weight was supported by the rear leg, which could not handle this and collapsed.

From Figure 18, it can be seen that the deviation between experimental and simulation data fall within the error margin of 1 cm. The simulation data shows that the maximum reach height can be obtained around a slope of $-5°$. At this slope the reach height is limited by both the tail and rear leg, since both are fully extended. At a slope of $-7.5°$ and $-10°$ the reach height is the same. At these slopes the reach height depends on the rear leg, since this leg is now fully extended. The reach height is measured as the distance between the bottom of the rear leg to the top of the front leg, so it will remain the same for more decreasing slopes. At increasing slopes the reach height decreases. The reach height depends on the tail, since it is now fully extended. The base of the tail however decreases.

3) Payload: TT-Bot was tested on its ability to carry a payload and stand up while doing so. For the payloads of 1, 2, and 3 kg, it was able to do so. For a payload of 4 kg, it was able to go through the transition phase, but collapsed through its rear leg while standing up. We expect that future research can resolve this.

D. Future research

Before a fully quadruped robot is to be designed and build, more experiments can be done with this prototype to further discover its capabilities. Adjustments to the design are suggested to counter the warping of the body. The ends of the upper leg should be made stiffer to counteract the outward bending.

Additionally, changing the actuator control such that the initial position of the actuators no longer needs calibration will make the robots movement more precise.

TT-Bot moves in a pre-programmed path and is not able to correct itself. For future research, the control of the robot can be expanded to make it more agile.

In the end, a design for the whole quadruped robot can be made. A SolidWorks model of what it could look like can be seen in Figure 19.

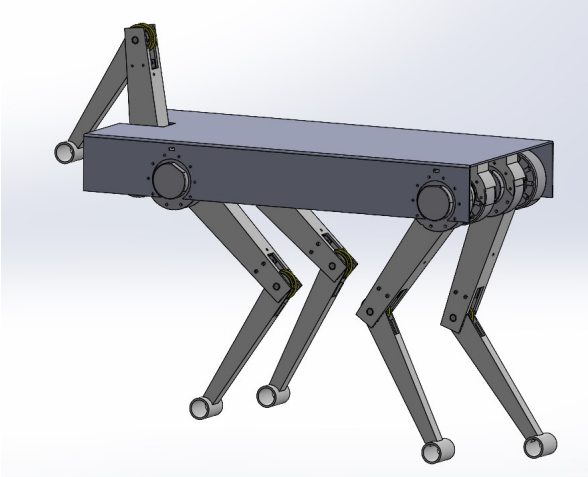


Fig. 19: SolidWorks model of a full quadruped for future research

A fully functional quadruped robot needs extra actuators in the body for hip abduction-adduction. This make the robot more complex, but gives more lateral stability, which may be needed on uneven terrain.

The ultimate research goal is to design a robot monkey. This monkey should be able to stand on its rear legs and climb different stairs and industrial ladders. Therefore more research needs to be done on robotic arms and grasping mechanisms.

V. Conclusion

The goal of this study was to design and develop a prototype for a quadruped robot that is able to perform a quasi-static transition from quadrupedal to upright standing while remaining statically stable in the end position. This was achieved through a lightweight design featuring 3D printed parts and a belt-pulley system, a centralized weight in the robot's body and a tail. This Tail-assisted Transition Robot: TT-Bot uses the tail to maintain static stability during the transition and in final position. Kinematic analysis is used to simulate the movements of TT-Bot. Together with the PID controller, it makes it possible to control TT-Bot in various experiments. TT-Bot has the ability to reach practical heights, while carrying a payload of 3 kg. Therefore, in future research, it is able to carry its own computer system with a battery. It is also able to transition on a slope of 5°. Its performance on steeper slopes and with heavier payloads highlighted some limitations regarding the transition. In general, the results of this work show that this proof-of-concept prototype has successfully met the outlined requirements and constraints, affirming its potential.

References

[1] Manuel Fernando Silva and J. A.Tenreiro MacHado. *A literature review on the optimization of legged robots*. Oct. 2012, pp. 1753–1767. DOI: 10.1177/1077546311403180.

[2] G. Satheesh Kumar et al. "Literature Survey on Four-Legged Robots". In: Springer Science and Business Media Deutschland GmbH, 2021, pp. 691–702. ISBN: 9789811544873. DOI: 10.1007/978-981-15-4488-0_58.

[3] Boston Dynamics. *ABOUT SPOT*. 2023. URL: https : / / dev . bostondynamics.com/docs/concepts/about_spot#about-spot.

[4] Benjamin Katz, Jared Di Carlo, and Sangbae Kim. "Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control". In: 2019. ISBN: 9781538660270.

[5] ANYBotics. *ANYmal Technical Specifications*. 2022. URL: https : // www.anybotics.com/anymal-specifications-sheet/.

[6] Xingxing Wang. *Go2, Unitree Robotics*. 2023. URL: https://m.unitree.com/en/go2/.

[7] Chen Yu and Andre Rosendo. "Multi-Modal Legged Locomotion Framework With Automated Residual Reinforcement Learning". In: *IEEE Robotics and Automation Letters* 7 (4 Oct. 2022), pp. 10312–10319. ISSN: 23773766. DOI: 10.1109/LRA.2022.3191071.

[8] Alexander Dettmann, Daniel Kühn, and Frank Kirchner. "Control of active multi-point-contact feet for quadrupedal locomotion". In: *International Journal of Mechanical Engineering and Robotics Research* 9 (4 2020), pp. 481–488. ISSN: 22780149. DOI: 10.18178/ijmerr.9.4.481-488.

[9] Avular. *Avular — Mobile Robotics*. 2024. URL: https://avular.com/.

[10] MyActuator. *RMD-X8 Pro*. 2024. URL: https://www.myactuator.com/product-page/x8-pro-1-9.

[11] M. Grübler. "Allgemeine Eigenschaften der Zwangläufigen ebenen kinematischen Ketten, Part I." In: (1883).

[12] Serdar Kucuk and Zafer Bingul. "Robot Kinematics: Forward and Inverse Kinematics". In: InTech, 2006, pp. 117–148. ISBN: 3866112858.

[13] J. P. Merlet. *Parallel Robots*. Ed. by G. M. L. Gladwell. Second. Springer, 2006.

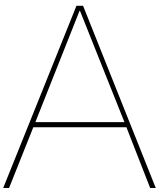
4

Conclusion

This report shows a new design of a prototype for a robot monkey. At first a literature review was performed to get a better insight in the current status of quadruped robots and their capabilities. A concept study was performed in order to choose the best solution for the final design. This final design is presented in the research paper.

In the literature review multiple aspects of a quadruped robot are analysed, such as their gait, mechanical design and actuation. Afterwards existing quadruped robots were divided in categories based on their actuation method and design. These categories are compared on their speed, payload capacity and efficiency. Electrically actuated robots generally achieve higher speeds and efficiency compared to hydraulically actuated ones. In general, robots with larger masses are hydraulically actuated. The literature review served as a reference for developments in quadruped robots. The review also showed that very few existing quadruped robots are able to transition from quadrupedal to a bipedal position while remaining statically stable, showing a research gap in this field.

The goal of the research paper is to present a new design for a quadruped robot, that is able to perform a quasi-static transition from quadrupedal to upright standing and remain statically stable in the end position. This was achieved through a lightweight design featuring 3D printed parts and a belt-pulley system, centralized weight in the robot's body and a tail. This Tail-assisted Transition Robot: (further: TT-Bot) uses its tail to maintain static stability during the transition and in final position. Kinematic analysis is used to simulate the movements of TT-Bot. Together with the PID controller, it makes it possible to control TT-Bot in various experiments. TT-Bot has the ability to reach practical heights, while carrying a payload of 3 kg. Therefore, in future research, it is able to carry its own computer system with a battery. It is also able to transition on a slope of $5°$. Its performance on steeper slopes and with heavier payloads highlighted some limitations regarding the transition. In general, the results of this work show that this proof-of-concept prototype successfully met the outlined requirements and constraints, affirming its potential.



Appendix A - Concept study

Before choosing the final design of the robot, a concept study is performed. The main goal of the research is divided into subgoals. For every subgoal, multiple solutions are presented. All these solutions are weighed, and one solution is chosen for the final design.

A.1. Goals

The main goal of this research is:

Develop a quadruped robot that is able to stand on its rear legs.
This goal can be divided into 3 subgoals:

1. Dynamic quadruped gait
2. Transition between quadruped standing and bipedal standing
3. Statically stable on rear legs

There have been plenty of research efforts on the quadruped gait [10] [11] [12] [13]. Therefore the focus of this research lies on the last two sub goals, the transition and standing on rear legs.

For both subgoals multiple solutions are evaluated.

A.2. Transition

The different methods for transition that were considered are:

- Tuck in and stretch
- Rotate body
- Front legs push off
- Spring in front legs
- Fall backwards

A.2.1. Tuck in and stretch

The robot sinks through its rear legs, shifting its Center of Mass backwards. When CoM is positioned in the back, it can stand up. A schematic overview of the transition can be seen in Figure A.1.

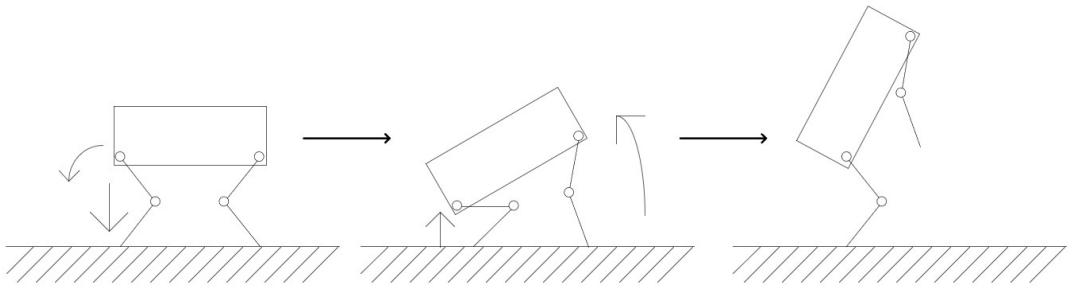


Figure A.1: Tuck in and stretch

A.2.2. Rotate body

The robot rotates its entire body around the rear hip. A schematic overview of the transition can be seen in Figure A.2.

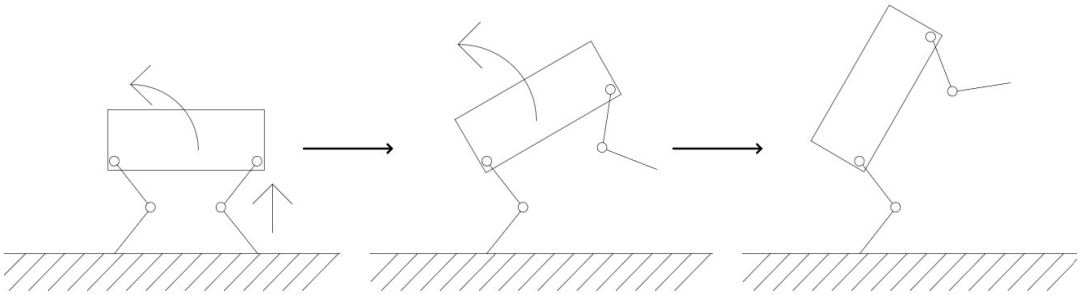


Figure A.2: Caption

A.2.3. Front legs push off

The robot sinks through its front legs. Then it pushes off with its front legs to create a momentum. It uses this momentum to rotate and stand upright. A schematic overview of the transition can be seen in Figure A.3.

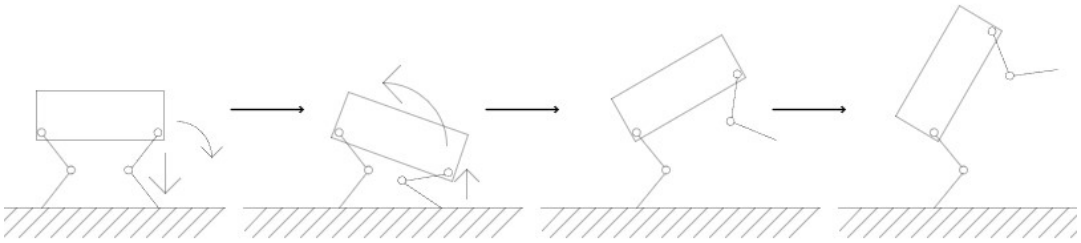


Figure A.3: Caption

A.2.4. Spring in front legs

It uses the same principle as in subsection A.2.3, but a spring is added in the front leg to gain more momentum. A schematic overview of the transition can be seen in Figure A.4.

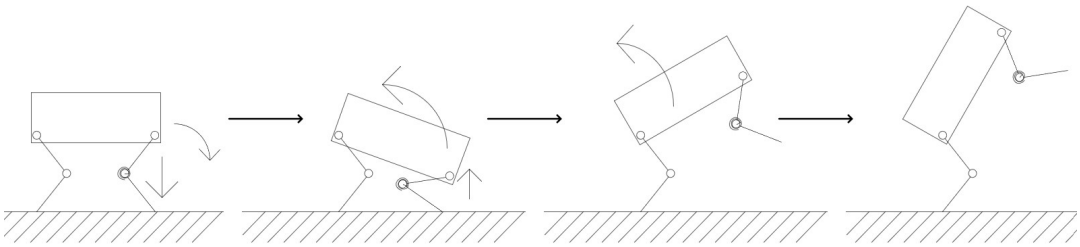


Figure A.4: Caption

A.2.5. Fall backwards

The robots moves its rear leg to the front. When doing so the robot rotates and falls backwards. The robot then moves its rear leg backwards and uses the momentum of the rotation to stand up. A schematic overview of the transition can be seen in Figure A.5.

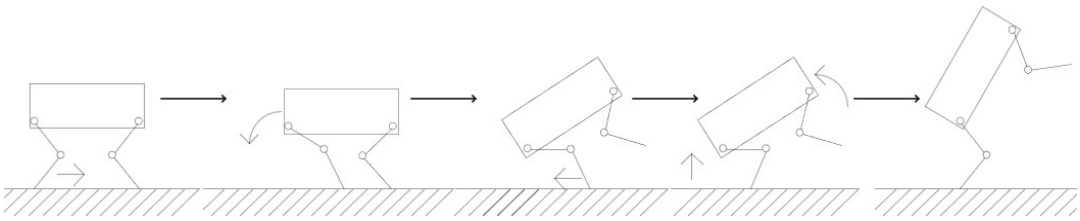


Figure A.5: Caption

The different concepts are graded on the following criteria:

- Simplicity
- Torque needed
- Overshoot
- Range of Motion

Simplicity: How complex is the transition method.
1: Very complex. Extra components and actuators are required.
2: Medium complex. The robot needs to perform multiple different movements.
3: Simple. Only one movement is needed.

Torque needed: How much more torque does the hip actuator need for transition with respect to quadruped walking.
1: A lot more torque is needed for the robot to rotate than needed for walking.
2: Some more torque is needed for the robot to rotate than needed for walking.
3: A bit more torque is needed for the robot to rotate than needed for walking.

Overshoot: The robot rotates too far.
1: The chance of overshoot happening is very high.
2: The chance of overshoot happening is present.
3: The chance of overshoot happening is very low.

Range of motion: Do the joints need more range of motion than for quadruped walking.
1: Joints need much more operating range for transition than needed for walking.
2: Joints need some more operating range for transition than needed for walking.
3: Joints do not need any more operating range for transition than needed for walking.

Table A.1: Assessment of the transition methods

Criteria	Weight	Tuck in	Rotate body	Front legs	Spring	Fall
Simplicity	0.2	2	3	2	1	2
Torque	0.3	3	1	2	2	2
Overshoot	0.25	3	1	2	2	2
Range of Motion	0.25	2	3	3	3	2
Total	**1.0**	**2.6**	**1.9**	**2.3**	**2.1**	**2.0**

From Table A.1 it can be seen that Tuck in is the best transition method. This method is used for the final design.

A.3. Upright Standing

The different methods to perform upright standing that were considered are:

- Tail
- Actuated tail
- Innovative shins [7]
- Foot
- Foldable foot
- Lower leg

A.3.1. Tail

A rigid tail is added to the quadruped robot. With this tail the quadruped robot has three contact points when standing on its rear feet. A schematic overview can be seen in Figure A.6.

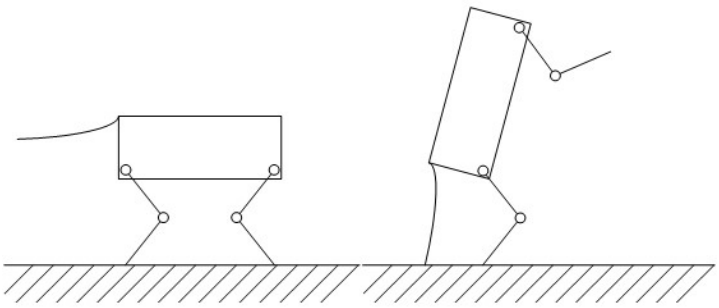


Figure A.6: Quadruped with a tail, left: quadruped stance, right: upright stance

A.3.2. Actuated tail

An actuated tail is added to the quadruped. When not in use this tail is retracted. With this tail the quadruped robot has three contact points when standing on its rear feet. A schematic overview can be seen in Figure A.7.

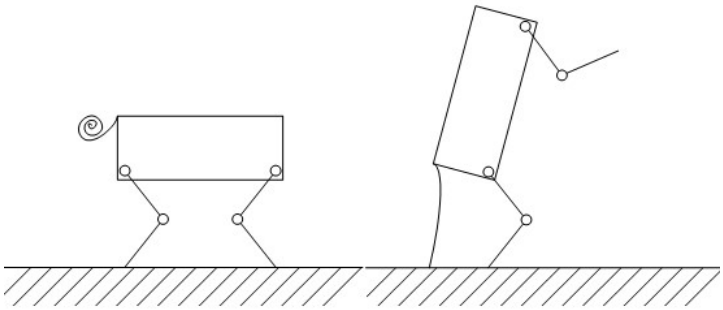


Figure A.7: Quadruped with actuated tail, left: quadruped stance, right: upright stance

A.3.3. Innovative shins

An additional link is added to the lower leg. In the upright stance it forms a ”foot”. A schematic overview can be seen in Figure A.8, this principle is described in the chapter 1.

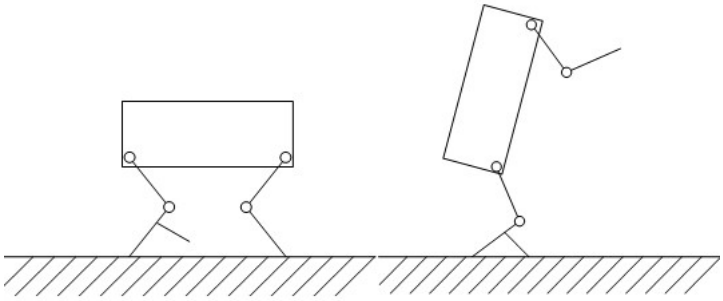


Figure A.8: Quadruped with innovative shins, left: quadruped stance, right: upright stance

A.3.4. Foot

A foot is added to the rear leg of the robot where it can stand on during upright stance. Robot ”Charlie” mentioned in chapter 1 makes use of a foot [8]. A schematic overview can be seen in Figure A.9.

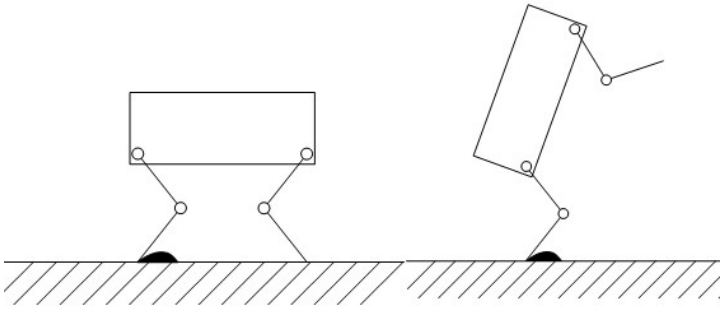


Figure A.9: Quadruped with a foot, left: quadruped stance, right: upright stance

A.3.5. Fold-able foot

An expandable foot is added to the rear leg of the robot where it can stand on during upright stance. When not in use this foot is retracted in the lower leg. A schematic overview can be seen in Figure A.10.

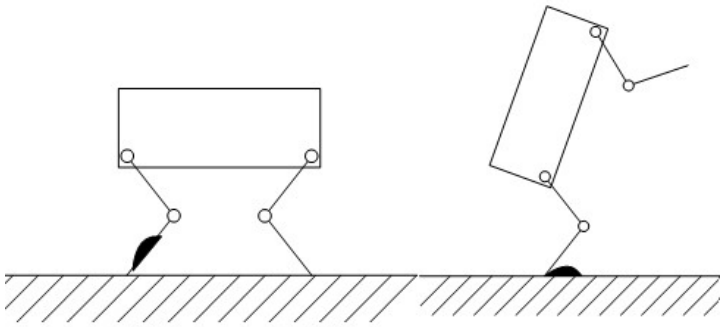


Figure A.10: Quadruped with fold-able feet, left: quadruped stance, right: upright stance

The different concepts are graded on the following criteria:

- Simplicity
- Interference with quadruped walking
- Stability
- Mobility

Simplicity: How complex is the design. A simple design is appreciated.
1: Very complex. Mechanism makes the whole system more complex, not only for bipedal standing but also for walking. Extra link, joint, actuator.
2: Medium complex. Additional actuator is needed only for bipedal standing.
3: Simple. Little adjustments on a quadruped need to be done for it to work. No additional actuators.

Interference with walking: Does the design interfere with quadruped walking. Could the extra part or other design make it harder to walk on four feet.
1: A lot of interference with quadruped walking.
2: Could have some interference with quadruped walking.
3: No interference at all.

Stability: How stable can the robot stand on its rear feet. How big is the support polygon.
1: Small support polygon. $< 100E3mm^2$
2: Medium support polygon. $100E3mm^2 < x < 150E3mm^2$
3: Large support polygon. $> 150E3mm^2$

Manoeuvrability: Can the robot perform different tasks? (walking on two feet)
1: Robot is only able to stand on two feet and can't perform any tasks.
2: Robot would be able to move a little bit or with small steps.
3: Robot would be able to walk on two feet.

Table A.2: Assessment of the methods for upright standing

Criteria	Weight	Tail	Actuated tail	Shin	Foot	Fold-able foot	Lower leg
Simplicity	0.2	3	2	3	3	2	1
Quadruped walking	0.3	2	3	1	1	3	3
Stability	0.3	3	3	2	1	1	2
Manoeuvrability	0.2	2	3	2	3	3	3
Total	**1**	**2.4**	**2.8**	**1.9**	**1.8**	**2.2**	**2.3**

From Table A.2 it can be seen that the actuated tail is the best method for upright standing. This method is used in the final design.

A.4. Final design

The transition method “Tuck in” and method for upright standing “Actuated tail” are chosen and need to be integrated in one final design. Still there are multiple options for an actuated tail. Three different options can be seen in Figure A.11, Figure A.12 and Figure A.13. The pantographic tail design is based on the legs of MIT Cheetah [14]. This design can be seen in chapter 2.

(a) Prismatic tail is retracted in transition phase

(b) Prismatic tail is extended in upright position

Figure A.11: Robot with prismatic tail

(a) Pantographic tail is retracted in transition phase

(b) Pantographic tail is extended in upright position

Figure A.12: Robot with pantographic tail

(a) Tail is retracted in transition phase

(b) Tail is extended in upright position

Figure A.13: Robot with articulated tail

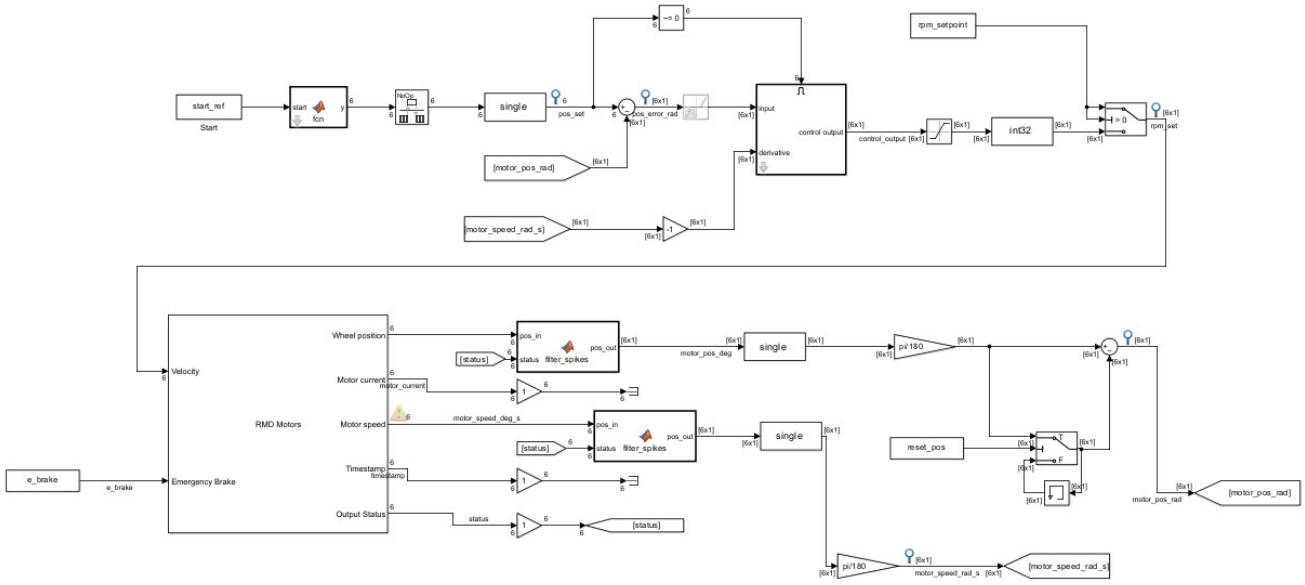
The articulated tail design was selected, eliminating the need for two separate designs and allowing for a single design to be created for the legs with minor adjustments for the tail. Consequently, the tail functions similarly to the legs, utilizing the same components and sharing the control system. This simplifies the overall robot design significantly.

A.5. Material

The choice of material significantly impacts the final design. Two different options were considered for fabricating the legs and tail. The first option involves using rectangular aluminum tubes finished with laser cutting. The second option is to 3D print the legs, offering a wider range of design possibilities and significantly lower costs. The decisive factor ultimately was the cost, with 3D printing being much cheaper than laser-cut aluminum tubes. Additionally, for the pulleys in the hips and knees, 3D printing was chosen over ordering, as the latter option incurred significantly higher costs.

B

Appendix B - Simulink model



C

Appendix C - PID controller

The RMD motors incorporate an internal PID controller, initially set to factory defaults. The self-made PID controller, used to convert positional angles into rpm inputs, underwent fine-tuning to ensure the robot's smooth transition into motion. This was achieved by analyzing the input trajectory, output trajectory of motor angles and their associated errors. Figure C.1 illustrates two graphs depicting the input and output data of Motor 1 and Motor 2 for alternative PID values.

Figure C.1: Motor input and output signals. Green line is the input angle, yellow line is the output angle and the red line is the error. This is shown for motor 1, top graph and motor 2, bottom graph

D

Appendix D - MATLAB Code

D.1. Main code

```
clear all; close all; clc

% Manipulator parameters
l1 = 265; % mm, length of the lower links
l2 = 238.99; % mm, length of the upper links
L = 480; % mm, length between the rear leg ”hip joint” and front leg ”hip joint”
t1 = 245; % mm, length of lower tail
t2 = 202.99; % mm, length of upper tail
T = 110; % mm, length between tail ”hip joint” to rear leg ”hip joint”
COM_x = 145; % mm, x_length between rear leg ”hip joint” and CoM
COM_y = 7; % mm, y_length between rear leg ”hip joint” and CoM
B0 = 480; % mm, length between rear foot and front foot
B1 = 470; % mm, length between rear foot and tail foot

% Base joints positions
angle = 0;

x_base_rear = 0;
y_base_rear = B1*sind(angle);
x_base_front = B0*cosd(angle);
y_base_front = y_base_rear + B0*sind(angle);
x_base_tail = -B1*cosd(angle);
y_base_tail = 0;
x_base_COM = COM_x*cosd(angle);
y_base_COM = y_base_rear + COM_x*sind(angle);

dx = sind(angle)*(l1+l2+COM_y);
dy = cosd(angle)*(l1+l2+COM_y);
dx1 = sind(angle)*(350);
dy1 = cosd(angle)*(350);
%% Path Trajectory
% Path of COM:
x_values = [x_base_COM-dx x_base_COM-dx1 50 -50 -COM_y-175  -3/8*B1-COM_y];
y_values = [y_base_COM+dy y_base_COM+dy1 y_base_COM+270 y_base_COM+280 COM_x+325 COM_x+430];
phi_values = [0+angle 0+angle 24+angle 26+angle 89 89];

% Values needed for Slope stability test
x_COM_end = x_values(end);
y_COM_end = y_values(end);

% Positions of x, y and phi for the whole path
[x_pos, y_pos, phi_pos] = ShortPathTrajectory(x_values, y_values, phi_values);
phi_pos = deg2rad(phi_pos);

% Number of positions
N = size(x_pos, 1);

% Pre-allocate arrays for results
```

```
x_elbow_rears = NaN(N, 1);
y_elbow_rears = NaN(N, 1);
x_elbow_fronts = NaN(N, 1);
y_elbow_fronts = NaN(N, 1);
x_elbow_tails = NaN(N, 1);
y_elbow_tails = NaN(N, 1);
theta11s = NaN(N, 1);
theta12s = NaN(N, 1);
theta21s = NaN(N, 1);
theta22s = NaN(N, 1);
theta31s = NaN(N, 1);
theta32s = NaN(N, 1);

% Pre-allocate arrays for results
thetaM1s = NaN(N, 1); % Angle of motor 1
thetaM2s = NaN(N, 1); % Angle of motor 2
thetaM3s = NaN(N, 1); % Angle of motor 3
thetaM4s = NaN(N, 1); % Angle of motor 4
thetaM5s = NaN(N, 1); % Angle of motor 5
thetaM6s = NaN(N, 1); % Angle of motor 6

% Pre-allocate arrays for the body positions
x_rears = NaN(N, 1);
y_rears = NaN(N, 1);
x_fronts = NaN(N, 1);
y_fronts = NaN(N, 1);
x_tails = NaN(N, 1);
y_tails = NaN(N, 1);
x_COMs = NaN(N, 1);
y_COMs = NaN(N, 1);

% Arrays for foot joints
x_base_rears = ones(N, 1) * x_base_rear;
y_base_rears = ones(N, 1) * y_base_rear;
x_base_fronts = ones(N, 1) * x_base_front;
y_base_fronts = ones(N, 1) * y_base_front;
x_base_tails = ones(N, 1) * x_base_tail;
y_base_tails = ones(N, 1) * y_base_tail;

%% Maximum slope possible

[alpha, beta] = COM_angle(x_COM_end, y_COM_end, x_base_rear, y_base_rear, x_base_tail,
    y_base_tail);
%%
% Loop through each position
for i = 1:N

    % Initialize variables
    x_elbow_rear = NaN; y_elbow_rear = NaN;
    x_elbow_front = NaN; y_elbow_front = NaN;
    x_elbow_tail = NaN; y_elbow_tail = NaN;
    theta11 = NaN; theta12 = NaN;
    theta21 = NaN; theta22 = NaN;
    theta31 = NaN; theta32 = NaN;
    thetaM1 = NaN; thetaM2 = NaN;
    thetaM3 = NaN; thetaM4 = NaN;
    thetaM5 = NaN; thetaM6 = NaN;

    % Extract current position and phi
    x = x_pos(i);
    y = y_pos(i);
    phi = phi_pos(i);

    % % Platform endpoints based on desired position and orientation
    % Center of Mass remains at (x, y)
    x_COM = x;
    y_COM = y;
    % Rear
    x_rear = x - COM_x * cos(phi) + COM_y * sin(phi);
    y_rear = y - COM_x * sin(phi) - COM_y * cos(phi);
    % Front
```

```
119     x_front = x + (L - COM_x)*cos(phi) + COM_y * sin(phi);
120     y_front = y + (L - COM_x)*sin(phi) - COM_y * cos(phi);
121     % Tail
122     x_tail = x - (COM_x + T) * cos(phi) + COM_y * sin(phi);
123     y_tail = y - (COM_x + T) * sin(phi) - COM_y * cos(phi);
124
125     % Solve position for rear limb
126     try
127         [x_elbow_rear, y_elbow_rear] = Elbowposition(x_base_rear, y_base_rear, x_rear, y_rear
                , l1, l2, 'max');
128         if isempty(x_elbow_rear) || isempty(y_elbow_rear)
129             error('No solution found.');
130         end
131         theta11 = calculateTheta1(x_base_rear, y_base_rear, x_elbow_rear, y_elbow_rear);
132         theta12 = calculateTheta2(x_base_rear, y_base_rear, x_rear, y_rear, l1, l2);
133         thetaM1 = CalculateThetam1(x_base_rear, y_base_rear, x_elbow_rear, y_elbow_rear, phi,
                 'rear');
134         thetaM2 = CalculateThetam2(x_rear, y_rear, x_elbow_rear, y_elbow_rear, phi, 'rear');
135     catch
136
137     end
138
139 % Repeat similar try-catch blocks for solving the position for front limb and tail
140     try
141         [x_elbow_front, y_elbow_front] = Elbowposition(x_base_front, y_base_front, x_front,
                y_front, l1, l2, 'min');
142         if isempty(x_elbow_front) || isempty(y_elbow_front)
143             error('No solution found.');
144         end
145         theta21 = calculateTheta1(x_base_front, y_base_front, x_elbow_front, y_elbow_front);
146         theta22 = calculateTheta2(x_base_front, y_base_front, x_front, y_front, l1, l2);
147         thetaM3 = CalculateThetam1(x_base_front, y_base_front, x_elbow_front, y_elbow_front,
                phi, 'front');
148         thetaM4 = CalculateThetam2(x_front, y_front, x_elbow_front, y_elbow_front, phi, '
                front');
149     catch
150
151     end
152
153     try
154         [x_elbow_tail, y_elbow_tail] = Elbowposition(x_base_tail, y_base_tail, x_tail, y_tail
                , t1, t2, 'min');
155         if isempty(x_elbow_tail) || isempty(y_elbow_tail)
156             error('No solution found.');
157         end
158         theta31 = calculateTheta1(x_base_tail, y_base_tail, x_elbow_tail, y_elbow_tail);
159         theta32 = calculateTheta2(x_base_tail, y_base_tail, x_tail, y_tail, t1, t2);
160         thetaM5 = CalculateThetam1(x_base_tail, y_base_tail, x_elbow_tail, y_elbow_tail, phi,
                 'tail');
161         thetaM6 = CalculateThetam2(x_tail, y_tail, x_elbow_tail, y_elbow_tail, phi, 'tail');
162     catch
163
164     end
165
166     % Store the results
167     x_elbow_rears(i) = x_elbow_rear;
168     y_elbow_rears(i) = y_elbow_rear;
169     x_elbow_fronts(i) = x_elbow_front; % After handling similar for front limb
170     y_elbow_fronts(i) = y_elbow_front; % After handling similar for front limb
171     x_elbow_tails(i) = x_elbow_tail; % After handling similar for tail
172     y_elbow_tails(i) = y_elbow_tail; % After handling similar for tail
173     theta11s(i) = theta11;
174     theta12s(i) = theta12;
175     theta21s(i) = theta21; % After calculations
176     theta22s(i) = theta22; % After calculations
177     theta31s(i) = theta31; % After handling similar for tail
178     theta32s(i) = theta32; % After handling similar for tail
179
180     %Store the results for motor angles
181     thetaM1s(i) = thetaM1;
182     thetaM2s(i) = thetaM2;
```

```
    thetaM3s(i) = thetaM3;
    thetaM4s(i) = thetaM4;
    thetaM5s(i) = thetaM5;
    thetaM6s(i) = thetaM6;

    % Store resuls for hip joints and COM locations
    x_rears(i) = x_rear;
    y_rears(i) = y_rear;
    x_fronts(i) = x_front;
    y_fronts(i) = y_front;
    x_tails(i) = x_tail;
    y_tails(i) = y_tail;
    x_COMs(i) = x_COM;
    y_COMs(i) = y_COM;

    % % Optionally, print the results for each position
    % fprintf('Position %d: x = %d, y = %d, phi = %d\n', i, x, y, rad2deg(phi));
    % fprintf('Theta11: %f, Theta12: %f\n\n', theta11, theta12);
    % fprintf('Theta21: %f, Theta22: %f\n\n', theta21, theta22);
    % fprintf('Theta31: %f, Theta32: %f\n\n', theta31, theta32);
end

%% Path Trajectory 2

% Path Trajectory of "Quadruped stance"
% 3 stappen, end at 1001

ThetaM5_values = [0 -0.584685 thetaM5s(1001)];
ThetaM6_values = [0 pi thetaM6s(1001)];

[ThetaM5_extra, thetaM6_extra] = ShortExtraTrajectory(ThetaM5_values, ThetaM6_values);

for i = 1:size(ThetaM5_extra)
    thetaM5s(i) = ThetaM5_extra(i);
    thetaM6s(i) = thetaM6_extra(i);

    [x_elbow_tails(i), y_elbow_tails(i), x_base_tails(i), y_base_tails(i)] =
        calculateTailKinematics(phi_pos(i), thetaM5s(i), thetaM6s(i), t1, t2, x_tails(i),
        y_tails(i));
    distanceBaseToElbow = sqrt((x_elbow_tails(i) - x_tails(i))^2 + (y_elbow_tails(i) -
        y_tails(i))^2);
    distanceElbowToFoot = sqrt((x_base_tails(i) - x_elbow_tails(i))^2 + (y_base_tails(i) -
        y_elbow_tails(i))^2);
    tolerance = 0.00001;

    if distanceElbowToFoot - tolerance > t1 || distanceBaseToElbow - tolerance > t2
        error('No solution')
    end
end

%%
% Path Trajectory of "Tripod stance"
% 3 stappen, begin at 1502
ThetaM3_values = [thetaM3s(1502) -0.35 0.5*pi];
ThetaM4_values = [thetaM4s(1502) 1.2 -0.8];

[thetaM3_extra, thetaM4_extra] = ShortExtraTrajectory(ThetaM3_values, ThetaM4_values);

for i = 1:size(thetaM3_extra)
    thetaM3s(1504+i) = thetaM3_extra(i);
    thetaM4s(1504+i) = thetaM4_extra(i);

    [x_elbow_fronts(1504+i), y_elbow_fronts(1504+i), x_base_fronts(1504+i), y_base_fronts
        (1504+i)] = calculateTailKinematics(phi_pos(1504+i), thetaM3_extra(i), thetaM4_extra(
        i), l1, l2, x_fronts(1504+i), y_fronts(1504+i));
    distanceBaseToElbow = sqrt((x_elbow_fronts(1504+i) - x_fronts(1504+i))^2 + (
        y_elbow_fronts(1504+i) - y_fronts(1504+i))^2);
    distanceElbowToFoot = sqrt((x_base_fronts(1504+i) - x_elbow_fronts(1504+i))^2 + (
        y_base_fronts(1504+i) - y_elbow_fronts(1504+i))^2);
    tolerance = 0.00001;

```

```
246     if distanceElbowToFoot - tolerance > l1 || distanceBaseToElbow - tolerance > l2
247         error('No solution')
248     end
249 
250 end
251 %% Check if thetaM6 exceeds fysical boundary
252 
253 thetaM6max = 3.4; % Tail can't go further than this in the body
254 
255 for i = 1:length(thetaM6s)
256     if thetaM6s(i) > thetaM6max
257         error('ThetaM6 exceeds the maximum allowable angle of %f radians')
258     end
259 end
260 
261 % Make sure the first value of motor angles is exactly 0 (not 0.00003 bv)
262 thetaM1s(1,1) = 0;
263 thetaM2s(1,1) = 0;
264 thetaM3s(1,1) = 0;
265 thetaM4s(1,1) = 0;
266 thetaM5s(1,1) = 0;
267 thetaM6s(1,1) = 0;
268 
269 save("C:\Users\tomku\OneDrive - Delft University of Technology\Master HTE\Thesis\MATLAB\Path
        trajectory\Motorhoeken_(0_0kg).mat", "thetaM1s", "thetaM2s", "thetaM3s", "thetaM4s", "
        thetaM5s", "thetaM6s");
270 %% Visualization video of movement
271 
272 % Setup the figure
273 figure;
274 hold on;
275 grid on;
276 axis equal;
277 xlabel('X Coordinate');
278 ylabel('Y Coordinate');
279 title('Simulation');
280 
281 % Set axex limits
282 xlim([-500, 500]);
283 ylim([0, 1600]);
284 
285 % Initial plot elements for robot body
286 h_tail = plot(x_tails(1), y_tails(1), 'ro');  % Tail position marker
287 h_rear = plot(x_rears(1), y_rears(1), 'ro'); % Rear position marker
288 h_front = plot(x_fronts(1), y_fronts(1), 'ro');  % Front position marker
289 h_line_body = line([x_tails(1), x_fronts(1)], [y_tails(1), y_fronts(1)], 'Color', 'k');  %
        Body link
290 h_COM = plot(x_COMs(1), y_COMs(1), 'gx', 'MarkerSize', 8);  % COM marker
291 
292 % Initial plot elements floor
293 p_line = line([x_base_tail, x_base_front], [y_base_tail, y_base_front], 'Color', 'k'); %
        Ground
294 
295 % Initial plot elements for tail
296 h_elbow_tail = plot(x_elbow_tails(1), y_elbow_tails(1), 'ro');  % Elbow position marker
297 h_foot_tail = plot(x_base_tails(1), y_base_tails(1), 'ro');  % Foot position marker
298 h_line_tail1 = line([x_tails(1), x_elbow_tails(1)], [y_tails(1), y_elbow_tails(1)], 'Color',
        'k');  % Line from tail to elbow
299 h_line_tail2 = line([x_elbow_tails(1), x_base_tails(1)], [y_elbow_tails(1), y_base_tails(1)],
         'Color', 'k');  % Line from elbow to foot
300 
301 % Initial plot elements for front leg
302 h_elbow_front = plot(x_elbow_fronts(1), y_elbow_fronts(1), 'ro');  % Elbow position marker
303 h_foot_front = plot(x_base_fronts(1), y_base_fronts(1), 'ro');  % Foot position marker
304 h_line_front1 = line([x_fronts(1), x_elbow_fronts(1)], [y_fronts(1), y_elbow_fronts(1)], '
        Color', 'k');  % Line from tail to elbow
305 h_line_front2 = line([x_elbow_fronts(1), x_base_fronts(1)], [y_elbow_fronts(1), y_base_fronts
        (1)], 'Color', 'k');  % Line from elbow to foot
306 
307 % Initial plot elements for rear leg
308 h_elbow_rear = plot(x_elbow_rears(1), y_elbow_rears(1), 'ro');  % Rear elbow position marker
```

```
h_foot_rear = plot(x_base_rears(1), y_base_rears(1), 'ro');  % Rear foot position marker
h_line_rear1 = line([x_rears(1), x_elbow_rears(1)], [y_rears(1), y_elbow_rears(1)], 'Color', 'k');  % Line from tail to rear elbow
h_line_rear2 = line([x_elbow_rears(1), x_base_rears(1)], [y_elbow_rears(1), y_base_rears(1)], 'Color', 'k');  % Line from rear elbow to foot

% Create a VideoWriter object to record the video
outputVideo = VideoWriter('simulation_video.avi', 'Motion JPEG AVI');
outputVideo.FrameRate = 20; % Specify the frame rate
open(outputVideo);

% Animate the movement
for k = 1:length(x_tails)
    % Update body positions
    set(h_tail, 'XData', x_tails(k), 'YData', y_tails(k));
    set(h_front, 'XData', x_fronts(k), 'YData', y_fronts(k));
    set(h_rear, 'XData', x_rears(k), 'YData', y_rears(k));
    set(h_line_body, 'XData', [x_tails(k), x_fronts(k)], 'YData', [y_tails(k), y_fronts(k)]);
    set(h_COM, 'XData', x_COMs(k), 'YData', y_COMs(k));

    % Update tail positions
    set(h_elbow_tail, 'XData', x_elbow_tails(k), 'YData', y_elbow_tails(k));
    set(h_foot_tail, 'XData', x_base_tails(k), 'YData', y_base_tails(k));
    set(h_line_tail1, 'XData', [x_tails(k), x_elbow_tails(k)], 'YData', [y_tails(k), y_elbow_tails(k)]);
    set(h_line_tail2, 'XData', [x_elbow_tails(k), x_base_tails(k)], 'YData', [y_elbow_tails(k), y_base_tails(k)]);

    % Update front leg positions
    set(h_elbow_front, 'XData', x_elbow_fronts(k), 'YData', y_elbow_fronts(k));
    set(h_foot_front, 'XData', x_base_fronts(k), 'YData', y_base_fronts(k));
    set(h_line_front1, 'XData', [x_fronts(k), x_elbow_fronts(k)], 'YData', [y_fronts(k), y_elbow_fronts(k)]);
    set(h_line_front2, 'XData', [x_elbow_fronts(k), x_base_fronts(k)], 'YData', [y_elbow_fronts(k), y_base_fronts(k)]);

    % Update rear leg positions
    set(h_elbow_rear, 'XData', x_elbow_rears(k), 'YData', y_elbow_rears(k));
    set(h_foot_rear, 'XData', x_base_rears(k), 'YData', y_base_rears(k));
    set(h_line_rear1, 'XData', [x_rears(k), x_elbow_rears(k)], 'YData', [y_rears(k), y_elbow_rears(k)]);
    set(h_line_rear2, 'XData', [x_elbow_rears(k), x_base_rears(k)], 'YData', [y_elbow_rears(k), y_base_rears(k)]);

    if mod(k, 5) == 0
        drawnow;
        frame = getframe(gcf); % Capture the frame
        writeVideo(outputVideo, frame); % Write the frame to the video
    end
end
close(outputVideo); % Close the video file
hold off;

%% Motion diagram

% Setup the figure
figure;
hold on;
grid on;
axis equal;
xlabel('X Coordinate');
ylabel('Y Coordinate');
title('Robot Movement Simulation');

% Set axes limits
xlim([-500, 500]);
ylim([0, 1500]);

% Define key frames as per your request
key_frames = [1, 600, 1500, 1800, 2505];

```

```
% Define grayscale colors for the key frames
colors = [0.9 0.9 0.9; 0.7 0.7 0.7; 0.5 0.5 0.5; 0.3 0.3 0.3; 0.1 0.1 0.1]; % Dark grey, grey
    , light grey

% Create plot handles for the legend
h_key_frames = [];

for i = 1:length(key_frames)
    k = key_frames(i);
    % Plot body
    plot([x_tails(k), x_fronts(k)], [y_tails(k), y_fronts(k)], 'Color', colors(i,:), '
        LineWidth', 2);
    plot(x_tails(k), y_tails(k), 'o', 'MarkerFaceColor', colors(i,:), 'MarkerEdgeColor',
        colors(i,:));  % Tail position marker
    plot(x_fronts(k), y_fronts(k), 'o', 'MarkerFaceColor', colors(i,:), 'MarkerEdgeColor',
        colors(i,:));  % Front position marker
    h_COM = plot(x_COMs(k), y_COMs(k), 'x', 'Color', colors(i,:), 'MarkerSize', 10, '
        LineWidth', 2);  % COM marker

    % Plot tail leg
    plot([x_tails(k), x_elbow_tails(k)], [y_tails(k), y_elbow_tails(k)], 'Color', colors(i,:)
        , 'LineWidth', 2);
    plot(x_elbow_tails(k), y_elbow_tails(k), 'o', 'MarkerFaceColor', colors(i,:), '
        MarkerEdgeColor', colors(i,:));  % Tail elbow joint
    plot([x_elbow_tails(k), x_base_tails(k)], [y_elbow_tails(k), y_base_tails(k)], 'Color',
        colors(i,:), 'LineWidth', 2);
    plot(x_base_tails(k), y_base_tails(k), 'o', 'MarkerFaceColor', colors(i,:), '
        MarkerEdgeColor', colors(i,:));  % Tail foot joint

    % Plot front leg
    plot([x_fronts(k), x_elbow_fronts(k)], [y_fronts(k), y_elbow_fronts(k)], 'Color', colors(
        i,:), 'LineWidth', 2);
    plot(x_elbow_fronts(k), y_elbow_fronts(k), 'o', 'MarkerFaceColor', colors(i,:), '
        MarkerEdgeColor', colors(i,:));  % Front elbow joint
    plot([x_elbow_fronts(k), x_base_fronts(k)], [y_elbow_fronts(k), y_base_fronts(k)], 'Color
        ', colors(i,:), 'LineWidth', 2);
    plot(x_base_fronts(k), y_base_fronts(k), 'o', 'MarkerFaceColor', colors(i,:), '
        MarkerEdgeColor', colors(i,:));  % Front foot joint

    % Plot rear leg
    plot([x_rears(k), x_elbow_rears(k)], [y_rears(k), y_elbow_rears(k)], 'Color', colors(i,:)
        , 'LineWidth', 2);
    plot(x_rears(k), y_rears(k), 'o', 'MarkerFaceColor', colors(i,:), 'MarkerEdgeColor',
        colors(i,:));  % Rear hip joint
    plot(x_elbow_rears(k), y_elbow_rears(k), 'o', 'MarkerFaceColor', colors(i,:), '
        MarkerEdgeColor', colors(i,:));  % Rear elbow joint
    plot([x_elbow_rears(k), x_base_rears(k)], [y_elbow_rears(k), y_base_rears(k)], 'Color',
        colors(i,:), 'LineWidth', 2);
    plot(x_base_rears(k), y_base_rears(k), 'o', 'MarkerFaceColor', colors(i,:), '
        MarkerEdgeColor', colors(i,:));  % Rear foot joint

    % Store the handle for the first plot of each key frame
    if i == 1
        h_key_frames = [h_key_frames, plot(nan, nan, 'Color', colors(i,:), 'LineWidth', 2)];
    elseif i == 2
        h_key_frames = [h_key_frames, plot(nan, nan, 'Color', colors(i,:), 'LineWidth', 2)];
    elseif i == 3
        h_key_frames = [h_key_frames, plot(nan, nan, 'Color', colors(i,:), 'LineWidth', 2)];
    elseif i == 4
        h_key_frames = [h_key_frames, plot(nan, nan, 'Color', colors(i,:), 'LineWidth', 2)];
    elseif i ==5
        h_key_frames = [h_key_frames, plot(nan, nan, 'Color', colors(i,:), 'LineWidth', 2)];
    end
end

% Add legend
legend([h_key_frames, h_COM], {'"Quadruped" stance', '"Quadruped" stance', '"Transition"
    stance', '"Upright" stance', '"Upright" stance', 'Center of Mass'}, ...
       'Location', 'best');

hold off;
```

```
%
% % % Save the image
% % saveas(gcf, 'robot_motion_diagram.png');
%
%
%
```

D.2. Function to generate path trajecory of CoM

```
function [x, y, phi] = ShortPathTrajectory(x_values, y_values, phi_values)
    wpts = [x_values; y_values; phi_values];
    tpts = 0:(size(x_values, 2)-1);

    % Custom time vector
    % Default step size
    defaultStep = 0.002;
    % Initialize time vector
    tvec = [];
    % Loop through each segment
    for i = 1:length(tpts)-1
        stepSize = defaultStep;
        tvec = [tvec, tpts(i):stepSize:tpts(i+1)];
    end

    % Ensure the last point is included
    if tvec(end) ~= tpts(end)
        tvec = [tvec tpts(end)];
    end

    % Cubic polynomial trajectory interpolation
    [q, qd, qdd, ppval] = cubicpolytraj(wpts, tpts, tvec);

    x = transpose(q(1,:));
    y = transpose(q(2,:));
    phi = transpose(q(3,:));

    % Define the indices for every 5th point
    point_indices = 1:10:length(q(1,:));

    % Indices for arrows, selecting every third point from the already selected points
    arrow_indices = point_indices(1:10:end);

    % Calculate the components of the direction vectors
    arrowLength = 20; % Adjust this value based on your scale
    u = arrowLength * cosd(q(3, arrow_indices)); % Change in x, based on the angle
    v = arrowLength * sind(q(3, arrow_indices)); % Change in y, based on the angle

    % Normalize vectors to have the same length
    for i = 1:length(u)
        norm = sqrt(u(i)^2 + v(i)^2);
        u(i) = arrowLength * (u(i) / norm); % Normalize and scale
        v(i) = arrowLength * (v(i) / norm); % Normalize and scale
    end

    % Initialize arrays for segments
    seg1_points = [];
    seg2_points = [];
    seg3_points = [];

    % Colors for the segments
    colors = {'g', 'b', 'c'};

    % Figure for path trajectory of CoM
    figure;
    hold on;

```

```
    % Segment the trajectory points
    for i = 1:length(tpts)-1
        segment_indices = find(tvec >= tpts(i) & tvec <= tpts(i+1));
        seg_point_indices = intersect(segment_indices, point_indices);

        % Allocate points to respective segment arrays
        if i <= 2 % First two segments
            seg1_points = [seg1_points, q(:, seg_point_indices)];
        elseif i == 3 % Third segment
            seg2_points = q(:, seg_point_indices);
        else % Last two segments
            seg3_points = [seg3_points, q(:, seg_point_indices)];
        end
    end

    % Plotting segments with different colors
    if ~isempty(seg1_points)
        plot(seg1_points(1,:), seg1_points(2,:), ['.' colors{1}], 'MarkerSize', 10, '
            DisplayName', '"Quadruped" stance');
    end
    if ~isempty(seg2_points)
        plot(seg2_points(1,:), seg2_points(2,:), ['.' colors{2}], 'MarkerSize', 10, '
            DisplayName', '"Transition" stance');
    end
    if ~isempty(seg3_points)
        plot(seg3_points(1,:), seg3_points(2,:), ['.' colors{3}], 'MarkerSize', 10, '
            DisplayName', '"Upright" stance');
    end

    % Plot waypoints
    plot(wpts(1,:), wpts(2,:), 'or', 'DisplayName', 'Waypoints');
    % Add arrows for orientation
    orientationArrows = quiver(q(1,arrow_indices), q(2,arrow_indices), u, v, 0, 'r', '
        LineWidth', 1, 'DisplayName', 'Orientation of body');


    xlabel('X');
    ylabel('Y');
    title('Trajectory of CoM with orientation');
    grid on;
    legend('show', 'Location', 'best');
    axis equal;
    xlim([-200, 200]);
    ylim([200, 650]);
    hold off;
end
```

D.3. Function to calculate knee joint locations

```
function [x_elbow, y_elbow] = Elbowposition(x_base, y_base, x_target, y_target, l1, l2,
    criterion)

% Relative position of target point to base joint
    dx = x_target - x_base;
    dy = y_target - y_base;

    % Distance from base joint to target point
    r = sqrt(dx^2 + dy^2);

    if r > l1+l2  % Checking if target is reachable
        x_elbow = NaN;
        y_elbow = NaN;
        return;
    end

    a = (l1^2 - l2^2 + r^2) / (2*r);
    h_sq = l1^2 - a^2;
    h = sqrt(h_sq);

```

```
    % Coordinates of point P
    x_p = x_base + a*dx/r;
    y_p = y_base + a*dy/r;

    % Calculate coordinate of the knee joint
    if strcmp(criterion, 'max') % Location for rear leg
        x_elbow = x_p + h*dy/r;
        y_elbow = y_p - h*dx/r;
    else
        x_elbow = x_p - h*dy/r; % Location for front leg and tail
        y_elbow = y_p + h*dx/r;
    end
end
```

D.4. Function to calculate motor angles

```
function thetam1 = CalculateThetam1(x_base, y_base, x_elbow, y_elbow, phi, criterion)

if strcmp(criterion, 'rear')
    dx = x_elbow - x_base;
    dy = y_elbow - y_base;
    thetam1 = atan2(dx, dy) + phi;
else if strcmp(criterion, 'front')
    dx = x_base - x_elbow;
    dy = y_elbow - y_base;
    thetam1 = atan2(dx, dy) - phi;
else
    dx = x_base - x_elbow;
    dy = y_elbow - y_base;
    thetam1 = atan2(dx, dy) - phi;
end
end
```

```
function thetam2 = CalculateThetam2(x, y, x_elbow, y_elbow, phi, criterion)

if strcmp(criterion, 'rear')
    dx = x_elbow - x;
    dy = y - y_elbow;
    thetam2 = atan2(dx, dy) - phi;
else if strcmp(criterion, 'front')
    dx = x - x_elbow;
    dy = y - y_elbow;
    thetam2 = atan2(dx, dy) + phi;
else
    dx = x - x_elbow;
    dy = y - y_elbow;
    thetam2 = atan2(dx, dy) + phi;
end
end
```

D.5. Function to generate extra path trajectory for tail and front leg

```
function [ThetaM1, ThetaM2] = ShortExtraTrajectory(ThetaM1_values, ThetaM2_values)
    wpts = [ThetaM1_values; ThetaM2_values];
    tpts = 0:(size(ThetaM1_values, 2)-1);

    tvec= 0:0.002:(size(ThetaM1_values, 2)-1);

    [q, qd, qdd, ppval] = cubicpolytraj(wpts, tpts, tvec);

    ThetaM1 = transpose(q(1,:));
    ThetaM2 = transpose(q(2,:));
end
```

```
function [x_elbow, y_elbow, x_foot, y_foot] = calculateTailKinematics(phi_pos, thetaM5,
    thetaM6, t1, t2, x_tail, y_tail)
```

```
    % calculateTailKinematics Calculates positions of the tail's elbow joint and foot
    % Inputs:
    %   phi_pos - Body orientation in radians
    %   thetaM5 - Angle of the first motor in radians
    %   thetaM6 - Angle of the second motor in radians
    %   t1 - Length of the first segment of the tail
    %   t2 - Length of the second segment of the tail
    %   x_tail - X coordinate of the base of the tail
    %   y_tail - Y coordinate of the base of the tail
    %
    % Outputs:
    %   x_elbow - X coordinate of the elbow joint
    %   y_elbow - Y coordinate of the elbow joint
    %   x_foot - X coordinate of the foot (end effector)
    %   y_foot - Y coordinate of the foot (end effector)

    % Calculate the adjusted angles including the body orientation
    adjustedThetaM5 = phi_pos + thetaM5 - 1/2*pi;
    adjustedThetaM6 = phi_pos - thetaM6 - 1/2*pi;

    % Calculate the position of the elbow joint
    x_elbow = x_tail + t2 * cos(adjustedThetaM6);
    y_elbow = y_tail + t2 * sin(adjustedThetaM6);

    % Calculate the position of the foot (end effector)
    x_foot = x_elbow + t1 * cos(adjustedThetaM5);
    y_foot = y_elbow + t1 * sin(adjustedThetaM5);
```

D.6. Function to calculate the stability angle

```
function [alpha, beta] = COM_angle(x_com, y_com, x_base_rear, y_base_rear, x_base_tail,
    y_base_tail)

dx1 = x_base_rear - x_com;
dy1 = y_com - y_base_rear;

dx2 = x_com - x_base_tail;
dy2 = y_com - y_base_tail;

alpha = rad2deg(atan(dx1/dy1));
beta = rad2deg(atan(dx2/dy2));
end
```

Bibliography

[1] Manuel Fernando Silva and J. A.Tenreiro MacHado. *A literature review on the optimization of legged robots*. Oct. 2012, pp. 1753–1767. DOI: 10.1177/1077546311403180.

[2] G. Satheesh Kumar et al. “Literature Survey on Four-Legged Robots”. In: Springer Science and Business Media Deutschland GmbH, 2021, pp. 691–702. ISBN: 9789811544873. DOI: 10.1007/978-981-15-4488-0_58.

[3] Boston Dynamics. *ABOUT SPOT*. 2023. URL: https://dev.bostondynamics.com/docs/concepts/about_spot#about-spot.

[4] Benjamin Katz, Jared Di Carlo, and Sangbae Kim. “Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control”. In: 2019. ISBN: 9781538660270.

[5] ANYBotics. *ANYmal Technical Specifications*. 2022. URL: https://www.anybotics.com/anymal-specifications-sheet/.

[6] Xingxing Wang. *Go2, Unitree Robotics*. 2023. URL: https://m.unitree.com/en/go2/.

[7] Chen Yu and Andre Rosendo. “Multi-Modal Legged Locomotion Framework With Automated Residual Reinforcement Learning”. In: *IEEE Robotics and Automation Letters* 7 (4 Oct. 2022), pp. 10312–10319. ISSN: 23773766. DOI: 10.1109/LRA.2022.3191071.

[8] Alexander Dettmann, Daniel Kühn, and Frank Kirchner. “Control of active multi-point-contact feet for quadrupedal locomotion”. In: *International Journal of Mechanical Engineering and Robotics Research* 9 (4 2020), pp. 481–488. ISSN: 22780149. DOI: 10.18178/ijmerr.9.4.481-488.

[9] Avular. *Avular | Mobile Robotics*. 2024. URL: https://avular.com/.

[10] R. McN. Alexander. *Locomotion of Animals*. 1982.

[11] R. McN. Alexander. “The Gaits of Bipedal and Quadrupedal Animals”. In: *The International Journal of Robotics Research* 3 (1984), pp. 49–59. DOI: 10.1177/027836498400300205.

[12] R B Mcghee and A A Fkask. *On the Stability Properties of Quadruped Creeping Gaits**. 1968.

[13] Danpu Zhao et al. “Gait definition and successive gait-transition method based on energy consumption for a quadruped”. In: *Chinese Journal of Mechanical Engineering (English Edition)* 25 (1 Jan. 2012), pp. 29–37. ISSN: 10009345. DOI: 10.3901/CJME.2012.01.029.

[14] Sangok Seok et al. *Design Principles for Highly Efficient Quadrupeds and Implementationon the MIT Cheetah Robot*. 2013, pp. 3307–3312. ISBN: 9781467356435.