

COMPLIANT LEG LOCOMOTION FOR DYNAMIC QUADRUPED ROBOTICS

by

Vasilije Rakcevic

December 10, 2021

A Thesis

done as a collaboration - supervision:

Technische Universität Berlin - Simon Stapperfend,

Trento University - Andrea Del Prete,

and Augmented Robotics GmbH - Evgeni Melan and Tony Nitschke.

Submitted to the University of Trento

in partial fulfilment of the requirements for the

degree of

Master in Computer Science

(EIT Digital - ES)

Department of Information Engineering and Computer Science

Page intentionally left blank

Table of Contents

Table of Contents	iii
Acronyms	vi
List of Tables	viii
List of Figures	ix
Abstract	xi
Manifest	xii
Chapter 1:	
Design approach	1
1.1 State of the Art	1
1.2 On robot dynamics	3
1.3 Inspiration and observation	4
1.4 Additional aspects of three-link locomotion	5
1.5 Compliance and actuation	7
1.6 Leg structures	9
1.7 Summary on Design approach	11
Chapter 2:	
Mechanical considerations	13

2.1	Initial Parameters	13
2.2	Actuator Choice	16
2.3	Motor	18
Chapter 3:		
	Robotic leg design	23
3.1	Electronics - Brushless Direct Current Motor Controller	23
3.2	Mechanics - Computer Aided Design	25
3.3	Software - Low-level Controller	29
Chapter 4:		
	Motion Testing	33
4.1	Test-stand	33
4.2	High-level Control and Simulation	35
4.3	Motion Execution	39
4.4	Shown potential and future enchantments	44
Chapter 5:		
	Conclusion	47
Appendix A:		
	PCB design considerations (general guidelines)	49
Appendix B:		
	Additional data on simulation and execution of robot motion	51
Appendix C:		
	Hands-on lessons learned	54

Appendix D:**Legged robot project page links 56****Bibliography 56**

Acronyms

ADC Analog to Digital Converter.

BLDC Brushless direct current.

CAD Computer aided design.

CAN Controller Area Network.

CoM Center of mass.

DDP Differential Dynamic Programming.

DMA Direct Memory Access.

DoF Degrees of freedom.

FOC Field-oriented control.

KT Motor torque constant.

KV Motor speed constant.

P Proportional.

PCB Printed circuit board.

PI Proportional-integral.

RC Remotely controlled.

RPM Revolution per minute.

SPI Serial Peripheral Interface.

SVM Space Vector Modulation.

TTY Teletype.

UART Universal Asynchronous Receiver Transmitter.

URDF Unified Robot Description Format.

USB Universal Serial Bus.

List of Tables

2.1	Basic Robotic Leg Parameters	15
2.2	Review of Robot Actuators from SOLO and from MIT Mini Cheetah.	17
2.3	Explanation of some of the important parameters specified for BLDC RC drone motors.	19
2.4	Motors chosen for the design.	20
3.1	Parameters and capabilities of the BLDC control board.	23
3.2	CAN data format.	25
3.3	Gear ratio for the actuators.	26
3.4	Controller interface.	30
4.1	Torque to current ratio for motors.	44

List of Figures

1.1	Dog running up the stairs.	3
1.2	Dog Running.	4
1.3	Body height in two vs three links design before and after impact.	5
1.4	Force applied to a contact point independently by every joint in two-link vs three-link design.	6
1.5	Different leg structures.	10
1.6	The simplest tensegrity structure (a T3-prism).	11
1.7	Material to density ratio for optimised car chassis topology.	11
2.1	Leg position and forces used for static load approximation.	14
2.2	Motor simulation with 24V and 48V power supply.	20
2.3	Comparison of Anti-gravity T-Motors: 6007II, 6007 and 5008.	22
3.1	PCB design.	24
3.2	Generative design setup for the robot leg - Thigh Link.	26
3.3	Generative Design outcomes.	27
3.4	Parts to be manufactured in a mechanical shop.	28
3.5	Finished CAD design of the whole leg.	29
3.6	Low-level controller events timings.	30
3.7	Low-level controller regulating in the transformed domain of DQ currents.	32
3.8	PI controller of inner current loop.	32
4.1	Leg test-stand.	34

4.2	Gepetto-GUI (Simulation tool used in Crocoddyl library) with representation of a robot.	35
4.3	Jumping motion convergence parameters through optimizer iterations.	38
4.4	Simulated jumping motion.	38
4.5	CoM position and body height of simulated jumping motion.	39
4.6	Setpoint position step from zero to the random robot joint configuration. . .	41
4.7	Tracking of sine wave with 1 <i>rad</i> amplitude and 1 <i>Hz</i> frequency.	41
4.8	Simulated jumping motion trajectory for Knee, Hip and Ankle joints.	42
4.9	Execution of simulated motion on the robot with position tracking.	42
4.10	Performing simulated motion in the air with position tracking controller. . .	43
4.11	Leg performance when dropped from 0.4 <i>m</i> height.	44
B.1	Trajectory of generated jumping motion - position, speed and torques at the joints with bounds depicted.	52
B.2	Step response of the Knee actuator. Sensed position, Phase A and Phase B currents.	53
B.3	Tracking of sine wave with 1 <i>rad</i> amplitude and 2 <i>Hz</i> frequency.	53

Abstract

Inspiration in animal movements and structures is not a novel approach, it is being utilized for the development of popular robots such as Boston Dynamics - Spot [1], largely in MIT Cheetah [2], as well as many other legged robots being introduced recently. As to which resemblance seems reasonable - it varies; all of the solutions see animal movement as superior and worth further exploration and exploitation. Already, developed solutions are finding wider applications, legged robots are planned to be used even when it comes to new planet explorations, as recently presented by NASA [3].

Despite all the advancements, the gap between nature and robotics is still quite large. Robots are characterized by "robot like" movements - mainly sequential and precise. Precise positioning is not enough for achieving good locomotion and other adopted strategies are not able to mitigate all the problems. The resulting movement can appear quite coarse and slow compared to that which can be observed in biological species.

The thesis represents a development on the idea of a highly dynamic design and compliant leg structure, allowing for legged locomotion that resembles a running movement of biological animals. One leg solution is presented, which can be used as a blueprint for the full scale quadruped robot design. It is being argued that the three-link design and tight integration of different compliance principles, as presented further, will lead to better dynamics in comparison to the current solutions. Feasibility of such design will be proven in respect to minimisation of inertia and allowing independent control of each link. Furthermore, important concepts of contact force control, leg compliance and the way of its in-design integration will be introduced. The leg has been manufactured and some of the underlying principles were tested. It represents an important foundation for future research and development.

Manifest

As this work is not intended to be complete or sufficient to tackle every aspect of the stated problem, the idea is to frame it as a toolbox/template for future development. As such, current developments and all future enhances and contributions should be founded on the following pillars:

- Open-Source
- Modular
- Research oriented

Despite popular misconception that Open-Source means just disclosing your achievements, there are numerous benefits to it. One of the most important aspects is the ability to engage with a community - as in shared economy, the entity becomes more valuable as more individuals would engage. For that reason Modularity is a necessity - a project is covering wide areas, and any participant should be able to work independently on improvements of certain aspects and deploying it by respecting the common interface (mechanical, software or any other). Research orientation would provide the ability to engage with the scientific community, opportunities for collaboration, exchange of ideas and sharing acquired knowledge in a scientific manner - hopefully making significant impact in this domain.

Chapter 1

Design approach

1.1 State of the Art

Envisioned capabilities motivate researchers to engage, further improve characteristics and make legged robotics more affordable option. Developed solutions are becoming a largely popular for tackling the problem of movements in rough and unpredictable terrains, offering many advantages to its wheel predecessors , which are largely limited to flat or gently sloped surfaces.

For widespread of technological awareness and facilitation of opportunities to contribute, especially, the role of Open-Source is to be acknowledged. Some hardware and software achievements from MIT Mini Cheetah are freely available [4], others offer complete open hardware and software legged robot solutions; all aiming to attract interest and boost development. As all this has been only recently introduced to a broad attention - in the last several years. This topic represents a new, undeveloped and interesting research area.

As for the current solutions themselves, when observing the nature, it is quite difficult to isolate a single component for solving a particular problem. It is always a synergy of solutions required for the adoption to specific living conditions, such as food, climate, lifestyle... In certain species, some aspects are more predominant than others, making them easier to separate. As for running, the MIT Biomimetic Robotics Lab has based its design on a

cheetah, in their MIT Cheetah robot line [2]. MIT Cheetah 3 is able to perform jumps over obstacles and run at 6 m/s. They introduced a novel actuator design - Proprioceptive actuator [5], which is used for all three robot actuators - Ab/Ad (Abduction/adduction - side movement), Hip and Knee actuation. Their recent development is a smaller scale quadruped - MIT Mini Cheetah [6] - with similar dynamic capabilities to its predecessor showing advanced jumping motions. For example, its ability to perform back-flips, go through narrow passages, ... This robot is presented as low cost, lightweight and much safer to operate version. Many design aspects are available through the articles and other published work (A low cost modular actuator for dynamic robot by Benjamin G. Katz [4] - MIT Mini Cheetah actuator design). It has been developed for the purpose of researching quadruped robot dynamics and represents a great reference to get familiar with the state of the art from the implementation point of view. Another solution - SOLO [7] - from the Open Source Dynamic Robot Initiative is moving entry boundaries even closer by providing step by step instructions on how to assemble a robot. It is built with mostly off-the-shelf components and 3d printed parts, weighs 2.2 kg, has a large range of motion and among others, can perform vertical jumps.

From the other side, different use-cases such as space exploration can boost development of some other aspects of legged robotics as-well. Space introduces several new challenges - operation in different gravity, extreme temperatures, energy harvesting, ... However, achievements are shared successes and can be used interchangeably between different fields of robotics. Interesting way of dynamic locomotion is presented by SpaceBok - a robot designed for mars and moon exploration. It performs advanced movements in highly dynamic pronking gait. They are leveraging extended flight phases for more efficient locomotion in low gravity applications. Robot has a mass of 20 kg, a hip height of 500 mm and can perform jumps up to 1.05 m and reach velocity of 1 m s⁻¹. [8] Others focus on less dynamic solutions but nevertheless capable of exploring some of the extreme surfaces of foreign planets. MANTIS, a multi legged robot with manipulative abilities. They leverage multiple contact

points for locomotion over difficult terrain. [9].

It is hard to reason about which approach is the best general use-case, however, dynamic improvements will enhance performance of all mentioned robots.

1.2 On robot dynamics

As presented in previous chapter, legged robots implement some of the most advanced technologies, however they are still not able to outperform dynamic abilities of biological animals. One can argue that adopted design choices are rather a necessity introduced to make robots more robust. However, if that is the only way, why are humans made different? Why bone structures are not made from a harder material? Maybe our body structure is not robust enough. So, if these methods are being adopted, does it mean that one is lacking somewhere else? Would this be a worthy compensation?

This thesis is not trying to outperform or shade out developed technologies, but it wants to tackle one particular aspect of it - dynamics. In particular, one can formulate a problem statement: "Legged robot running up the stairs - comparable with dog abilities", as shown in Figure 1.1. There are several key observations and difficulties associated with such motion:

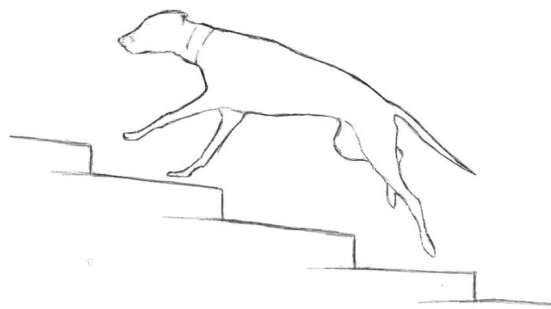


Figure 1.1: Dog running up the stairs.

- The motion is a combination of jumping over stair steps and overall a locomotion in quite difficult environment.

- Smoothness of the motion and similarity to the dog movement can be a good visual measure of the robot capabilities.
- Reaction speed to unpredictable situations and obstacles, and ability to recover as an important indication of the robot performance.

If we can agree about the limitations of current robots, it can be concluded that this task can not be fulfilled.

1.3 Inspiration and observation

It is worth taking a closer look on some of the dog running dynamics.



Figure 1.2: Dog Running. [10]

Several key observations with respect to Figure 1.2:

- Center of mass (CoM) height fluctuations are kept minimal.
- Each leg has three planar links, which allow redundancy in the foot planar position.
- While running, legs are trying to push the body forward rather than upwards.

The argumentation will be clarified by the analysis of the usual two-link leg robot design and comparison to the three-link solution. When the CoM height is considered, it is important to note that to utilize more energy efficient locomotion, legs must push the inertia in the direction of running and minimize its other components (perpendicular to the running direction, pointing to the side and upwards). As a simplification, only the planar motion was analyzed for this scenario. In running, leg dynamics can be simplified as a sequence of

landing, energy transfer (inverted pendulum), pushing and flying phases. In a two-link design, the problem arises when trying to transfer the energy in the landing sequence. If trying to comply for the impact force reduction, considering that links are stiff, the ankle and hip joints will not be holding their positions. Unless they are held stiff - design over-provisioning would be required in order to sustain high impacts. It will result in the violation of first observation as shown on Figure 1.3. This is solved by introducing a third link. If one link complies, due to the redundancy, other link can compensate that compliance by expanding in the other direction - resulting in the ability to better hold the CoM height constant.

The two-link design is often the main reason why the robot might appear as jumping while running. It results in a huge energy loss and requires more powerful joint actuators compared to a three-link design.

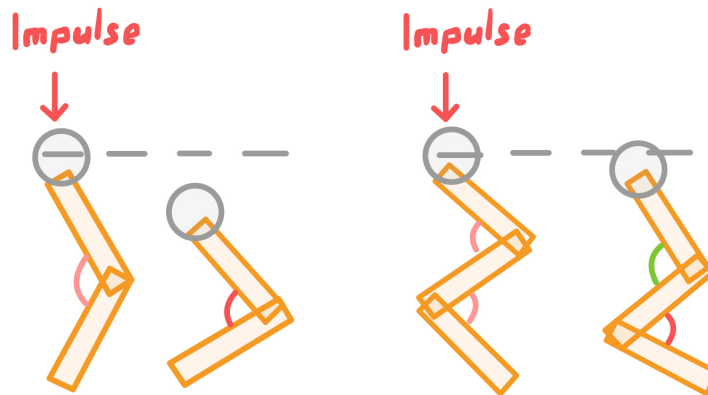


Figure 1.3: Body height in two vs three links design before and after impact.

1.4 Additional aspects of three-link locomotion

To continue locomotion after landing, the leg is acting as an inverted pendulum by transferring a mass around the contact point, after which the dog engages into a new push for the next running sequence. In a two-link design, this push is initiated with a maximum of two joints. Because of the geometrical limitations of a two-link leg structure, it is much harder

to achieve proper configuration for which ankle and hip joints engagement would minimize undesired upwards force - Figure 1.4. In such situation the robot movement is much more influenced by physical processes of motion (robot inertia) - timing becomes quite critical for the ability to properly engage in the next running sequence. Thus, every obstacle can largely disturb the motion. In a three-link design the situation changes: At any moment, the contact force vector can be adjusted through change of the robot configuration. Such strategy is much more robust and allows better controlled and dynamic locomotion.

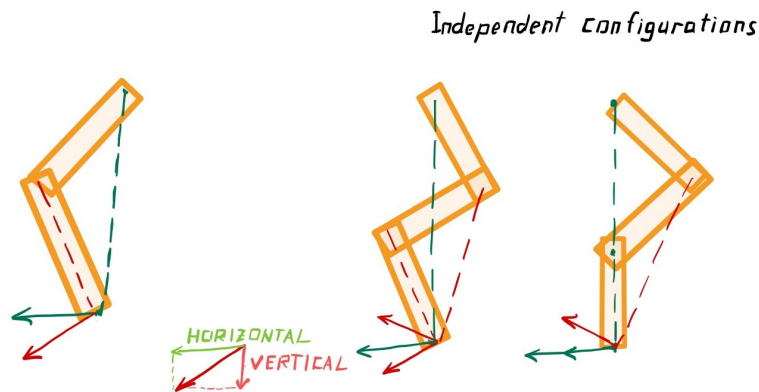


Figure 1.4: Force applied to a contact point independently by every joint. In a two-link design the direction of forces at the contact point can not be adjusted in particular moment (they are likely to be changed only once the robot continues its inertial movement, or through further engagement of the joints, which would start influencing height of the robot). In a three-link design the robot can adjust the configuration independently of the other factors, thus controlling the direction of applied forces at the contact point for particular situation.

There are some trade-offs which should be made when it comes to the three-link design. In comparison to two links, an additional link will increase the inertia of the leg. In animal legs this is controlled by maintaining all muscles (actuation) in the upper part of the leg and connecting to the other parts through ligaments and other structures in order to be able to transfer forces. These structures transfer forces in both directions, thus protecting the bones. Even being strongest among the body elements, bones mostly redirect forces and sustain strongest impacts acting alongside their biggest dimension. To protect the muscle and bone structures, all the supporting elements have a certain level of compliance.

In conclusion - there is a well-founded argument to use a leg with three planar links.

Among others, the structure should be able to transfer the forces to the actuators, sustain impacts alongside their biggest dimension and distribute impacts otherwise. Compliance should be added to different elements to be able to reduce the forces acting on the actuators.

1.5 Compliance and actuation

Quantifying the capabilities of the robot is often closely tightened to its precision of movements, the ability to sustain force at particular position and the ability to recover from failures if not able to respect given requirements. These notions are often separated from compliance, which is rather seen as a necessity and trade-off in robotic control. But actually, the ability to leverage it, can have positive effects on controllability, as proven by the Compliant Hand Design [11]. Control should affect boundaries, but motion is synergy between the environment and the robot itself. The key lays in observation that once we try to pick up a coin from the table, hand control is not in any way precise, but rather, is being guided by the obstacles around (hand first hits the table and than continues motion under that constraint). One can conclude that nature has designed human hands to be quite compliant rather than strictly precise in certain situations. These concepts are shared among other "mechanical" structures in humans. When we walk, we don't differ much when it comes to walking on smaller bumps, or completely even terrain - we might try to change next step position, but other than that, our conscious control would stay similar. In interaction, the leg would adopt to uneven surface without us having to engage in any way as long as it is in the particular boundaries of our physical abilities (not over-stretched, over-twisted ...). Remarkably, in animals and humans, these structures are able to change its appearance from more elastic to more stiff, depending on force exerted by our muscles, which is quite hard to recreate in a robot design. However, when looking in isolation, rather than as a whole, we can distinguish correlation between certain structure properties (stiffness/compliance) and particular movement gates.

One aspect that is often being neglected is elasticity, which should come along the muscle (torque) control. In animals muscles stiffness is changed by changing their engagement, which, on the other hand, reduces the elasticity. Elasticity is most apparent in case muscles are not engaged at all. On the other hand, stiffness is minimized in full engagement. Having only one aspect in the design automatically means depriving the machine of certain capabilities compared to animal structures.

Impedance control is one of the popular choices to solve this problem. Implemented mostly for stiff systems, such control introduces "elasticity" by having a controller acting as a spring-mass-damper system through maintaining the relation between force, position and its derivatives. [12] Its main challenge is to reduce the impact force in high gear ratio systems, but this design is likely to lead to extensive mass and not that robust design.

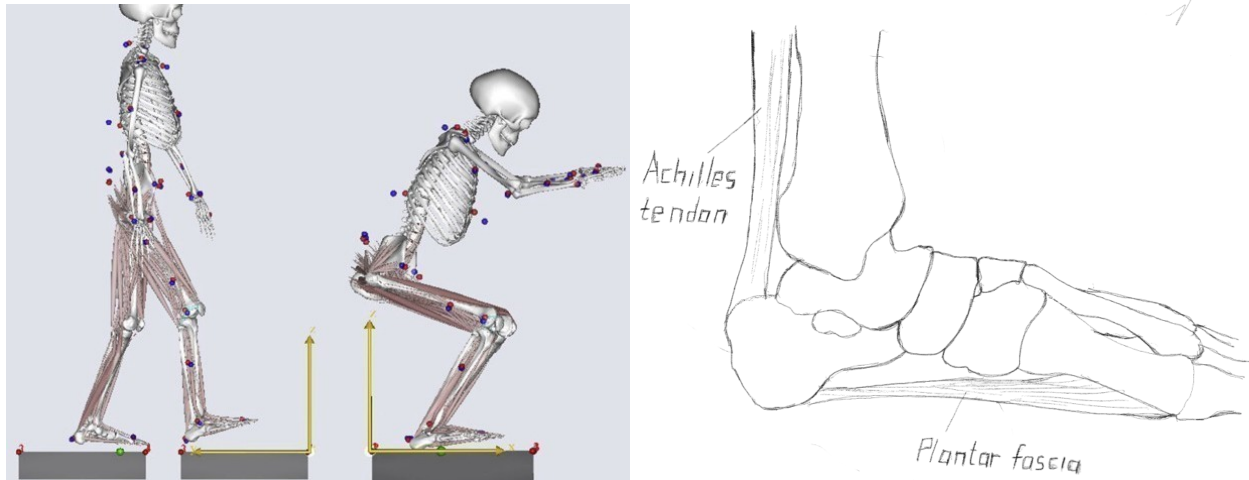
In Series Elastic Actuators the spring with encoder is added in series to the motor and gear system. The main advantage of such design is high impact mitigation, but it comes at a cost of reducing the torque bandwidth. Still, with stiffer springs, torque bandwidth can be kept within desired limits. For the StarLETH robot the joint torque bandwidth is about 28 Hz for low amplitudes and about 11 Hz for large amplitudes. [13] The main disadvantage of such design is the additional weights and overall design complications when it comes to accommodating springs.

The proprioceptive actuation concept introduced a new actuator design to approach the problem from a different perspective. Instead of adding springs to reduce impacts on the actuator, it is able to mitigate it by lowering the gear ratio in design stage and having a high force density motor to damp the impacts. Overall, it results in a highly backdrivable systems and a precise force estimation without additional external sensors through motor current sensing. The geometry of the motor is highly influential in this case. Larger gap radius motors (pancake shaped) are better suited for this scenario since such design has higher torque density, thus it requires a smaller gear ratio, resulting in overall better impact mitigation, while maintaining high torque bandwidth. [5]

All the previous solutions are solving the two problems (elasticity and force) through one domain (actuators enhancements, special control strategy) or by quite sophisticated additions (springs in series) which is possible to avoid to simplify design. Proprioceptive actuation performed best when it comes to legged robot dynamic control, but one drawback is that over-provisioning of actuation power is required to accommodate higher impact suspension - otherwise, some impact power would be absorbed by compliant structures, when present. However, as previously mentioned, accommodating springs will complicate the design. The goal is to leverage other system parameters in order to achieve the desired effect.

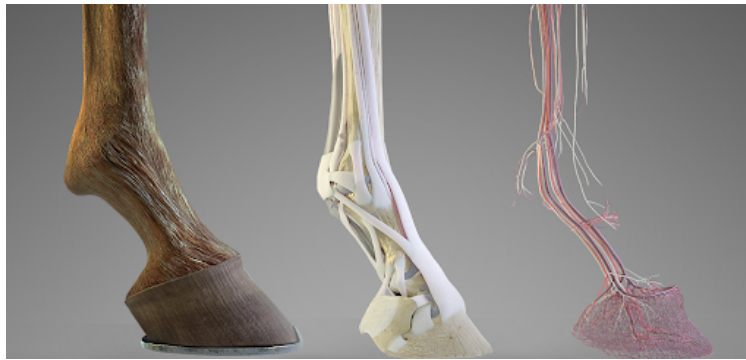
1.6 Leg structures

In [14] Prof. Sangbae Kim reasoned about the complexities of animal bone structure, which can be seen through the bio-tensegrity concept introduced by Steve Levin [15]. Animals are capable of withstanding large impacts on the extremities despite only consisting of disproportionately weak elements when looked in isolation. One can hardly recreate all fine muscles and ligaments in the leg structure (example of horse leg anatomy shown in Figure 1.5c), but we can focus on predominant elements for functions of dynamic locomotion. Prof. Sangbae Kim noted the importance of Plantar Fascia and Achilles tendon - Figure 1.5b. These are two main formations, which through elongation are able to reduce bending moments on the bone structures. Bones are responsible for handling the compression forces, while various tissues handle bending forces. Through bio-tensegrity, they are working together to create natively stable structures and better sustain impacts due to the proper forces distribution. For better understanding, an example of tensegrity structure is presented in Figure 1.6. It was applied to the MIT Cheetah robot by introducing tension material along the robot links [16]. Such design has proven to minimize stress concentrations (due to the bending stress) on stiff structures for certain situations. Bending stress is sustained by the soft elements (tension material) through their elongation.



(a) Standing (left) and squatting (right).[17]

(b) Plantar fascia and Achilles tendon.



(c) Horse leg anatomy. [18]

Figure 1.5: Different leg structures.

As we do sacrifice versatility of human motions by taking only some aspects of the leg structure, one can also argue that needed elongation-compression capacity can be achieved through special structures integration from elements of the same material. This concept is already a long living standard in automation industry with the goal of reducing the mass of the car - Figure 1.7. Another benefit is better impact distribution, which overall leads to much safer cars compared to the classical full body design. At any point in time during impacts, certain elements would be supporting by suppression, other by bending and elongation. The structure optimisation can be biased by introducing larger forces along certain direction in the optimisation model. The only drawback is that these structures are hard to manufacture by traditional methods. However, if suitable material can be found for Additive

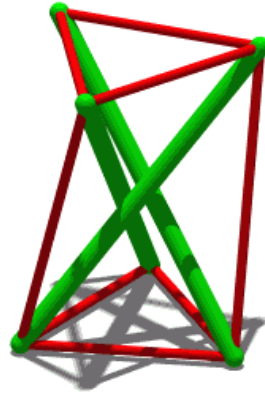


Figure 1.6: The simplest tensegrity structure (a T3-prism). [19] The green bars are stiff elements (compression members) and red are soft elements (cables). This structure is natively stable, meaning that cooperation between compression elements and strings allow it to stay at certain position even when influenced by disturbances. In case any of the elements would be removed, or if any compression member replaced by the soft element, the structure will lose its stability.

Manufacturing (3d Printing) no large sacrifices are required to make it manufacturable.

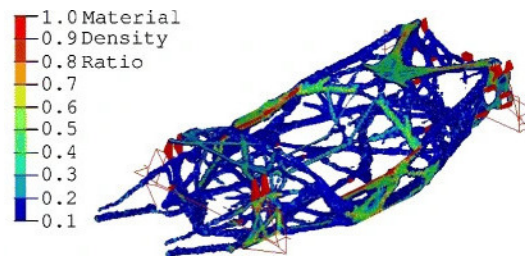


Figure 1.7: Material to density ratio for optimised car chassis topology. [20]

1.7 Summary on Design approach

So far we have discussed different existing and proposed solutions for the design of highly dynamic robots - elasticity-compression for impact mitigation within tolerable bandwidth limits, actuator force density and backdrivability. We can conclude that:

- Proprioceptive actuators are the best for force control abilities, offering reduction of size and complexity.

- To further reduce required actuator power, one can introduce elasticity (compliance) with sacrifices in the bandwidth domain (not a problem if kept within limits of quadruped bandwidth requirements). Elasticity should be introduced to reduce bending stress on the stiffer structures during impacts.
- Tailoring to certain characteristics of motion, compliance can be introduced by specially designed mechanical links with integrated elongation-suspensions through shape optimisation, thus freeing design from any excessive mass or additional elements usually required for spring introduction or usage of any other material for the purpose of introducing elasticity.
- Through shape optimisation it is possible to achieve symmetrical links to ensure direction independent movements.

The manifestation of tensegrity can be seen also through biased body configurations in humans for example. Staying in certain body positions requires less effort than in others, no big effort is required to keep neck vertical compared to holding it more horizontal. We can conclude that these positions are local equilibria in tensegrity structures. Robots usually do not have such equilibria and require active joints to maintain these local stability positions found in all walking animals. Since we are concerned with running and walking motion, the only equilibria in this domain is the standing posture - standing does not require much effort, however, in squatting position - Figure 1.5a, we can hold on for only a short amount of time. Additional considerations for mechanical design are required for that, which will be discussed in the next chapters.

Chapter2

Mechanical considerations

2.1 Initial Parameters

Several key points were taken into account for the initial mechanical design parameters approximation based on the conclusions of the previous chapter:

- Our quadruped robot should have legs with three links.
- Mass should be concentrated in the body and axial to the hip joints for links to be lightweight in order to reduce inertia.
- No constrains to the motion of the links should be considered beside their attachments to one another.

The designed robot is chosen to be of small dimensions (less than 8kg) for the ease of experimenting and safer manipulation. Each leg, in addition to its own body mass should be able to carry $m_a = 1kg$ of weight (modeled as additional weight applied to the hip joint). For forces approximation the leg is modeled in the static position where all joints are experiencing large forces - Figure 2.1. It was decided that this configuration provides best insights into force profile of the three-link design.

Parameters required for calculations can be found in Table 2.1. Forces are calculated by considering that the sum of the forces at any point of the leg should be zero, as shown

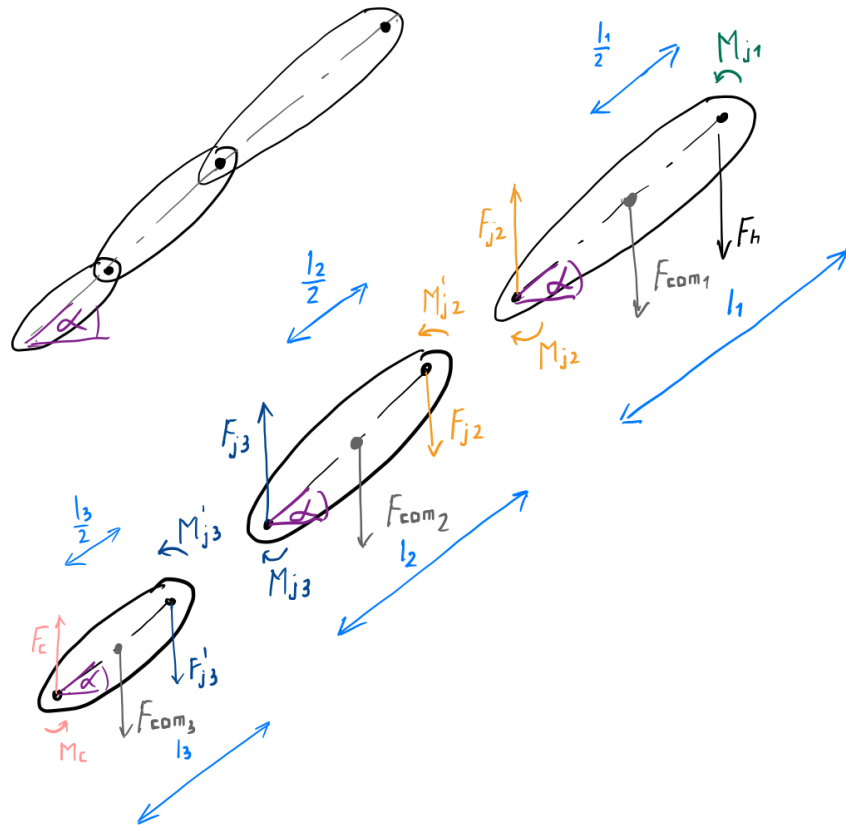


Figure 2.1: Leg position and forces used for static load approximation.

in (2.1). From the other side, moments are calculated by considering that the sum of the moments with respect to any point of the leg should be zero, as shown in (2.2).

Table 2.1: Basic Robotic Leg Parameters

	Value	Unit	Note
l_1	0.15	m	length link 1
l_2	0.12	m	length link 2
l_3	0.1	m	length link 3
m_h	0.6	kg	hip mass applied to its axis
m_a	1	kg	additional mass applied to hip axis
m_1	0.3	kg	mass of link 1
m_2	0.2	kg	mass of link 2
m_3	0.1	kg	mass of link 3
k_f	1.5		force safety coefficient
α	30	deg	leg angle in respect to ground

$$F_{j1} = k_f * (m_a + m_h) * g = 23.55N$$

$$F_{com1} = m_1 * g = 2.95N$$

$$F_{j2} = F_h + F_{com1} = 25.5N$$

$$\vec{F}'_{j2} = -\vec{F}_{j2}$$

$$F_{com2} = m_2 * g = 1.96N$$

$$F_{j3} = F'_{j2} + F_{com2} = 27.46N$$

$$\vec{F}'_{j3} = -\vec{F}_{j3}$$

$$F_{com3} = m_3 * g = 0.98N$$

$$F_c = F'_{j3} + F_{com3} = 28.44N$$

(2.1)

$$\begin{aligned}
M_c &= 0 \\
M'_{j3} &= M_c + \cos(\alpha) * (F_c * l_3 - F_{com3} * l_3/2) = 2.42Nm \\
\vec{M}_{j3} &= -\vec{M}'_{j3} \\
M'_{j2} &= M_{j3} + \cos(\alpha) * (F_{j3} * l_2 - F_{com2} * l_2/2) = 5.17Nm \\
\vec{M}_{j2} &= -\vec{M}'_{j2} \\
M'_{j1} &= M_{j2} + \cos(\alpha) * (F_{j2} * l_1 - F_{com1} * l_1/2) = 8.29Nm \\
\vec{M}_{j1} &= -\vec{M}'_{j1}
\end{aligned} \tag{2.2}$$

2.2 Actuator Choice

There are many open source actuator designs available, however, considerations were taken in respect to their already successful usage in the quadruped robotics, as well as ease of integration with other required system components. The ability to further develop and scale given solutions were also an important factor. Following our previous conclusions regarding the positive aspects of proprioceptive actuators, among considered, two platforms were chosen for more detailed analysis. Their overview is provided in Table 2.2.

Mini Cheetah actuator controller has certainly some advantages due to its customisation abilities as well as the capabilities, especially when it comes to handling more powerful motors. The controller is also an integrated part of the actuator and is placed on top of the motor (position sensor is part of the controller hardware). On the other side, SOLO's actuator strategy is quite convenient for faster prototyping by using readily available Printed circuit board (PCB) solutions as is, or through smaller redesign and adaptations. Another

Table 2.2: Review of Robot Actuators from SOLO [21] and from MIT Mini Cheetah [5].

Parameters	SOLO Actuator (TI Eval Board /Micro Driver)	MIT Mini Cheetah Ac- tuator
MAX Operating Voltage	45V	30V
MAX Continuous current	15A	30A
FOC Control	10kHz	40kHz (possibility for in- crease)
Communication	CAN and USB	CAN and UART
Low level driver customisa- tion available	Only as provided by TI Driver	FOC is completely ad- justable and replaceable by any other control im- plementation due to the Open-Source nature. Im- plementation for the STM microcontroller available
Motors suitability	T-motors 4008 or similar performance Brushless direct current (BLDC) Remotely controlled (RC) drone motor	T-Motor U8 or similar per- formance BLDC RC drone motor
Gear system	Pulleys/belts integrated in the link designs	Planetary gear system 6:1

advantage of SOLO's design is their strategy to use the pulley/belts systems which are separating gear elements from the actuator, thus reducing the size of the actuator and providing easier solution for implementation.

We find that the combination of both actuators ideas would be best suitable for this design requirements. Gear reduction should be achieved through a belt-pulley system, but to achieve size compactness, the in-actuator - integrated BLDC driver idea from MIT Mini Cheetah Actuator can be used.

MIT Mini Cheetah BLDC Driver design files are provided on GitHub [22]. Several companies leveraged the freely available design to create copies, start manufacturing and selling the driver. However, the copies are rather purely designed, not respecting many decisions and reworks done through experimentation of Ben G. Katz described in [23].

When it comes to other relevant and available driver solutions, ODrive board [24] shows

the best performance. However, there are several drawbacks in respect to this design requirements: (1)size considerably larger compared to MIT Mini Cheetah BLDC driver, (2)each board comes with support of two motors (when uneven number of actuators, one driver set will not be used), (3)requires additional external position sensor, (4)designed for much more powerful motor - design over-provisioning.

In conclusion, considering ODrive's performance, it can be a valid alternative. However, due to compact size, availability of all required components within one package, specifically tailored design for actuator applications, the MIT Mini Cheetah BLDC driver was chosen as a base for this design. ODrive software still can be used with smaller compatibility changes - both solutions are based on STM32 micro-controllers. For the purpose of PCB size adaptation, improvements and easier future debugging, it was decided to redesign this controller from scratch.

The control driver and the gear system design will be discussed in the following chapters.

2.3 Motor

As gear transmission efficiency is decreasing by the rise of gear ratio, choosing right gears is crucial for preserving backdrivability. Additionally, one needs to keep in mind that step-up directional efficiency is usually lower compared to step-down, and for 6:1 gear ratio efficiency was 96% and 98% respectively. The importance of smaller gear ratio is clear when a gear ratio of 248:1 is considered - it will have approximately 75% and 70% for step-down and step-up respectively [25]. SOLO robot is showing great performance metrics with 9:1 gear ratio [7], thus for this design we will try to keep 9:1 gear ratio or as close as possible.

Maximum torque required from the calculations in (2.1) is 8.29 Nm. With 9:1 gear ratio, the actuator maximum torque will be:

$$M_r = M_{j1}/9 = 0.92Nm \quad (2.3)$$

Table 2.3: Explanation of some of the important parameters specified for BLDC RC drone motors.

Parameters	Intuition and explanation
Motor speed constant (KV)	Ratio between no-load motor Revolution per minute (RPM) and voltage applied to the motor
Motor torque constant (KT)	Ratio between amount of torque produced to the motor current
Peak Current - I_{max}	Limited by power dissipation capability of the motor - $I_{max}^2 * R$ (smaller resistance, bigger maximum current limit)
Rated Voltage - U	Voltage Limit (related to internal resistance and peak current capabilities- bigger resistance leads to bigger voltage limit)
Max Power	Max Output power given by approximately $I_{max} * U$

BLDC RC drone motors are usually not produced for low speed application, thus specifications are not depicting their low speed (torque predominant) behaviour. However, that does not mean they are not suitable for the actuator design. From proprioceptive actuator [5] data we can conclude that the particular RC drone BLDC motors with larger gap radius and smaller height ("pancake shape") have a potential to be the best solution for backdrivable actuation.

To better understand motor specifications provided by the manufacturer, in the Table 2.3 some of the most important parameters are explained.

It is important to note that the rated voltage is just maximal suggested operating voltage. Since, in the legged robotics applications, the motor is operating in low RPM / high torque mode, lowering voltages from 48V to 24V will not largely influence motor performance provided that the low speed behaviour is limited by the controller and the motor power capabilities as shown in Figure 2.2.

Characteristics of the best suitable motors among Drone RC motors:

1. Lower KV motor, since the KT is inversely proportional to KV

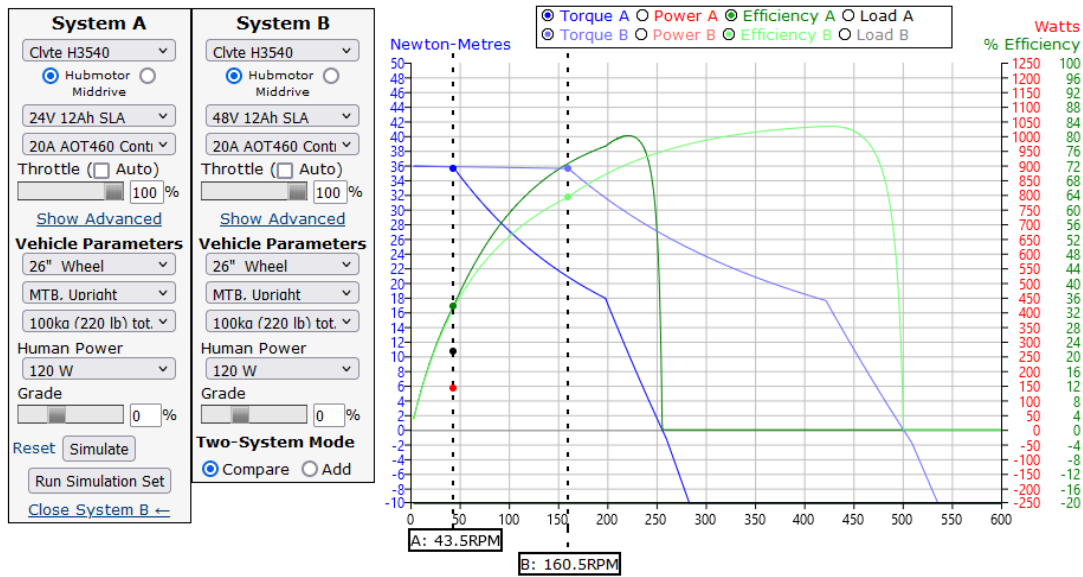


Figure 2.2: Based on simulation of the Clvte H3540 motor with two different batteries used (System A-24V and System B-48V) only the constant torque region is reduced but torque capabilities in low RPM region stay the same. It is due to the power limitation of the controller as well as the motor maximum power limit. [26]

2. Smaller winding resistance (more capable in respect to I_{max})
3. Highest overall power ability
4. Bigger number of poles (better torque distribution around the rotor)
5. Bigger gap radius and smaller thickness (pancake shaped geometry) for better torque density

Table 2.4: Motors chosen for the design.

Motor	KV	Weight	Max Peak Current	Torque and current from specification	Internal Resistance
T-Motors 6007II	160	0.159kg	23.7A	1.3Nm at 20.67A	0.178Ω
T-Motors 5008	170	0.128kg	15A	0.87Nm at 14.63A	0.270Ω

In a Drone RC motor specification, the manufacturer usually provides dependence between no-load throttle at certain supply voltage as input parameter, torque and current

as output parameters. In order to compare between different motors it is useful to plot torque-current ratio and torque value at certain RPM. To better understand the graphs, one needs to keep in mind that torque-RPM dependency is specified for the increase of voltage throttle. Several T-Motors were compared in Figure 2.3. They were chosen based on geometry, price (under 130 dollars), good KV rating, number of poles and overall availability. There are many other companies, mainly from China, which are producing similar motors in size but the build quality has not been tested by many users. Considering an affordable price, T-Motors 6007 and 5008 series were chosen for further evaluation - Table 2.4. Torque abilities of 6007 series satisfy the highest torque requirement from (2.3). From the two 6007 motors, one can see that they are similar in performance but the improved version (6007II) has a bit less weight and resistance (which means it can handle bigger currents) potentially giving higher torque capabilities. Motor 5008 is less powerful version but it is cheaper and lighter compared to 6007 series- that is why it was decided to use this motor for the weakest actuator. After all, the idea is to be able to compare the performance of several motors in practice.

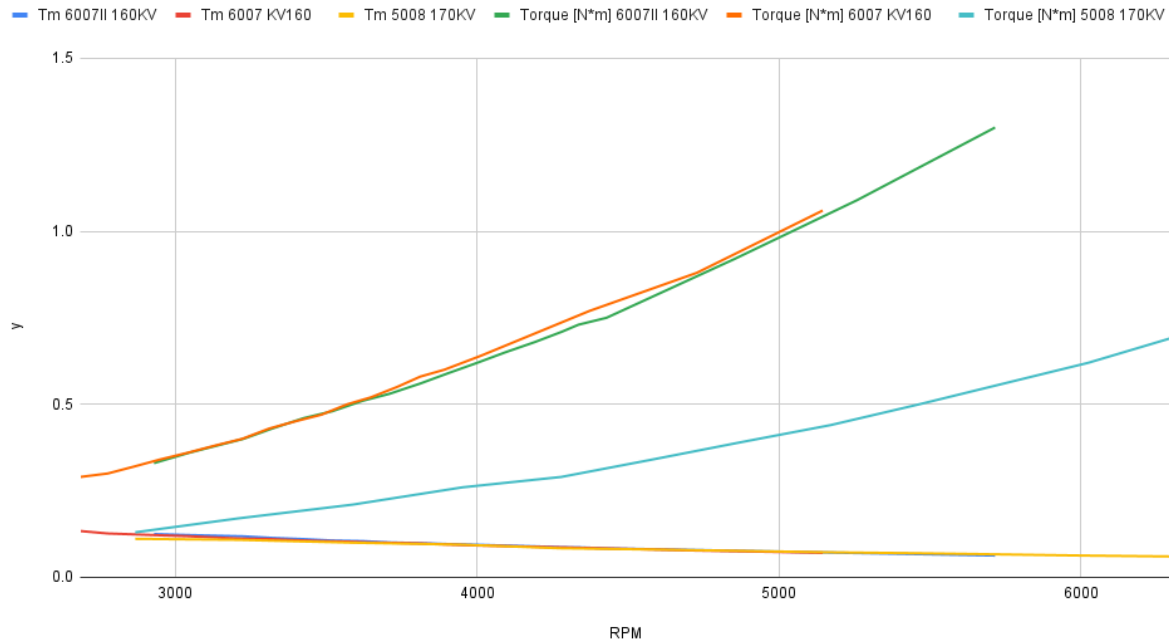


Figure 2.3: Comparison between Torque and $T_m = Torque/I$ at different RPM's, for Anti-gravity T-Motors: 6007II, 6007 and 5008. T_m constant stays the same among motors which suggests that the motors have similar build structure. However, due to the higher current capabilities, 6007 series are able to generate more torque in comparison to the 5008 series motors. T_m for all motors is decreasing since torque ability is likely to be reduced due to the induced electromagnetic currents as RPM is increasing. It differs to the usual motor chart where torque is maximal at lower speed, since in BLDC motor specification, speed and torque values are related to the increase of voltage throttle, that's why both values are being increased.

Chapter3

Robotic leg design

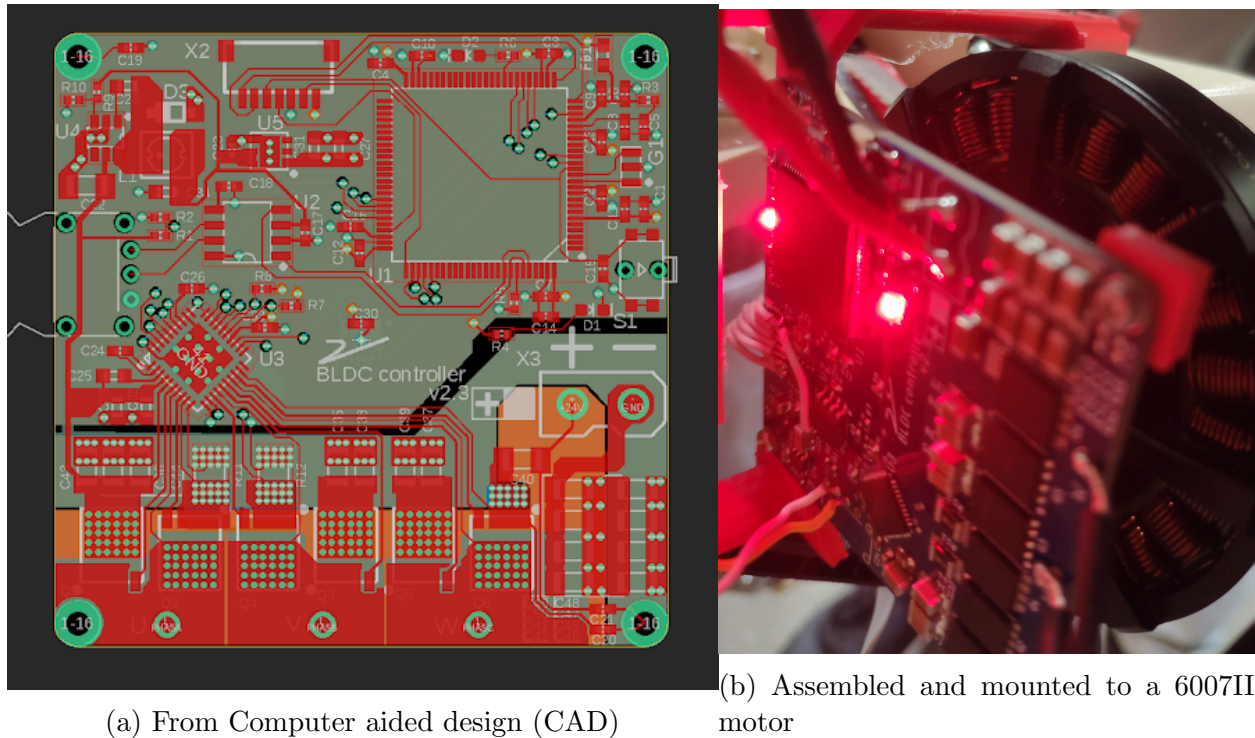
3.1 Electronics - Brushless Direct Current Motor Controller

Based on the previous analysis we have concluded that maximum peak current of the chosen motors is 23.7 A - Table 2.4. The BLDC controller from Ben G. Katz is designed for more powerful motors [4], but as described in Section 2.2, has been chosen as a base for this project controller. Its compact size, capabilities and documentation provide a convenient starting point for custom enhancements and further adaptations.

For the PCB design it was chosen to use Eagle Autodesk. Eagle has free student licence. This software was used for the initial PCB design as-well.

Table 3.1: Parameters and capabilities of the BLDC control board.

Parameters	Value
Max Operating Voltage	30V continuous (peak up to 45V)
MAX Continious current	30A (limited time)
Field-oriented control (FOC) Control	40kHz (adjustable)
Communication	Controller Area Network (CAN) (1Mbit/s) and Universal Asynchronous Receiver Transmitter (UART)
Customisation available	changing code on STM microcontroller
Motors suitability	Designed for T-motors 5008 and 6007II or similar performance BLDC RC drone motor



(a) From Computer aided design (CAD)

(b) Assembled and mounted to a 6007II motor

Figure 3.1: PCB design.

Some general guidelines about PCB design can be found in the Appendix A. The board design files are made freely available. It is small ($5\text{ cm} \times 5\text{ cm}$), four layers PCB and can handle currents of up to 30A - Table 3.1. All components, beside the magnetic encoder, are on the top side for easier assembly and debugging. Complete set of three boards costed less than 200 Euro to manufacture.

For the purpose of control between boards, communication is performed through CAN (up to 1Mbit/s). UART is only used for initial board configuration. CAN is quite robust, low cost interface, frequently used for automotive applications. It was originally used in the Ben G. Katz design. With data formatted as presented in the Table 3.2 they were able to accomplish stable communication for three Degrees of freedom (DoF) network at 1.2 kHz [4]. UART accepts inputs through Teletype (TTY) console and configuration is done through configuration menu - Box 3.1.

More information about driver and all the design files are accessible on dedicated GitHub page [27].

Table 3.2: CAN data format. As initially configured by [4].

Command(master)		Measurement(reply)	
Parameter	Resolution(Bits)	Parameter	Resolution(Bits)
Position Setpoint	16	Position	16
Velocity Setpoint	12	Velocity	12
Position Gain	12	Estimated Torque	12
Velocity Gain	12		
Torque	12		

Commands:

m - Motor Mode
c - Calibrate Encoder
s - Setup
e - Display Encoder
z - Set Zero Position
esc - Exit to Menu

Box 3.1: The motor support different modes and functionalities configurable through a terminal menu. Interface is as was created for Mini Cheetah Actuator. [4]

3.2 Mechanics - Computer Aided Design

Following previously made conclusions on importance of compliance and ability to leveraged it through optimised link structures, it was decided to use an environment which incorporates design, simulation and optimisation tools. Fusion 360 from Autodesk has all needed abilities, and the licence is free for student use as-well.

Inspiration for the design of the gear train systems was taken from SOLO [7], as discussed in Section 2.2. They have presented detailed manuals on how to manufacture needed parts and assemble the whole robot. Most of the components can be bought or 3d printed.

This design was inspired by their solutions. Following conclusion from Section 2.3, it was chosen to use mostly 9:1 gear ratio, only for ankle 10.8:1 stage was used (the ankle is being driven by a less powerful T-Motors 5008) - Table 3.3. The gear train is designed as a pulley-belt system and uses six AT3 belts and twelve pulleys. Custom designed ten-teeth

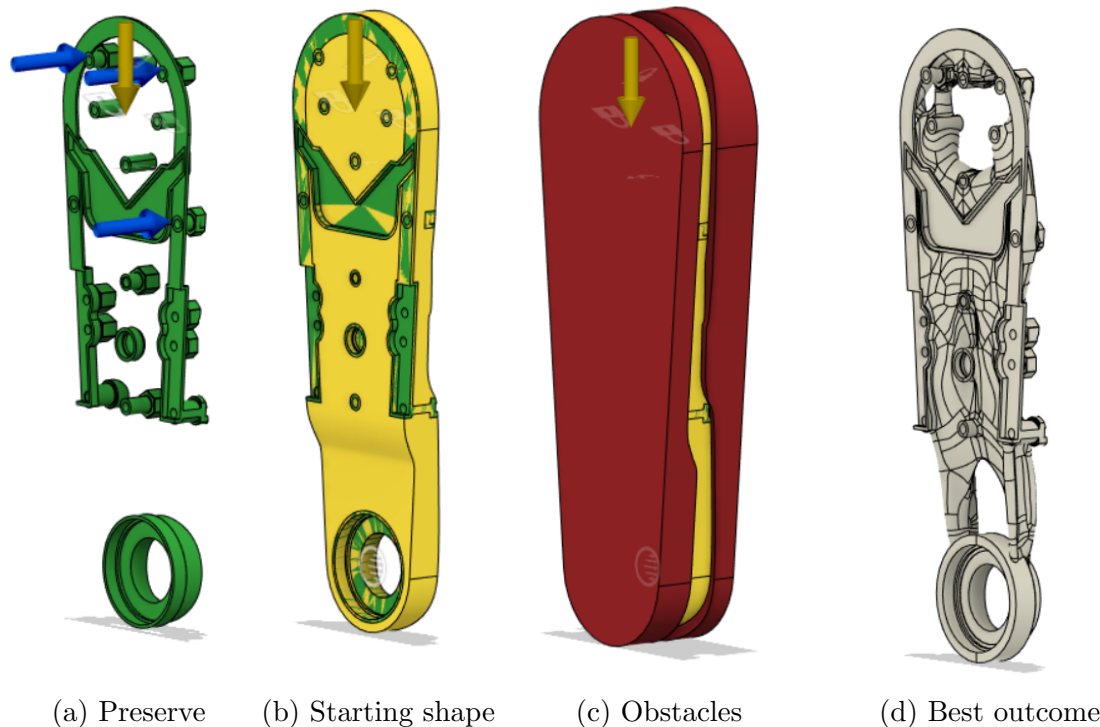


Figure 3.2: Generative design setup for the robot leg - Thigh Link. Loads (blue arrows) presented in 3.2a are just an example of one of the load configurations for the Generative Design Task.

pulleys [7], motor add-ons for holding the shafts, along several smaller shafts are the only parts that should be manufactured by the machine shop - Figure 3.4. The ten-teeth pulley requires custom form cutter, all information can be found in [28]. Since most of the parts will be 3d printed, design choice was to leave final adjustments to users because of the loose tolerances. For that reason, the screw adjustable tensioners were introduced to regulate the belts tension wherever necessary.

Table 3.3: Gear ratio for the actuators.

Actuator	Gear ratio
Hip	9:1
Knee	9:1
Ankle	10.8:1

Additionally, the springs should be placed in parallel to each joint in order to introduce the state bias for robot standing posture. Springs were chosen to only compensate static

torque and not to largely influence torque performance in other conditions. At ± 220 deg (spring limit - maximum rotation angle of each joint with springs introduced) maximum torque produced by the spring will be around 4% of the maximum required torque from the actuator. They should be positioned such that in vertical configuration - Figure 4.1b they exert zero torque, but once the leg is loaded (in contact with the ground) they should exert enough force to make the robot stand. Each spring would be holding 5N at 20mm force radius at joint positions of ± 90 deg, thus, not requiring any motor torques in order to keep the robot standing. Standard torsion springs were chosen for this purpose and load symmetry for the springs is achieved by putting one left-winded and one right-winded spring in parallel. Only one spring is active for each loading direction. Springs can be easily disabled by removing the supporting elements.

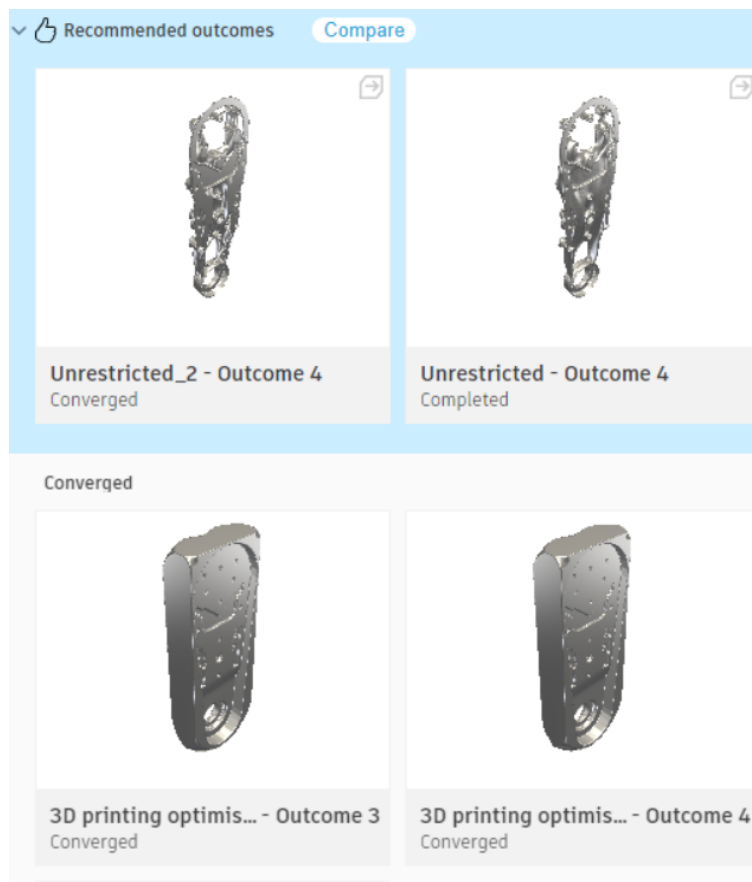


Figure 3.3: Generative Design outcomes.

The links design has been optimised based on the simulation results, under different loading conditions. Simulation was configured for Nylon materials (intended material to be used for 3d printing) with loads as calculated in (2.1). Displacement at required loads is desired, but limits introduced by the control bandwidth requirements must be respected. Currently it was not possible to simulate all material non-isotropic properties introduced by the 3d printing (layer adhesion, printing direction ...), and for that reason the loads indicated in the simulation are increased compared to the calculated loads. Further testing would be required to measure deviations and correct the simulation, but it is not intended to be a part of this thesis. Simulation setup example and some of the results can be seen in Figure 3.2 and Figure 3.3 respectively.

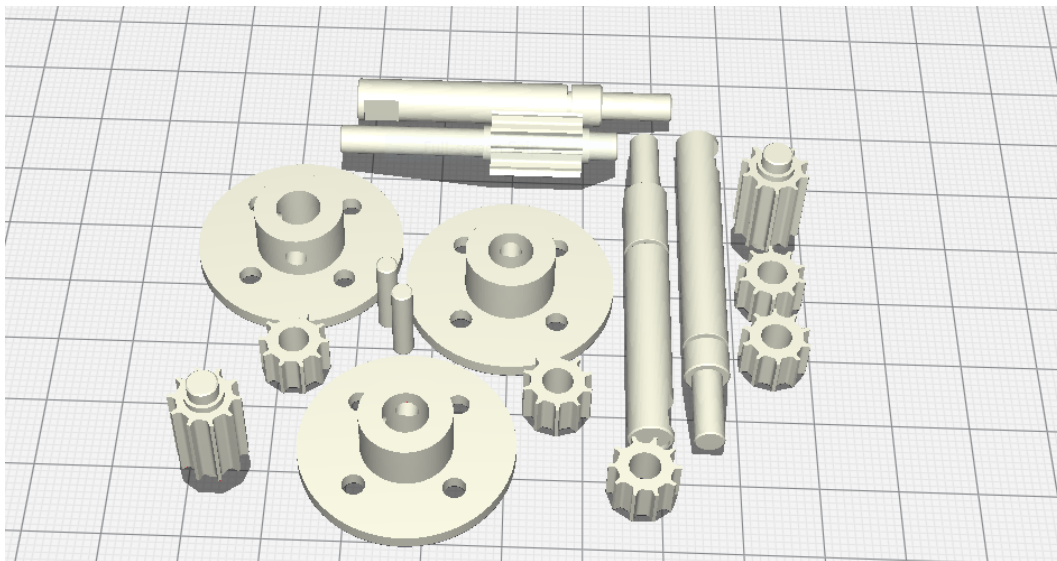


Figure 3.4: Parts to be manufactured in a mechanical shop.

In previous Sections all prerequisites have been established: motors choice, gear ratios and its implementation strategy, PCB design, number of links and robot size choice. Following discussion of this section, the whole CAD design of the leg was completed as shown in Figure 3.5.

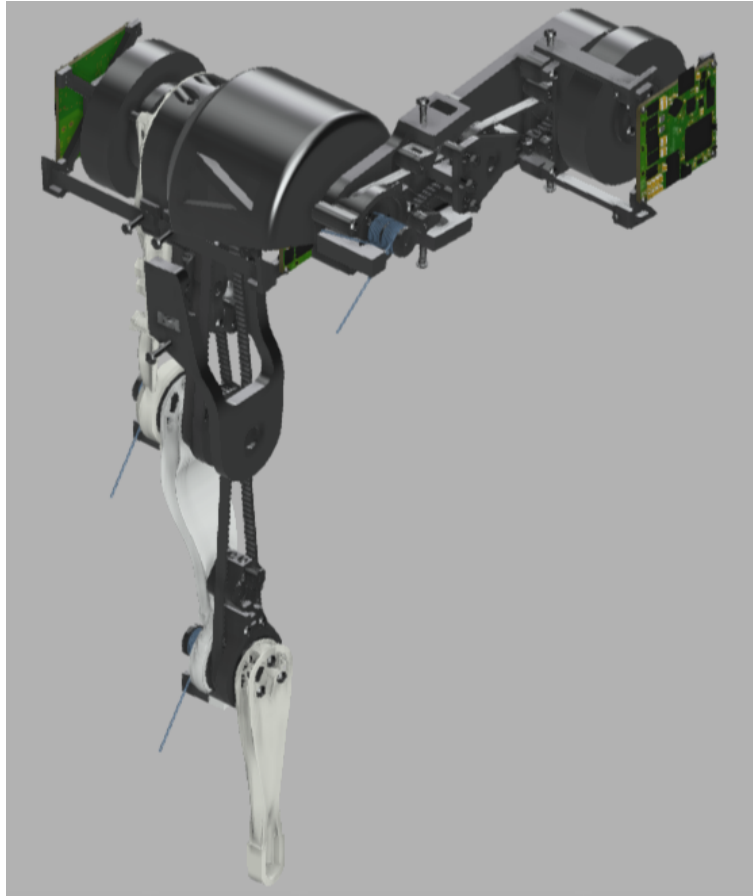


Figure 3.5: Finished CAD design of the whole leg. Three motors, through belt-pulley systems, independently control each joint: Hip, Knee or Ankle.

3.3 Software - Low-level Controller

Low-level control (motor driver controller) was implemented on STM32F4 micro-controller using STM32CUBEIDE software. Some of the key interface components required for the control are depicted in Table 3.4. Following functionalities were implemented within:

- Poles determination procedure by sensing number of poles through performing continuous electrical rotations until one mechanical rotation is achieved
- Ordering phases - sensing the direction of rotation with initial configuration and flipping the phases if it doesn't match expected [4]
- Position sensor linearization procedure [4]

- FOC control
- Communication and control through CAN [4]
- Configuration with UART interface [4]

Table 3.4: Controller interface.

Configuration	Timings
Input	
A and B current sense + Voltage sense - 3 Analog to Digital Converter (ADC)s	2 μ s
Absolute position - within 360deg - Serial Peripheral Interface (SPI)	SPI 22.5 Mbit/s + 500kHz data update rate
Output	
3 PWM Controls	40kHz
Motor driver communication - SPI	2.5Mbit/s
Input & Output	
CAN	1Mbit/s
UART	-

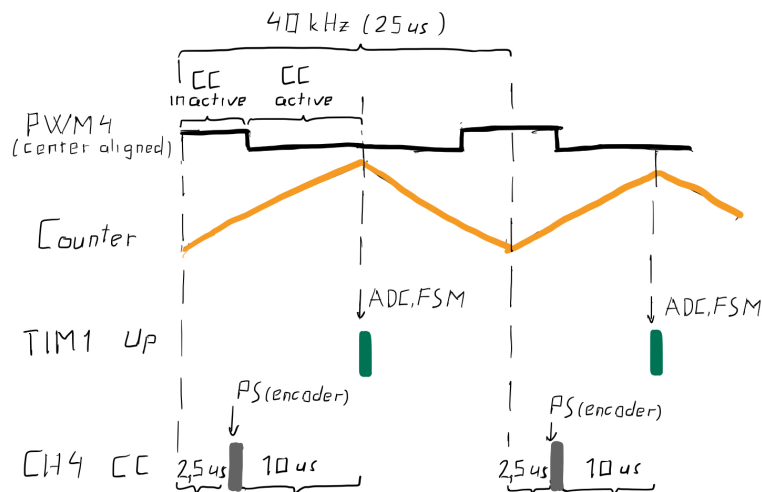


Figure 3.6: Low level controller events timings. Fixed interrupt schedule configuration. CC - counter compare; TIM1 UP - counter overflow event; CH4 CC - counter compare event of channel 4; PS - position sample; ADC - analog to digital conversion; FSM - Finite State Machine running.

Low-level FOC control implementation is similar to the Matlab simulation setup represented in Figure 3.7. Control is performed in the transformed domain of motor phase

currents (Clark and Park transforms applied to the values of Phase A and Phase B currents). Previously, based on received commands of desired motor speed and position, using Proportional (P) controller, desired currents are calculated. Proportional-integral (PI) controller is used for current regulation in transformed domains. The Controller outputs D and Q axis voltages (two-dimensional representation of three-dimensional motor voltages - achieved through Park and Clarke transformations) that should be applied to the motor as shown in Figure 3.8. After inverse Park and Clarke transformation and Space Vector Modulation (SVM) they are being transformed into duty cycle signals. Matlab simulation setup allows further analysis of different control strategies. Transformations and vector control are implemented based on [29].

The main control loop routine is activated every 40kHz (period $25\mu s$) based on the Timer 1 Update event interrupt - Figure 3.6. This interrupt occurs every counter overflow. On this event the ADCs are triggered. For a valid data, ADCs should be used only while the low half-bridge mosfets are active, since shunts for current measurements are placed on the bottom side of the bridge [30]. It takes around $2\mu s$ to return results, which are automatically stored in memory by raising an interrupt for Direct Memory Access (DMA) module after data conversions was finished. Beside the time for activation, dedicated ADC modules are not using any other CPU time. In the meantime, encoder data can be processed. In order for the encoder data to be available at the right time, it is sampled and pre-processed $10\mu s$ before the Timer 1 Update interrupt, which is ensured by configuring an additional channel interrupt. It takes a bit less than $8\mu s$ to perform data reception from the encoder and pre-process it. This ensures that the position data is available before the main processing routine has started. In the main processing routine, the finite state machine decides between control, communication and idle loop.

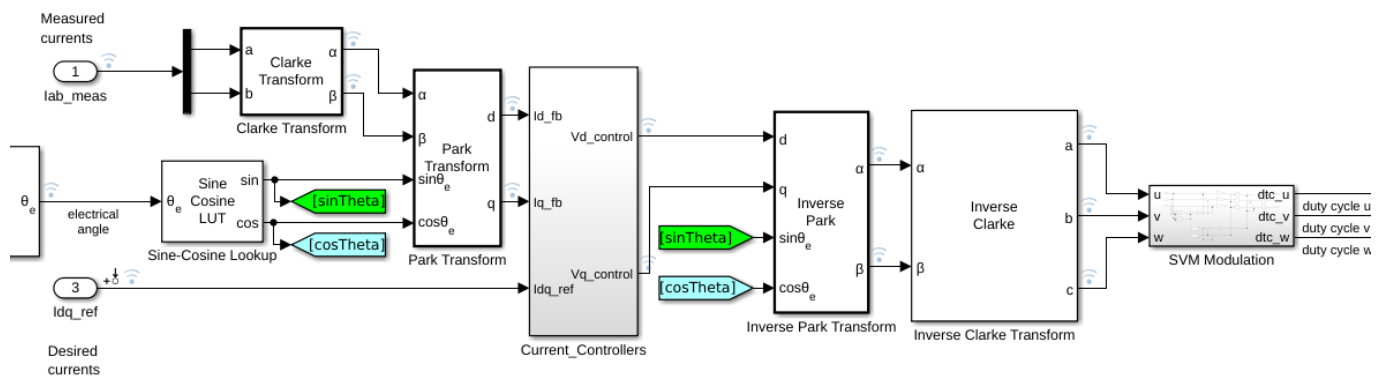


Figure 3.7: Low-level controller regulating in the transformed domain of DQ currents (DQ is a two-dimensional representation of the three-dimensional motor phase currents - achieved through Park and Clarke transformation). As input, measured A and B phase currents, along side the electrical rotor angle and desired DQ axis currents are received. Output is duty cycle value for each of the motor phases.

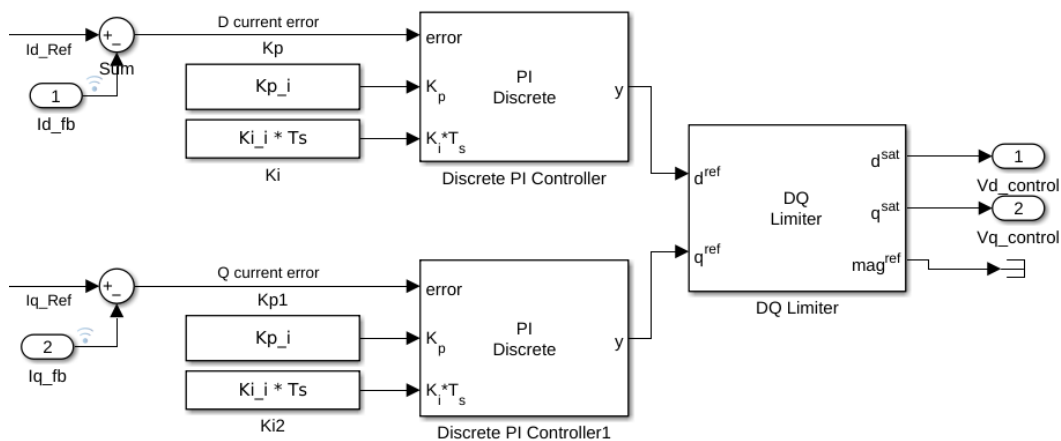


Figure 3.8: PI controller of inner current loop. Based on difference between provided reference and sensed D and Q axis current values, errors are calculated. DQ limiter is ensuring that control D and Q axis voltages do not exceed voltage level of the power supply.

Chapter4

Motion Testing

4.1 Test-stand

It was decided that the initial motion tests should be constrained to a single leg planar movements. This is sufficient for verification and further optimization of the key ideas presented in this thesis. Afterwards, it is planned to adopt solutions to the whole quadruped robot dynamics.

Idea is to benchmark jumping maneuvers in order to achieve better compliance, back-drivability, position redundancy, torque bandwidth and optimize other new aspects of this design. The test-stand allows jumps up to 1m in height. The mechanical design was inspired by Open Dynamic Robotic Initiative Test-stand [31], however some changes were introduced for adaptation to our design requirements as shown in Figure 4.1.

The leg is powered through an XT90 connector, and additionally has CAN interface wires exposed. CAN was being utilized for between controllers and high-level command communication. On Figure 4.1 one can see exposed connections on the left side of the test-stand.

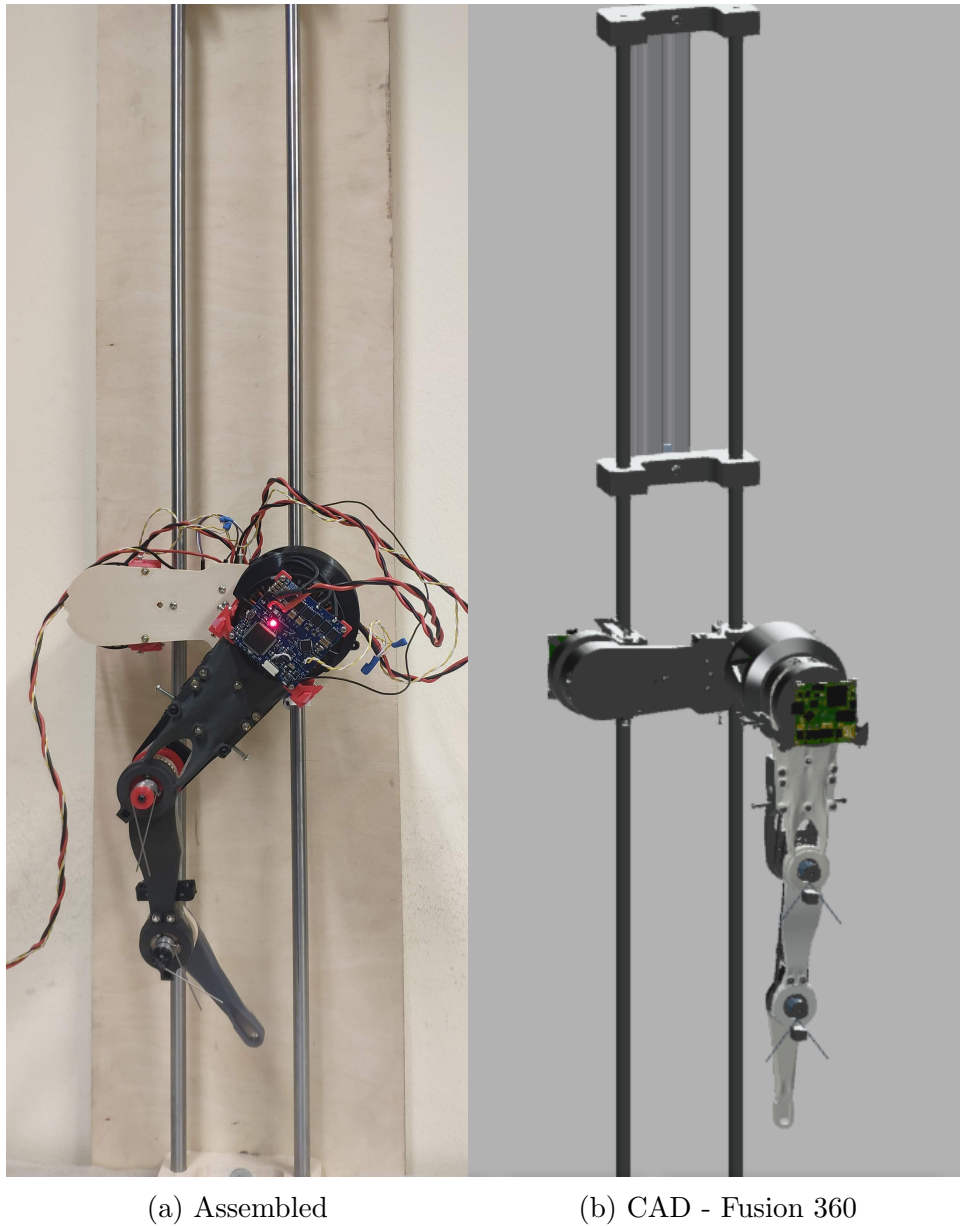


Figure 4.1: Leg test-stand.

4.2 High-level Control and Simulation

Among different frameworks for robot control and simulation, Crocoddyl [32] was chosen for its efficiency, speed (200 Hz computation frequency for walk-2d, one of the motion strategies benchmarked in the original paper, executed on i7-6700K with 4 threads) and because of features that it provides. It is an optimal control library for robotics. It uses solvers based on novel Differential Dynamic Programming (DDP) algorithms, efficient derivatives and Euclidean and non-Euclidean geometry via Pinocchio [33]. Additionally, it offers numerical differentiation and can handle autonomous and non-autonomous dynamical systems. It provides python binding for ease of prototyping. It is cache and multi-thread friendly and most important - fully Open-Source.

In order to experiment and generate different movements, simulation was used first. Realistic model of the robot was created in Unified Robot Description Format (URDF) format - Figure 4.2, with inertia configured as calculated by CAD design software.

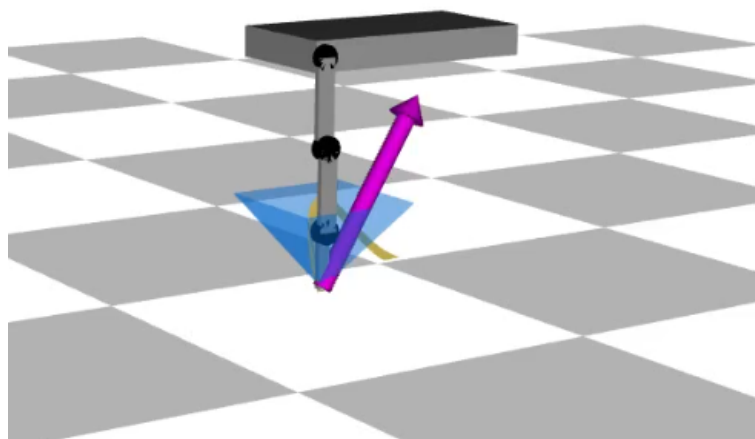


Figure 4.2: Gepetto-GUI (Simulation tool used in Crocoddyl library) with representation of a robot. As on the test-stand, movement is constrained to the planer motion in xz -plane.

Alongside control and state bounds costs, which are common for all the phases, cost functions for optimal control problem of robot jumping are divided in four stages:

- Take-off (contact, biased to initial CoM and reference state costs)
- Flying Up Phase (CoM and State tracking cost for height change)
- Flying Down Phase (biased to initial CoM and reference state costs)
- Landed (contact, biased to initial CoM and reference state costs)

Initial CoM is calculated in reference state. Reference state represents the initial robot position - a standing position when all links are vertical as shown in Figure 4.2. Robot interaction with environment is modeled through contact dynamics:

$$\begin{bmatrix} M & J_c^{-\top} \\ J_c & 0 \end{bmatrix} \begin{bmatrix} \dot{v} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \tau_b \\ -a_0 \end{bmatrix}$$

Where [34]:

- $q \in \mathbb{R}^{n_q}$ - joint angles
- $v \in \mathbb{R}^{n_v}$ - joint velocities
- $M(q) \in \mathbb{R}^{n_v \times n_v}$ - mass matrix
- $\tau_b \in \mathbb{R}^{n_v}$ - joint torques
- $J_c(q) = \frac{\partial \phi(q)}{\partial q}$; $J(q) \in \mathbb{R}^{n_f \times n_v}$ - contact Jacobian, and $\phi(q)$ is function mapping joint angles to contact point positions
- a_0 - desired acceleration in the constraint space

For contact model one can use holonomic scleronomous constraints on frame placement ($\phi(q) = 0$). In Crocoddyl, forward dynamic problem is being solved as:

Given q, v, τ compute $\begin{bmatrix} \dot{v} \\ -\lambda \end{bmatrix}$

It is defined as optimal control problem through constraints, using \dot{v} , λ as minimisation variables [32].

Calculations are performed with spatial algebra (angular-before-linear order notation). Where velocities $\hat{v}_O \in V^6$ and forces $\hat{f}_O \in F^6$ are represented as follows:

$$\hat{v}_O = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_{Ox} \\ v_{Oy} \\ v_{Oz} \end{bmatrix} \quad \text{and} \quad \hat{f}_O = \begin{bmatrix} n_{Ox} \\ n_{Oy} \\ n_{Oz} \\ f_x \\ f_y \\ f_z \end{bmatrix}$$

Where:

- ω - rotational part of the speed
- v_O - translational part of the speed
- n_O - orientation dependent forces
- f - orientation independent forces

To introduce contact cost function, linearized friction cone is used. The cone linearization is adopted for 2D case - xz-plane (leg test-stand motion is constrained to only vertical movements). Only columns corresponding to x and z of linearized friction cone matrix are used for residual computation. This is relevant representation indeed, if one to ignore any friction introduced along y dimension. It is a valid assumption because of the initial constrains imposed on the robot movement.

Optimizer task setup was able to solve the problem in around 200 iterations - Figure 4.3. It was found that beside costs, quite important parameters for convergence was time discretization and initial guess for control and state trajectory. Solver time step was configured to be twice as small as numerical integration time step. Initial state was taken as a guess

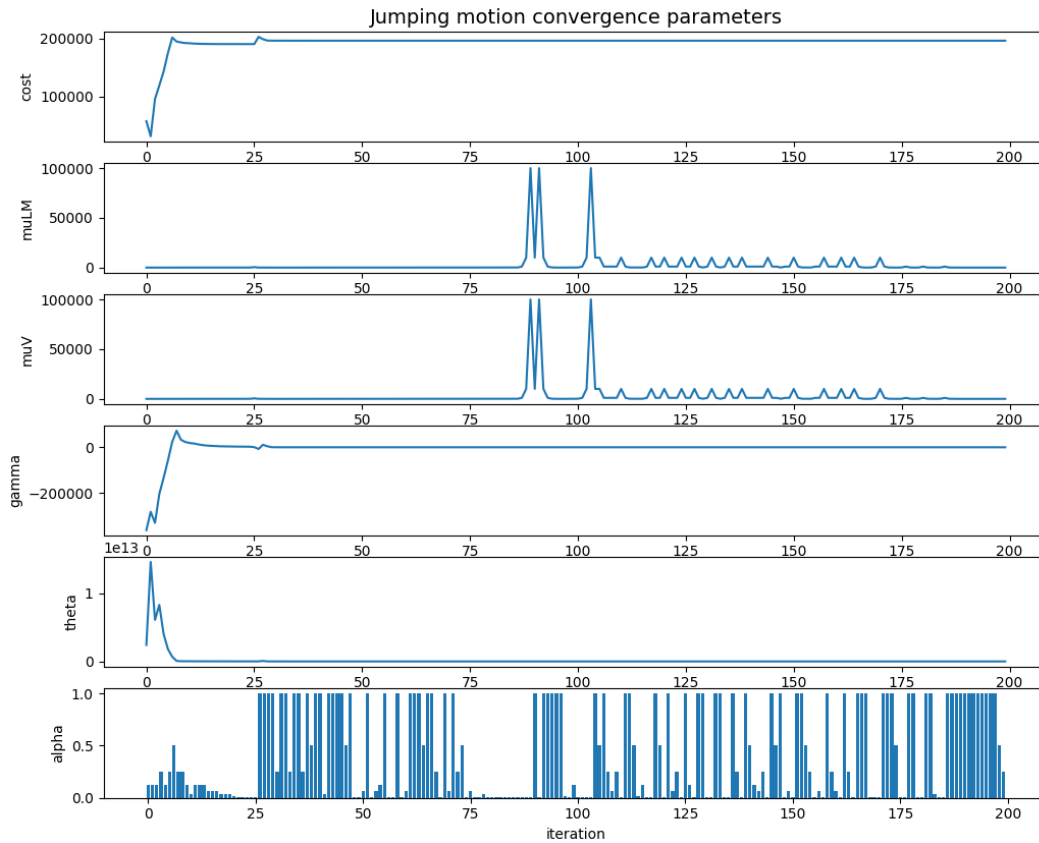


Figure 4.3: Jumping motion convergence parameters through optimizer iterations.

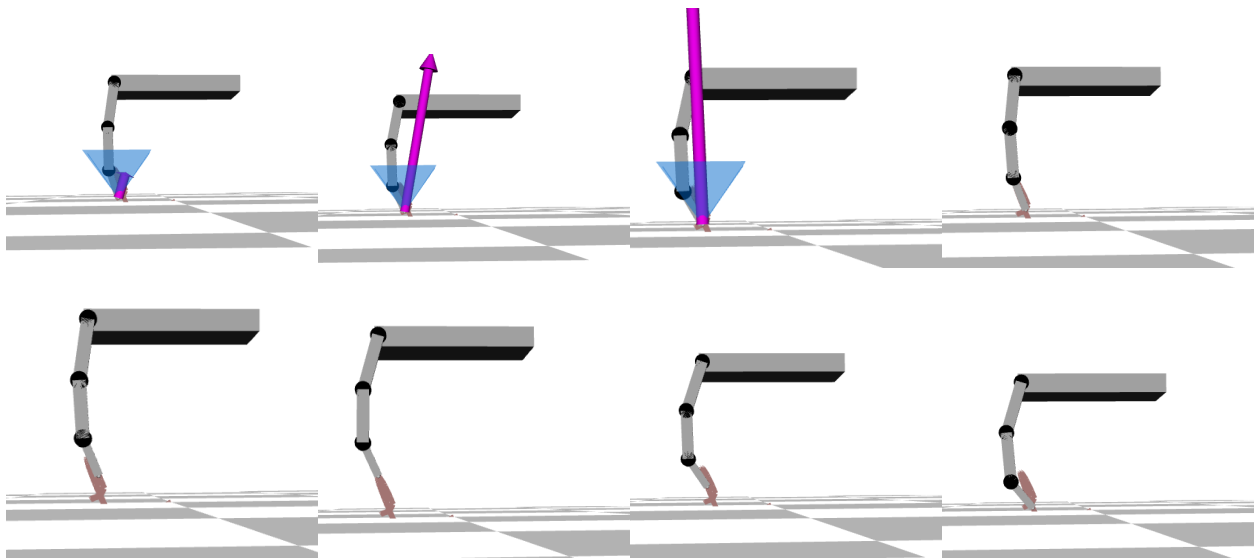


Figure 4.4: Simulated jumping motion.

for state trajectory. Guess for control trajectory represents the quasi-static control for the configured state trajectory. Visual inspection of generated motion - Figure 4.4 verifies that

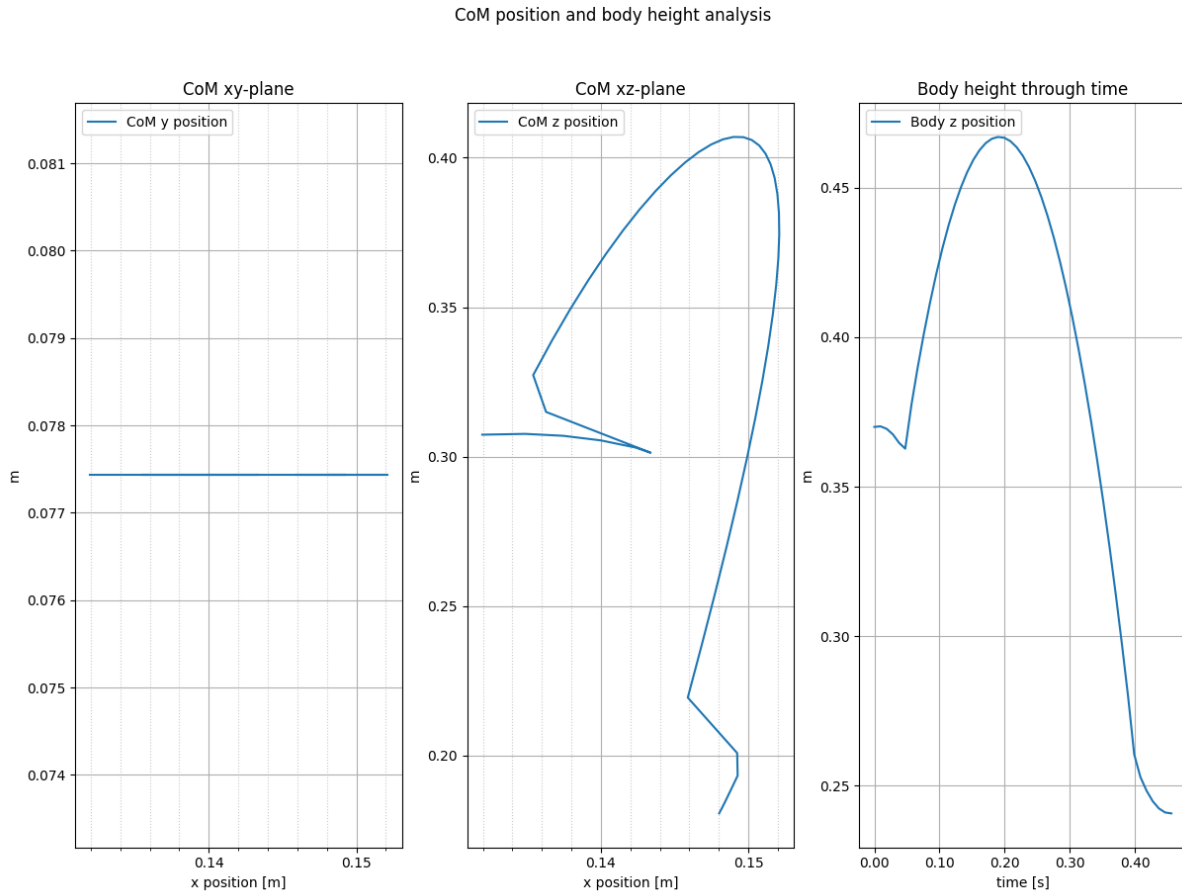


Figure 4.5: CoM position and body height of simulated jumping motion.

calculated dynamics corresponds to the jumping motion. Jump of around 10cm was achieved - Figure 4.5. Trajectory is limited by the torque abilities of the actuators - Appendix B, Figure B.1.

4.3 Motion Execution

Control of the leg test-stand was tested in two ways:

- Controlling through Arduino with CAN-bus shield. When it comes to the Arduino environment, it cannot guarantee high frequency and real-time processing, however, up to 1kHz open loop control can be achieved. The control was successfully executed while robot positioned in the air, by following the jumping motion data acquired through

simulation. Arduino provides an easy to use setup which we believe will help people get familiar with the test-stand more easily.

- Python code running on the computer. Additional hardware is required - a board capable of re-transmitting messages between Universal Serial Bus (USB) and CAN interfaces. The idea for the hardware solution came from Ben G. Katz USBtoCAN implementation [35]. However, for fast prototyping purpose STM32F407 Discovery board with the breakout board of CAN transceiver were used. Speed limitation in this case is USB communication speed, which is set to 256000 baud rate

Several tests were used to check the leg performance:

- Step response from resting position to a random joint configuration
- Execution of the simulated jumping motion, while robot position in the air
- Drop from 0.4 *m* height
- Tracking sinusoidal trajectory

When setting the next desired position, user can chose P gains for position and velocity of the motion. Tests showed that the step responses of all actuators have settling time within 300 *ms* as shown in Figure 4.6. Position gain adjusted to 100. User can receive different feedback data. It is intended to keep CAN response short (5 bytes) as shown in Table 3.4, but the different kind of data can be made available through smaller adaptations of the code. As example of alternative sensed values, knee actuator Phase A and Phase B currents can be found in Appendix B, Figure B.2. Further mostly velocity, position and current along Q axis (representing motor torque) will be shown on the plots.

In order to check bandwidth capabilities of the designed leg, sinusoidal trajectory tracking task was configured. It was chosen to use sine wave with 1 *rad* amplitude. Results for all the joints were similar. Average tracking error was 0.2 *rad* at 1 *Hz* frequency and larger tracking errors starting at 2 *Hz* frequency. Performance at 1 *Hz* is shown on Figure 4.7.

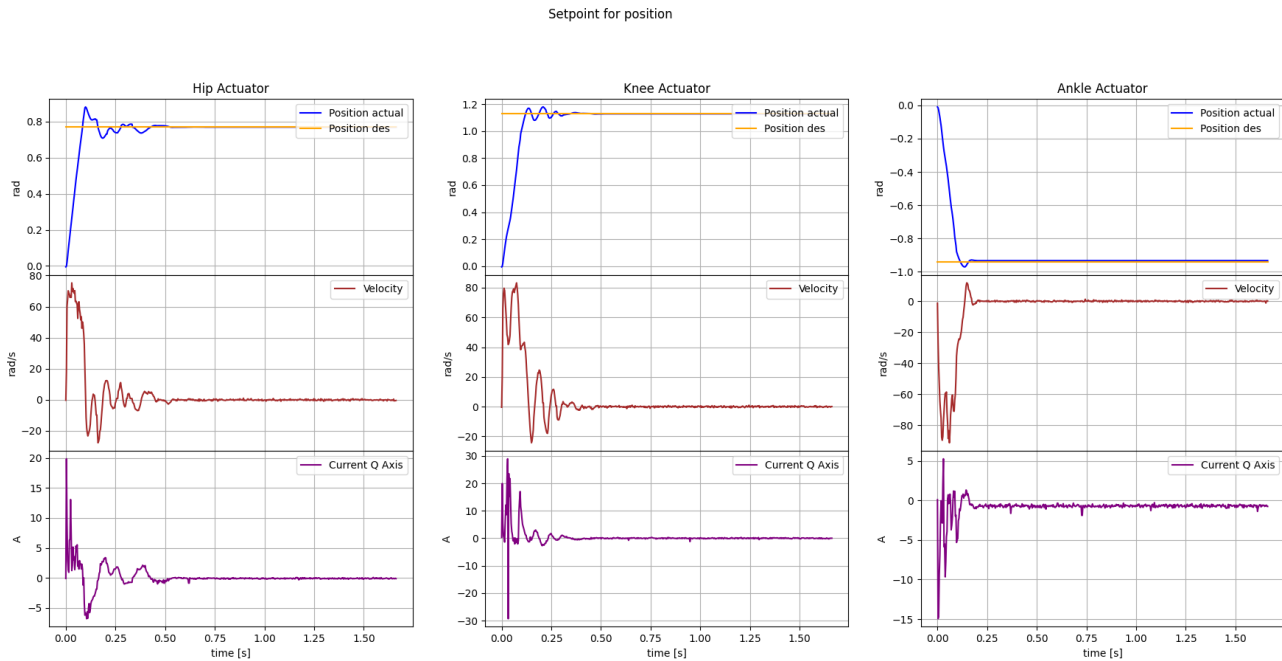


Figure 4.6: Setpoint position step from zero to the random robot joint configuration.

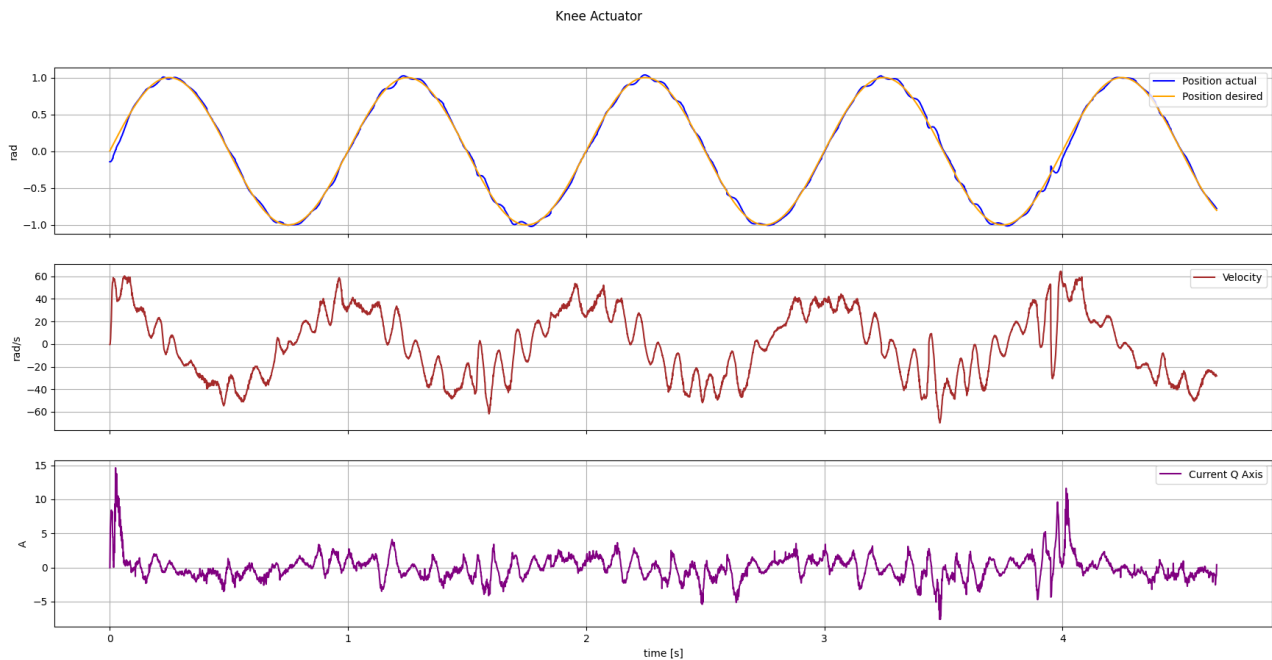


Figure 4.7: Tracking of sine wave with 1 *rad* amplitude and 1 *Hz* frequency. Average position error was 0.2 rad. Hip and Knee actuators performed similarly.

Simulated jumping motion was tested while robot positioned with all links vertical and touching the ground. Similar to how initial position was configured for simulation, as shown

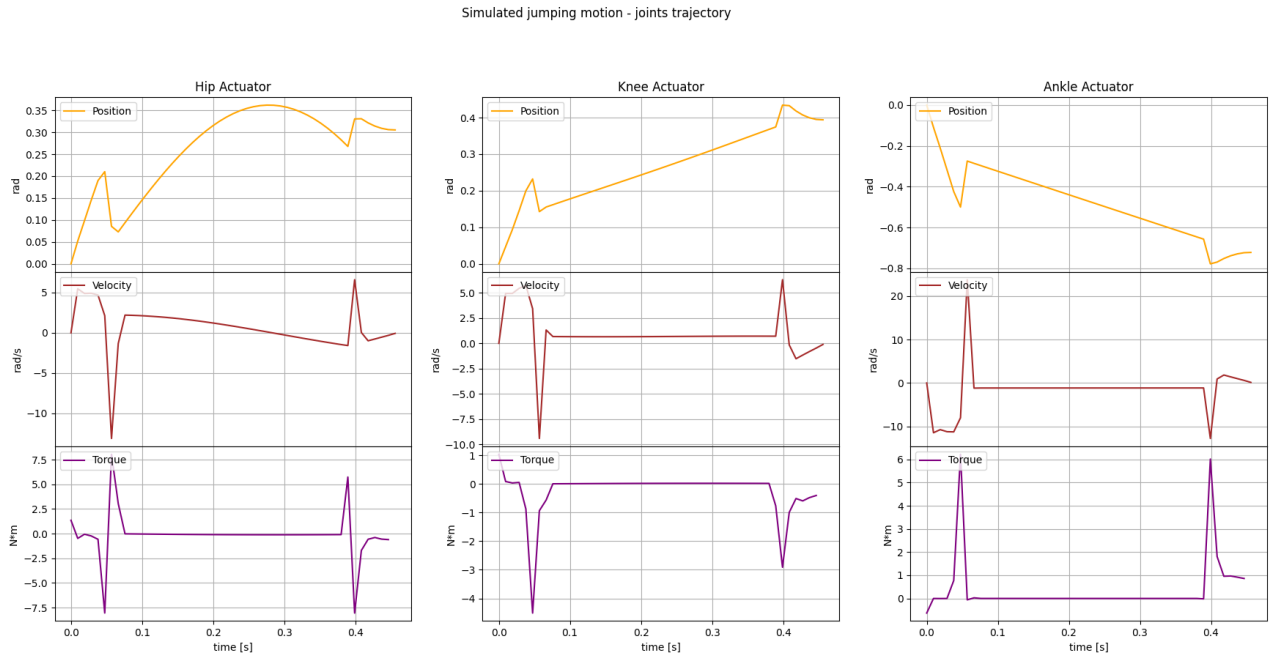


Figure 4.8: Simulated jumping motion trajectory for Knee, Hip and Ankle joints.

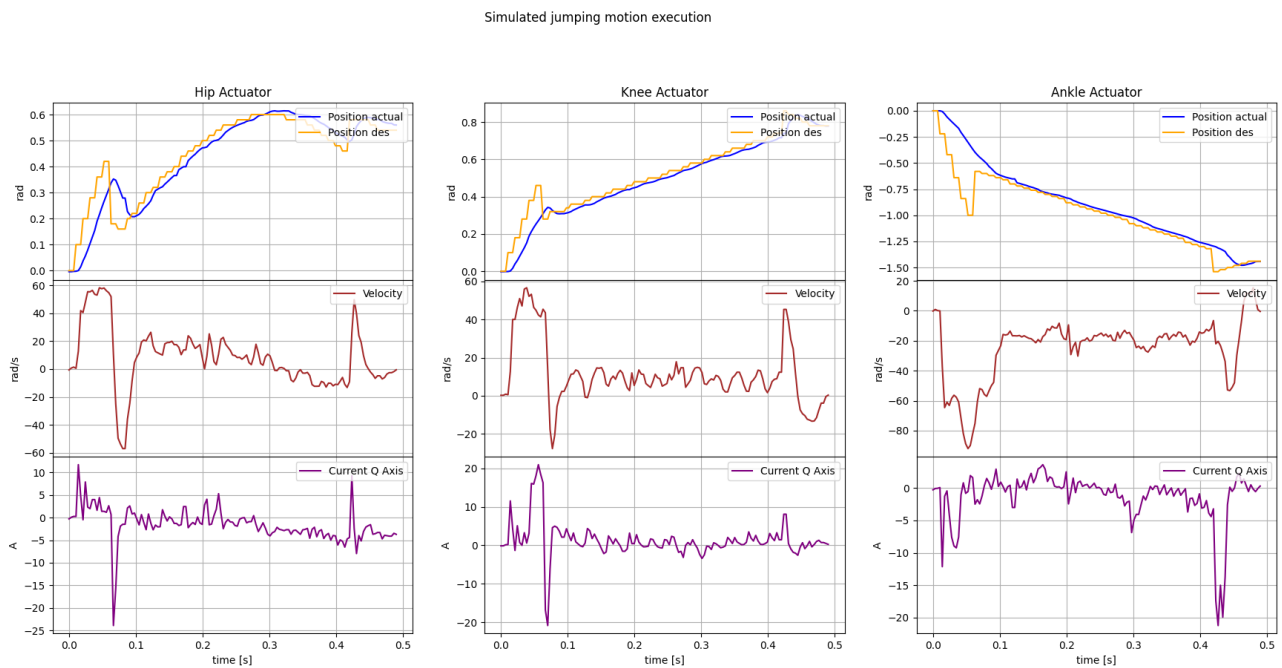


Figure 4.9: Execution of simulated motion on the robot with position tracking. Time discretization adjusted so it matches the simulation time step.

on Figure 4.2. The jumping motion has been stored as an array of data points, where each point contains desired speed and position for particular joint, at that instance of time. One

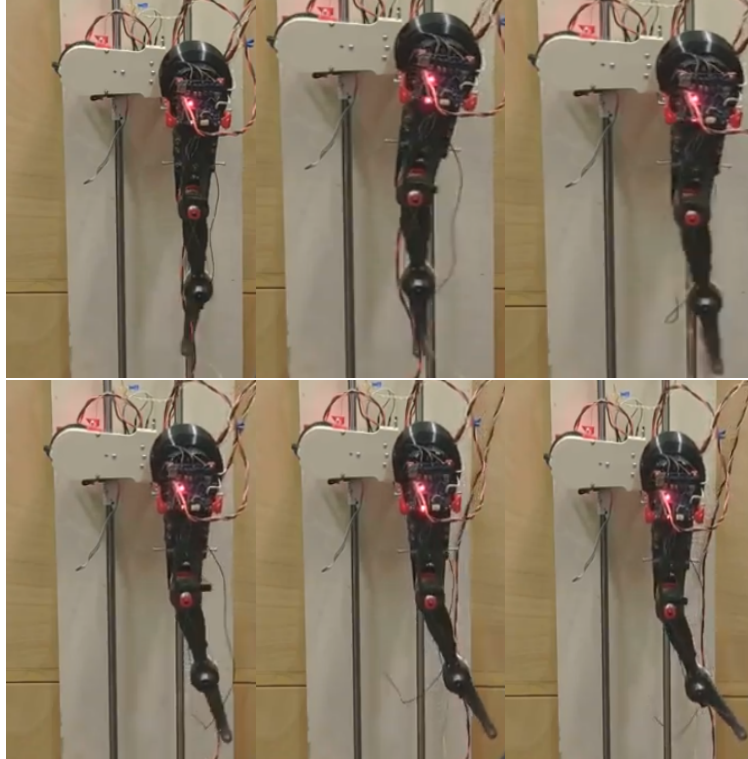


Figure 4.10: Performing simulated motion in the air with position tracking controller.

can compare simulated motion presented in Figure 4.8 and data from motion executed on the robot, shown on Figure 4.9. Robot was able to track the reference trajectory, however, better control and simulation should be implemented to reduce tracking error and to perform successful jump. Later, robot was also tested in the air, execution is depicted through series of images in Figure 4.10

Benchmarking backdrivability and compliance with jumping motions is planned for the future. Currently, it was decided to perform simpler tests. In particular, drop from 0.4 m height was performed, while robot positioned in landing configuration taken from simulation. In this scenario, with position gains set to 100, Hip stiffness was 6.14 Nm rad^{-1} Knee stiffness was 5.4 Nm rad^{-1} and Ankle stiffness 3.8 Nm rad^{-1} . For stiffness approximation, first, relation between motor current and force exerted on the motors are measured, data is shown in Table 4.1. Afterwards data from Figure 4.11 was taken to calculate ratio between the torque and the displacement measured at the motor. It is important to note that

stiffness calculated does not represent stiffness of the robot structure, it just depicts the motor stiffness, since both, displacement and torque, are calculated based on the motor feedback values. Stiffness can be controlled by further adjusting the gains. In future, it is planned to improve control strategy for better controllability of motor stiffness (compliance).

Table 4.1: Torque to current ratio for motors. It was determined experimentally by applying external force and measuring motor currents.

Actuator	Torque to current [$N m A^{-1}$]
Hip	0.063
Knee	0.063
Ankle	0.046

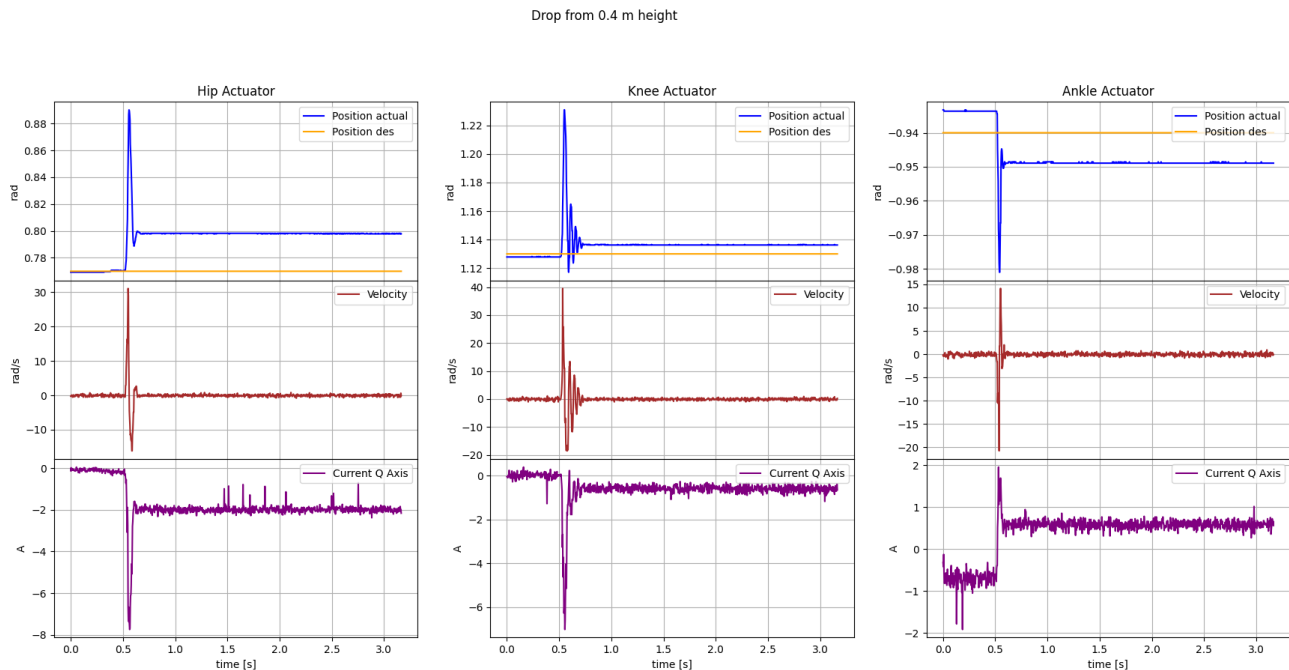


Figure 4.11: Leg performance when dropped from 0.4 m height. Data acquired is used for stiffness (compliance) and backdrivability approximation

4.4 Shown potential and future enchantments

The leg performance has been presented in previous chapter. The tests were mostly performed in the air. Results are showing great potential, but yet some mechanical and control

improvements are required to benchmark the foundational concepts of this thesis. Some of the verified benefits of this design:

- Possibility to achieve lightweight three-link structures with independently controlled joints.
- Compliant link design and the proper force distribution between elements ensures that the parts do not break under high impacts (the links rather comply than break).
- Change in the robot link configuration distributes forces differently among joints. For 0.4 m drop testing configuration from simulation is used to ensure the most optimised distribution, but it should be further improved.

Next short-term improvements are planned in the following order:

1. Design of a separate stand for testing of two stage belt pulley system in order to find optimal belt tension to prevent slipping and if needed, adjust pulley parameters for desired forces. Calculate the efficiency of such gear system and experiment with other gear configuration.
2. Comparison of simulation setup for link elements and achieved results. How 3d printing materials are influencing the properties and how simulation can be adjusted. The two important criteria should be: measured compliance (displacement) at different loads with different directions and maximum force that the links can withstand.
3. Motor test-stand for the purpose of checking the dynamic properties, plotting load-speed characteristics and understanding how temperature influences motor parameters. It is also important to check motor efficiency and power regeneration ability under different breaking and running modes. All in order to improve motor control.
4. Test and redesign, if required, for the robot links to accommodate previous findings.

5. Improvement of robot simulation to accommodate walking, running and further optimize jumping.
6. Accommodate treadmill under the robot leg on the test-stand to allow testing of running and walking. Develop different strategies of maintaining small height fluctuations of the CoM as explained in Figure 1.3 and check how contact forces can be adjusted through the change of the robot configuration as shown in Figure 1.4. Achieve maximization of forward force.
7. Measure structural compliance of the whole robot and how it affects its control bandwidth.
8. Introduce compliance in the Higher-level control, with the goal of achieving easier basic movements such as walking or standing (leveraging environment and the robot mechanical properties to guide robot movement, thus reducing amount of motor activity for these basic gaits). Compliance in this design is introduced through: (1) mechanical robot link design, (2) additional compliant mechanical elements such as springs in parallel, (3) software.

Chapter5

Conclusion

Project has modular design structure and all achievements are available Open-Source. Development loop has been established (motion simulation, motion execution, improvements...). Further work can be carried through enhancements of following modules:

- Electronics - controller design files.
- Low-level Software - Motor driver files.
- Mechanics - leg design files with robot Generative Design optimisation setup.
- High-level software - Motion simulation environment.
- Python control scripts.
- Arduino control scripts and all additional and latest software, files and documentation available on GitHub project page.

Links to individual project pages provided in the Appendix D.

The project represents an template for the legged robotics platform, which has been created taking into account relevant and latest achievements in the legged robotics domain. Merged with ideas taken from the nature, such design is expected to provide better dynamic capabilities, important for all domains of robotics - space exploration, industrial applications, research... All foundational ideas have not been tested yet, since the design requires

further improvements. However, first prototype, incorporating all important concepts for further work was designed and produced within the time allocated for this thesis work (in total a period of eight months). Development of such a leg usually takes much longer and requires multidisciplinary team to cover variety of skills - mechanics, electronics, software... Design verified several important concepts, such as feasibility of the three-link structure for legged robotics, advantages and abilities of in-links built compliance when it comes to impact mitigation and positive aspects of redundancy achieved with introduction of additional link for the robotics leg. It represents important foundation for future work. Shown potentials and planned enhancements were described in more details in Section 4.4. Such design is expected to introduce new ways of robot control in order to better accommodate compliance and take advantage of it.

AppendixA

PCB design considerations (general guidelines)

Based on the article of Eric Bogatin on Breaking Bad Habits in PCB Design: <https://www.signalintegrityjournal.com/blogs/12-fundamentals/post/1207-seven-habits-of-successful-2-layer-board-designers>

- Infill on the top layer in error-prone since its likely to cause problems if left unconnected to the ground or to power line
- Use thick traces - 6 mil for signals and thicker traces for powers (same consideration when it comes to vias - bigger vias should be used for power)
- Try to route signals only on one layer, use straps when routing in the other layer

Importance of decoupling capacitors, some important conclusions can be found in Analog Devices Application notes: <https://www.analog.com/media/en/training-seminars/tutorials/MT-101.pdf> Also, its important to understand the possibility of resonance effect due to different capacitors used in parallel (not always the case, in many application notes component manufacturer suggest most often to use 100nF and 10uF in parallel with smaller capacitor closer to the pin): <https://electronics.stackexchange.com/questions/327975/capacitance-vs-frequency-graph-of-ceramic-capacitors> More scientific approach regarding general guidelines on EMC considerations for capacitor can be

found in the lecture by Tzong-Lin Wu, Ph.D - EMC Considerations for DC Power Design:
<http://ntuemc.tw/upload/file/20120419205619a4fcf.pdf>

Importance of designing current return paths, more about that in the article Design Tips by Bruce Archambeault: <https://www.emcs.org/acstrial/newsletters/fall08/tips.pdf>

Twisting all the cables ensures that induced currents are being cancelled or reduced for possible sources (power cables can be the source of induced magnetic fields which can influence other parts of the circuit) as well as other cables which can pick up the undesired signals.

Ferrite bead are great for noise reduction in power lines. Understanding of how they work might be crucial for proper design. More information can be found in the article Ferrite Bead Demystified by Jefferson Eco and Aldrick Limjoco: <https://www.powersystemsdesign.com/articles/ferrite-bead-demystified/95/10031>

Thermal pads often need vias and there are other considerations which should be respected for their placements on the PCB, for general guidelines check Application Report - PowerPAD™ Thermally Enhanced Package by Texas Instruments -Application Report, SLMA002F – November 1997 – Revised August 2010.

Appendix B

Additional data on simulation and execution of robot motion

Simulated jumping motion - joints trajectory

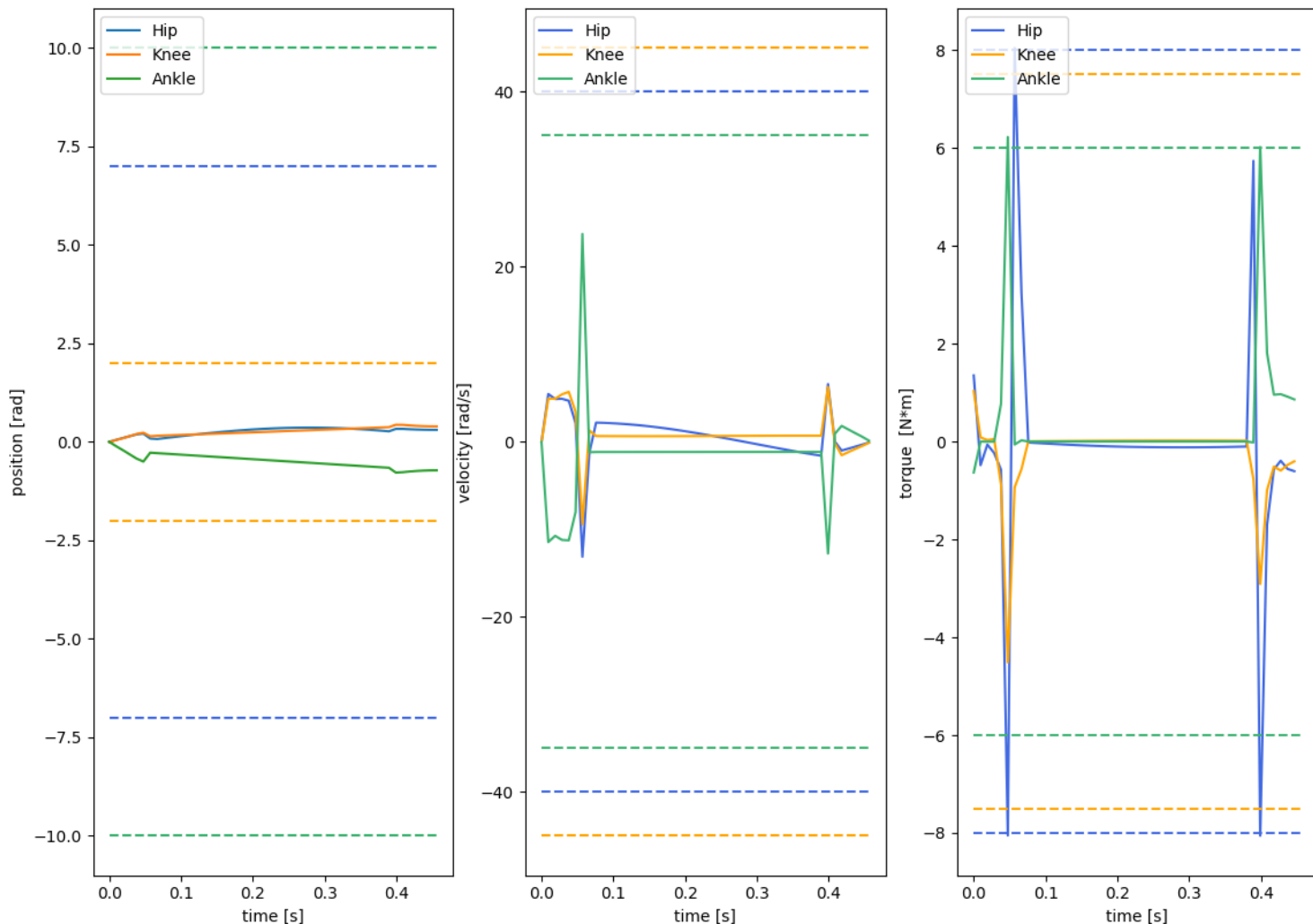


Figure B.1: Generated motion - position, speed and torques at the joints with bounds depicted. Performance is limited to the torque abilities of the actuators.

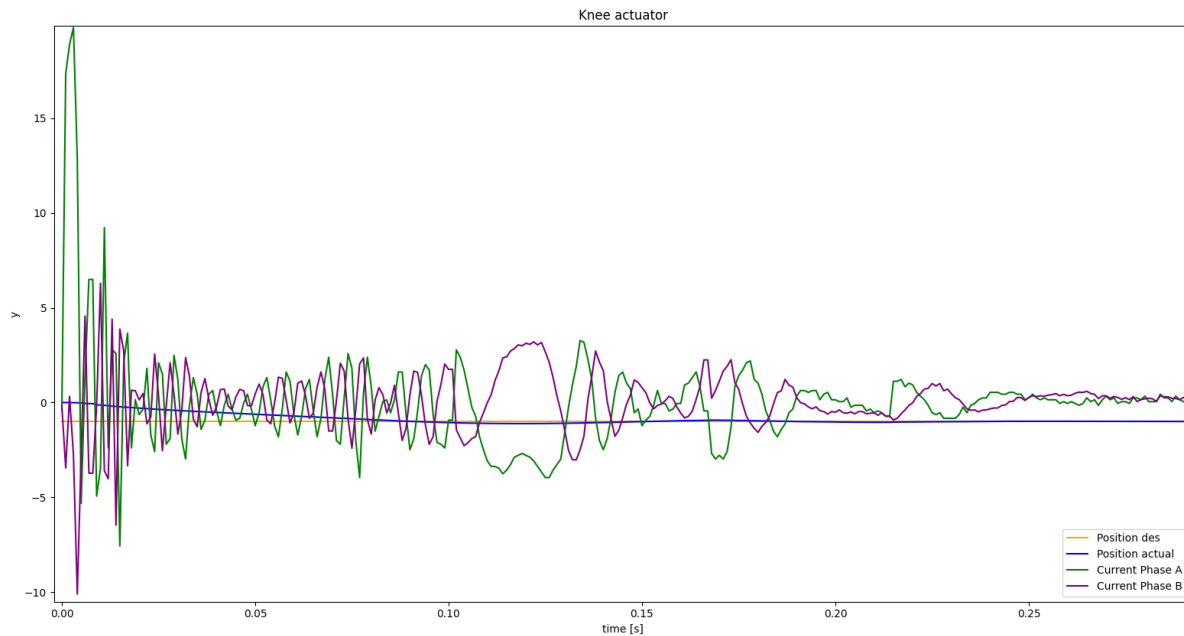


Figure B.2: Step response of the Knee actuator from actuator position of 1 *rad* to -1 *rad*, which corresponds to 18 *rad* of motor angle difference. Feedback of sensed position, Phase A and Phase B currents received. Position gain used for step response was set to 100.

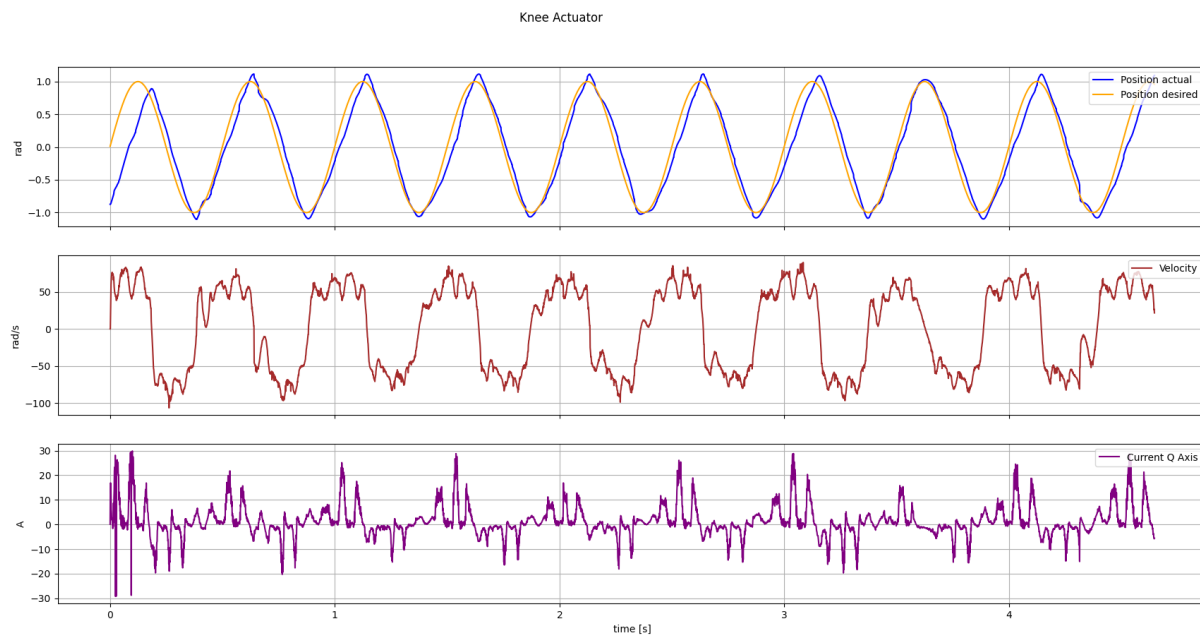


Figure B.3: Tracking of sine wave with 1 *rad* amplitude and 2 *Hz* frequency. Average position error was 0.2 *rad*. Other actuators performed similarly.

AppendixC

Hands-on lessons learned

Not categorised bullet-points to keep in mind:

- Small metal particles will be really hard to remove from the rotor of the motor. One needs to be really careful when disassembling, drilling or making any modifications to it.
- PCB holders are quite important for keeping pcb steady. However one needs to be really careful when attaching debug cables not to move it since it would be required to perform re-calibration because of position sensor displacement, which is located on the bottom of the board.
- It is not suggested to use robot with electrical power supply for any fast or loaded motions. Sometimes it can damage the device due to the return currents coming from the motors.
- Anti spark protection is important in order not to damage power supply. For that reason mosfets are added to the power lines with adjusted gate currents so it takes some time to switch on. Short explanation and simple schematics can be found here: <http://endless-sphere.com/forums/viewtopic.php?f=3&t=40142#p586436>
- Power can flow through the CAN lines. If one removes power connectors on all boards beside one, but CAN lines stay connected, the rest of the boards will be powered

through CAN, at least partially. This is not foreseen working condition, and might have unpredictable consequences when powering boards this way for longer times.

- There is an 3.3V output in debug connector. It is not suggested to use it as an input for the board supply. Some insights can be found here: <https://electronics.stackexchange.com/questions/571102/applying-voltage-to-the-output-of-turned-off-voltage-regulator-ldfm33pur/571205#571205>

AppendixD

Legged robot project page links

- Electronics - controller design files. [27]
- Low-level Software - Motor driver files. https://github.com/Legged-robot/motor_controller_driver
- Mechanics - leg design files with robot Generative Design optimisation setup. https://github.com/Legged-robot/mechanical_design
- High-level software - Motion simulation environment. https://github.com/Legged-robot/control_and_simulation
- Python control scripts. https://github.com/Legged-robot/usb_to_can_controller
- Arduino control scripts and all additional and latest software, files and documentation available on GitHub project page. [36].

Bibliography

- [1] Boson Dynamics. *Spot*. URL: <https://www.bostondynamics.com/spot>.
- [2] Gerardo Blede et al. “MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot”. In: Oct. 2018. DOI: 10.1109/IRDS.2018.8593885.
- [3] T. Touma et al. *Mars Dogs: Biomimetic Robots for the Exploration of Mars, from its Rugged Surface to its Hidden Caves*. URL: <https://agu.confex.com/agu/fm20/meetingapp.cgi/Paper/775032>.
- [4] Benjamin G Katz. *A low cost modular actuator for dynamic robots*. URL: <https://dspace.mit.edu/handle/1721.1/118671>.
- [5] Patrick Wensing et al. “Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots”. In: *IEEE Transactions on Robotics* PP (Jan. 2017), pp. 1–14. DOI: 10.1109/TR0.2016.2640183.
- [6] Benjamin Katz, Jared Di Carlo, and Sangbae Kim. “Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 6295–6301. DOI: 10.1109/ICRA.2019.8793865.
- [7] F. Grimminger et al. “An Open Torque-Controlled Modular Robot Architecture for Legged Locomotion Research”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3650–3657. DOI: 10.1109/LRA.2020.2976639.
- [8] Philip Arm et al. “SpaceBok: A Dynamic Legged Robot for Space Exploration”. In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019, pp. 6288–6294. DOI: 10.1109/ICRA.2019.8794136.
- [9] Marc Manz, Sebastian Bartsch, and Frank Kirchner. “MANTIS - A ROBOT WITH ADVANCED LOCOMOTION AND MANIPULATION ABILITIES”. In: May 2013. DOI: 10.13140/2.1.4617.1206.
- [10] Dogumentary TV. *Youtube - Dog running*. URL: <https://www.youtube.com/watch?v=SGAbKSVLb-w&t=442s>.
- [11] Raphael Deimel and Oliver Brock. “A compliant hand based on a novel pneumatic actuator”. In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 2047–2053. DOI: 10.1109/ICRA.2013.6630851.

- [12] *Wikipedia - Impedance Control*. URL: https://en.wikipedia.org/wiki/Impedance_control.
- [13] Marco Hutter et al. “Toward Combining Speed, Efficiency, Versatility, and Robustness in an Autonomous Quadruped”. In: *IEEE Transactions on Robotics* 30.6 (2014), pp. 1427–1440. DOI: 10.1109/TR0.2014.2360493.
- [14] Sangbae Kim. *RI Seminar: Sangbae Kim: Actuation, structure and control of the MIT cheetah robot*. URL: https://youtu.be/_90uxecV48g.
- [15] S. M. Levin. “The icosahedron as a biologic support system.” In: *In 34th Annual conference Alliance for engineering in medicine & biology*. 1981, p. 404.
- [16] Sangok Seok et al. “Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot”. In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 3307–3312. DOI: 10.1109/ICRA.2013.6631038.
- [17] Kyoung-Tak Kang et al. “Finite Element Analysis of the Biomechanical Effects of 3 Posterolateral Corner Reconstruction Techniques for the Knee Joint”. In: *Arthroscopy: The Journal of Arthroscopic and Related Surgery* 33.8 (2017), pp. 1537–1550. ISSN: 0749-8063. DOI: <https://doi.org/10.1016/j.arthro.2017.02.011>. URL: <https://www.sciencedirect.com/science/article/pii/S0749806317301780>.
- [18] Equine Hoof Explorer Online. *Horse leg anatomy*. URL: <https://hoofexplorer.com/en/>.
- [19] *Wikipedia - Tensegrity*. URL: <https://en.wikipedia.org/wiki/Tensegrity>.
- [20] Marco Cavazzuti et al. “High performance automotive chassis design: a topology optimization based approach”. In: *Structural and Multidisciplinary Optimization* 44.1 (July 2011), pp. 45–56. ISSN: 1615-1488. DOI: 10.1007/s00158-010-0578-7. URL: <https://doi.org/10.1007/s00158-010-0578-7>.
- [21] Open Dynamic Robot Initiative. *SOLO Micro Driver Electronics*. URL: https://github.com/open-dynamic-robot-initiative/open_robot_actuator_hardware/blob/master/electronics/micro_driver_electronics/README.md#micro-driver-electronics.
- [22] Ben G. Katz. *GitHub - BLDC Driver fro MIT Mini Cheetah*. URL: https://github.com/bgkatz/3phase_integrated.
- [23] Ben G. Katz. *HobbyKing Mini Cheetah blog*. URL: <https://build-its-inprogress.blogspot.com/search/label/HobbyKing%20Cheetah>.
- [24] ODrive. *BLDC Controller*. URL: <https://odriverobotics.com/>.

- [25] Albert Wang and Sangbae Kim. “Directional efficiency in geared transmissions: Characterization of backdrivability towards improved proprioceptive control”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 1055–1062. DOI: 10.1109/ICRA.2015.7139307.
- [26] *Ebikes Simulator - Clute H3540*. URL: https://ebikes.ca/tools/simulator.html?batt=B2412SLA&grade=0&axis=rpm&hp=120&batt_b=B4812_SL&hp_b=120&bopen=true.
- [27] Legged-robot. *GitHub - FORE BLDC motor driver hardware files*. URL: https://github.com/Legged-robot/bldc_hardware.
- [28] Open Dynamic Robot Initiative. *GitHub - SOLO Machined Parts*. URL: https://github.com/open-dynamic-robot-initiative/open_robot_actuator_hardware/blob/master/mechanics/actuator_module_v1/details/details_machined_parts.md#details-machined-parts.
- [29] Fang Qi et al. *Motor Handbook*. https://www.infineon.com/dgdl/Infineon-motorcontrol_handbook-AdditionalTechnicalInformation-v01_00-EN.pdf?fileId=5546d4626bb628d7016be6a9aa637e69(visited 2021-10-01). Infineon, 2019.
- [30] *Ben Katz - Motor Control Progress: Working Hardware, and a Field Oriented Control Implementation*. URL: <https://build-its-inprogress.blogspot.com/search/label/HobbyKing%20Cheetah?updated-max=2016-02-28T18:02:00-05:00&max-results=20&start=18&by-date=false>.
- [31] Open Dynamic Robot Initiative. *GitHub - SOLO leg teststand*. URL: https://github.com/open-dynamic-robot-initiative/open_robot_actuator_hardware/blob/master/mechanics/leg_test_stand_v2/README.md#leg-test-stand-v2.
- [32] Carlos Mastalli et al. “Crocodyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2020.
- [33] Justin Carpentier et al. “The Pinocchio C++ library – A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives”. In: *IEEE International Symposium on System Integrations (SII)*. 2019.
- [34] Andrea Del Prete. *Robot Modeling - Optimization-based Robot Control*. URL: https://andreadelprete.github.io/teaching/2019_PhD_obrc/1_modeling.pdf.
- [35] Ben G. Katz. *GitHub - USB to CAN interface*. URL: <https://github.com/bgkatz/USBtoCAN>.

-
- [36] Legged-robot. *GitHub - FORE Legged Robot Project Page*. URL: <https://github.com/Legged-robot>.