Автономная некоммерческая организация высшего образования «Университет Иннополис» (АНО ВО «Университет Иннополис»)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)
по направлению подготовки
09.04.01 – «Информатика и вычислительная техника»

GRADUATION THESIS (MASTER GRADUATE THESIS)
Field of Study
09.04.01 – «Computer Science»

Направленность (профиль) образовательной программы «Робототехника»
Area of Specialization / Academic Program Title: «Robotics»

Тема / Topic	**Динамическая походка четвероногого робота / Dynamic walking of quadruped robot**

Работу выполнил / Thesis is executed by	**Севостьянов Илья Евгеньевич / Sevostianov Ilia Evgenievich**	подпись / signature
Руководитель выпускной квалификационной работы / Graduation Thesis Supervisor	**Гапонов Игорь Юрьевич/ Gaponov Igor Yurievich**	подпись / signature

Иннополис, Innopolis, 2021

Contents

List of Tables

List of Figures

Abstract

This thesis explores the design, assembly and control of dynamic quadrupedal robot using direct and quasi-direct actuators and to make it move: balancing, jumping, crawling and so on.

Firstly, the model of the robot is designed. After that, one robot's leg is printed to provide some experiments with it. An experimental stand with a vertical rail is built for that purpose. The electronics for communication with motors is made and python API is written. After that, the robot with quasi-direct actuators in its joints is assembled and several control techniques are tested.

In result, I achieve next results: the quadruped robot is assembled, the communication electronics for it are built and various control techniques are implemented allowing the robot either to stand on its feet and twist or jump.

Chapter 1

Introduction

1.1 Motivation

Legged robots, in common, are used in a huge sphere of actions, such as: cargo delivery, inspection, terrain mapping, entertainment industry and so on.

As a rule, the motivation for the manufacture of such robots is the release of a person from difficult, repeatable tasks, an increase of the person's safety, and so on. However, wheeled robots have less cross-country ability and maneuverability. Humanoid robots are less sustainable. Quadruped robots, on the contrary, demonstrate promising results for rescue operations and for disaster response [1]. Also, they have been approved as promising candidates for operation over challenging terrain [2], [3], [4], [5].

Robots need to be able to manage contact interactions in unstructured environments.

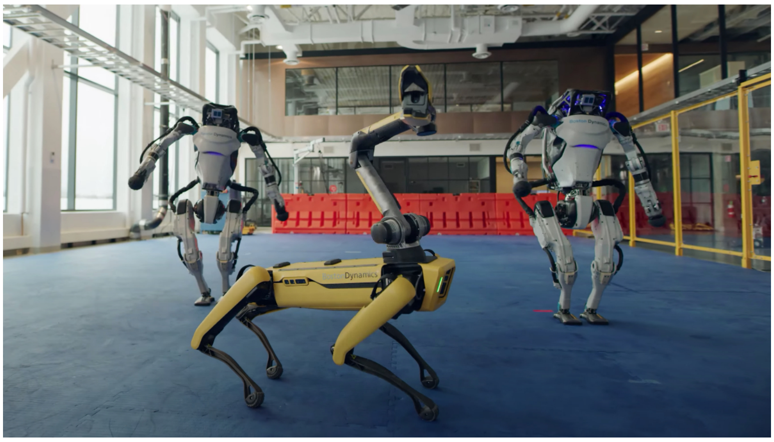


Figure 1.1: Bipedal and quadruped robots [6]

Robots should be able to control both vertical and horizontal force and impedance at feet and easily perform jumping, walking, running and turning [7].

For this purpose, a fit-able body structure should be investigated and designed that could handle all the collisions and interactions with the environment. Appropriate actuators capable of enabling highly-dynamic performance should be used and a control framework should be implemented to provide stable dynamic behaviour.

1.2 Research Hypotheses and Contributions

In thesis project, proprioceptive actuation is used to help controlling the quad-robot, to make it move and allow it to be more easy-to-develop because no exteroceptive sensors are used and the legged robot is localized using joint encoders and inertial measurement unit.

Besides using QDA (quasi-direct actuators), some control techniques are implemented, body of quadruped robot is adjusted, designed and printed and the electronics are made.

So, this work consists of the following parts:

- The quadruped robot is designed taking into account state-of-the-art quadrupeds.
- The electronics are designed to control motors in robot's joints using CAN interface. Also, the python API for communication is written.
- Various control algorithm are tested on experimental stand for jumps with one leg to verify the design of the leg and check the electronics.
- The whole robot is assembled and various control algorithms are implemented to balance on its feet, jump and walk.

The aim of the research is the understanding of the structure, assembly, hardware and control of such types of systems as quadrupeds robots. This research should enhance our knowledge about quadruped robots and allow us better understand which techniques are more appropriate in practice. Also, I should know how to resolve robot's dynamics and simplify it to be able to do locomotion.

Also, next hypotheses were set up:

- The dynamic model of the quadruped (bipedal) robot is quite complicated and should be simplified.
- Using quasi-direct actuators, it's easy to design, assemble and run quadruped robot
- The shift between a model in Gazebo simulation and a real robot model is insignificant.

- CoM position and orientation can be easily controlled by simple PD control with some simplifications.

- The robot's body is made of 3D printed plastic. It's easy to prototype and light-weighted.

Chapter 2

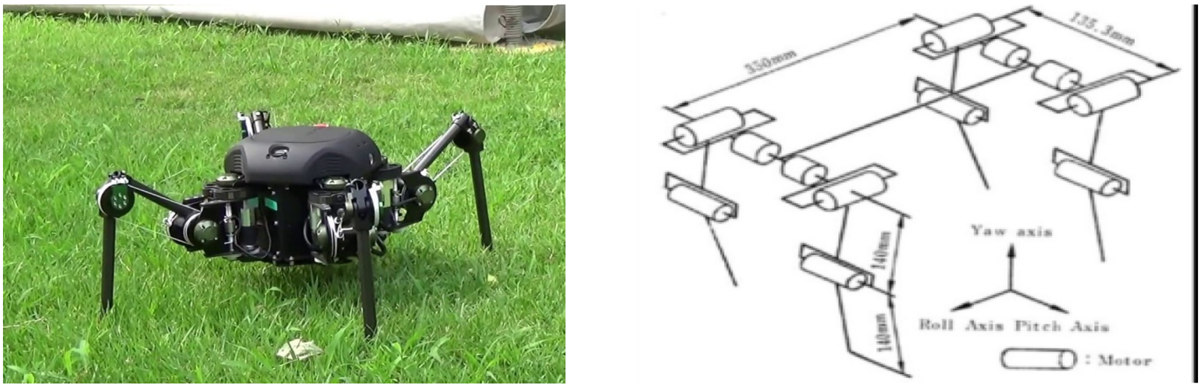
Literature Review

2.1 Mechanical Design Of Quadrupeds

Let's consider quadruped robot's designs existing in the literature. There are several works exploring various designs.

In [8] authors explore the TITAN XIII, the sprawling type quadruped robot . It can effectively work with 1.38 m/s speed and it's powered by battery. It uses the right-angle type wire driven mechanism for legs. The robot has higher stability and and lower center of mass than mammal-type robots.

In next two papers authors explore the dynamic walking robot Collie-I and Collie-II. The first one has 12 degrees-of-freedom (DoF) and a relatively small size with approximate dimensions 15 cm by 13.5 cm by 28 cm [9]. The second one has 5 DoF and actuators on each leg [10]. Both these robots has a weight about 7 kg.

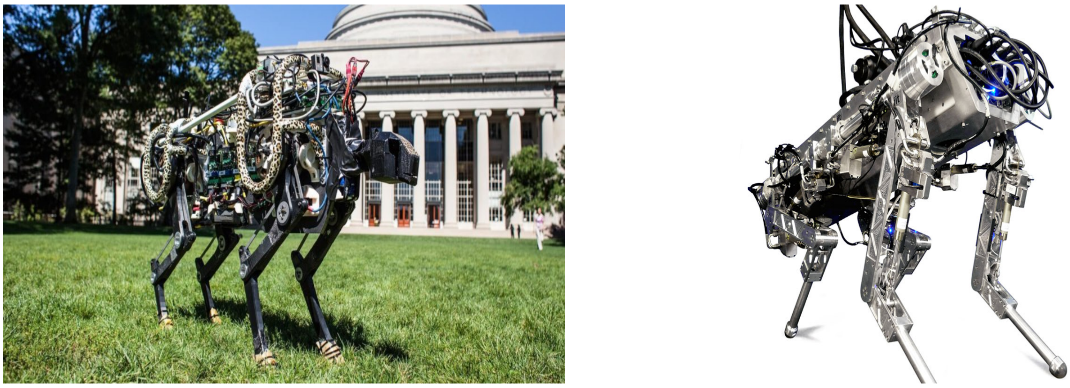


(a) (b)

Figure 2.1: The sprawling-type robots: (a) TITAN XIII [9]; (b) COLLIE-I [10]

Next paper observes the mammal-type hydraulic Quadruped (HyQ). The robot has a total amount of 8 hydraulic actuators and 4 electric ones. It's designed to perform jumping, hopping and running [3].

Let's consider the paper exploring the development of quadruped robot named MIT Cheetah. The total power utilized by the robot is almost 973 Watt, and the cost of power is 0.5 [11]. It has higher center of mass (CoM) than scrawling-type robots.



(a) (b)

Figure 2.2: The mammal-type robots: (a) MIT Cheetah [11]; (b) HyQ [3]

Finally, the last paper observes the quadruped robot named Spot. It can run 5.8 km/h speed and it's total weight is about 30 kg, and it can carry 14 kg load [12]. All the joints are electrically actuated. For knee rotation it uses parallelogram joint.

Figure 2.3: Spot [12]

Here I examined the examples of quadrupeds. As I investigated, there are two types of quadrupeds: mammal-type and scrawling-type. I consider mammal-type quadruped robot.

2.2 Control Techniques For Quadrupeds

Let's consider previous works on the topic of various control techniques for quadrupeds.

In first paper, the authors use the controller that consists of a high-level planner that plans footsteps over the terrain, a low-lever planner for feet trajectories and center of gravity, and low-level controller that tracks these desired trajectories [13]. After this, it uses PD control for joints. Experiments shows that the controller allows the robot to robustly cross a variety of challenging terrains and climbing (Fig. 2.4).

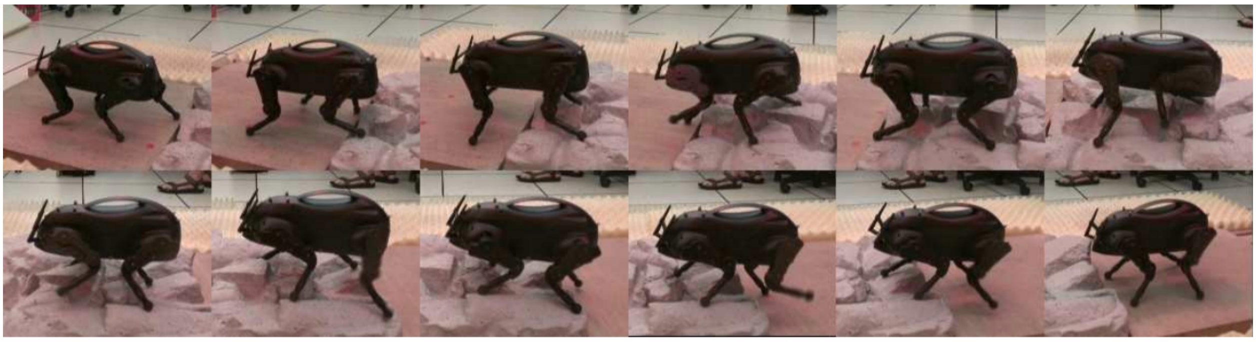


Figure 2.4: Robot crossing terrain [13]

In second paper, the authors use a learned foothold ranking function to minimize the danger of slipping and unplanned collisions. Also, they pre-process the terrain height map by using learnt function [13]. After that, they use online planner per footstep for trajectory generation and solve it via PD control, inverse dynamics force control. The joint PD control works at 400 Hz, Inverse dynamics works at 100 Hz and it doesn't require knowledge of the contact forces at the feet. Using inverse dynamics provide better tracking of trajectories. To negotiate unperceived obstacles, they use force position (P) control.

In the another paper, the authors explore the mini cheetah quadruped. Torque is transmitted to the knee joint through the chain belt transmission. For locomotion, they use convex model predictive control (cMPC) [14]. It uses simplified model of robot dynamics and determines ground reaction forces for the stance feet. The MPC is formulated as quadratic programming and is solved in around 5 ms on the on-board computer. It has demonstrates the locomotion at speed up to 2.45 m/s, the robot also can perform back-flips.

In the last paper, the authors use a neural network architecture for controlling the quadruped. At each frame, the neural network (NN) computes the character state in the current frame regarding the previous frame and operator inputs [15]. It's quite flexible, so it allows doing a wide range of actions such as walk, pace, canter, trot, jumping and turning.

In this part several control methods are described that work for quadruped robots, as inverse dynamics control, PD trajectory control, MPC, mode-adaptive neural network control, e.t.c., they demonstrate good performance for different gaits. However, control of quadrupeds is a complex task that usually requires robot dynamics obtaining, linearization of the whole system and simplification of it.

Chapter 3

Methodology

3.1 Pipeline

To achieve the stated research goals, the pipeline which you can see in the Fig. 3.1 is set up. Going through all the steps I can achieve the desired results.

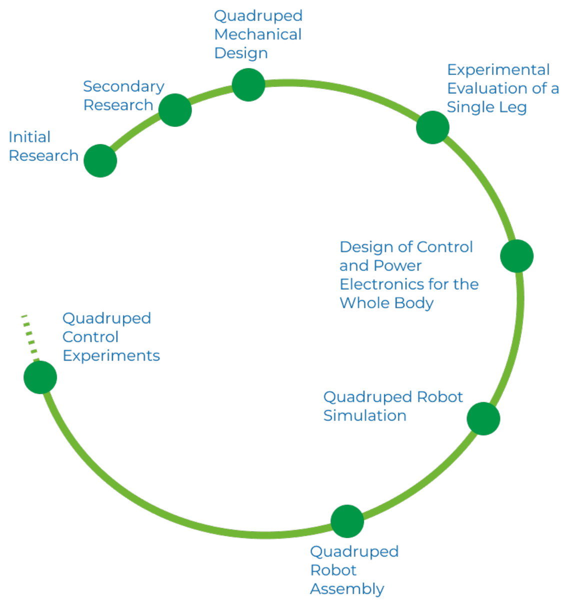


Figure 3.1: The research pipeline.

Secondary research. It's needed to know what approaches have already been implemented, what is the relevant prior work, various mechanical designs of quadrupeds, existent control frameworks for quadrupeds, issues with that, cons and prons of different methods, etc. [16]. Secondary research allows deepen into the problem.

Mechanical design of quadruped robot. Analysis of the quadrupeds' designs which already exist. The design should be 3D printed and it should be easy to assemble. Also, the legs design should imply the opportunity to use force sensors in the feet for future improvements. So, the cable could lie inside the leg body.

Experimental evaluation of a single leg. The main aims of the experiment are: to check the sustainability and strength of the plastic details made of, to understand the motor hardware and how to communicate with it, it's needed to do experiments with simplified system at first. So, I conduct such experiments as jumping on a single leg, PD control, inverse kinematics (IK) and forward kinematics (FK) solution (trajectory tracking) of just one leg fixed on vertical rails. You can see the experimental stand for jumps in Fig.3.2

Figure 3.2: Experimental stand for leg jumping

Design of control and power electronics for the whole body. To control the real quadruped robot, all 4 legs should be controlled in parallel. For that purpose, the electronics (Fig. 3.3) is developed.

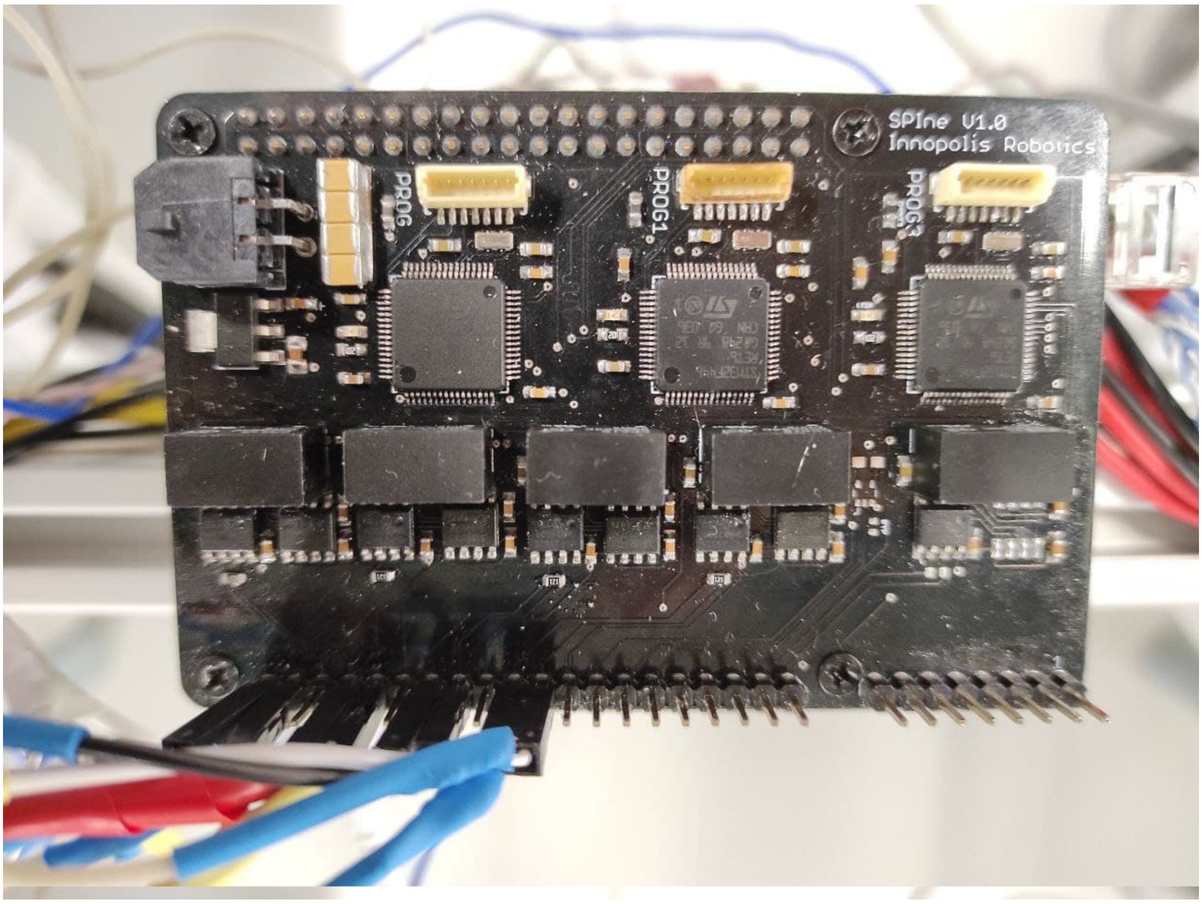


Figure 3.3: SpiNe pcb that communicates with the legs. It's shielded on Raspberry Pi 4B

Simulation of quadruped robot. Not to break everything and easily test control hypothesises, the simulation is vital thing in the research. For that purpose the ROS + Gazebo can be used [17]. Also, ROS allows easily export the robot model from Solidworks to URDF (Fig. 3.4).

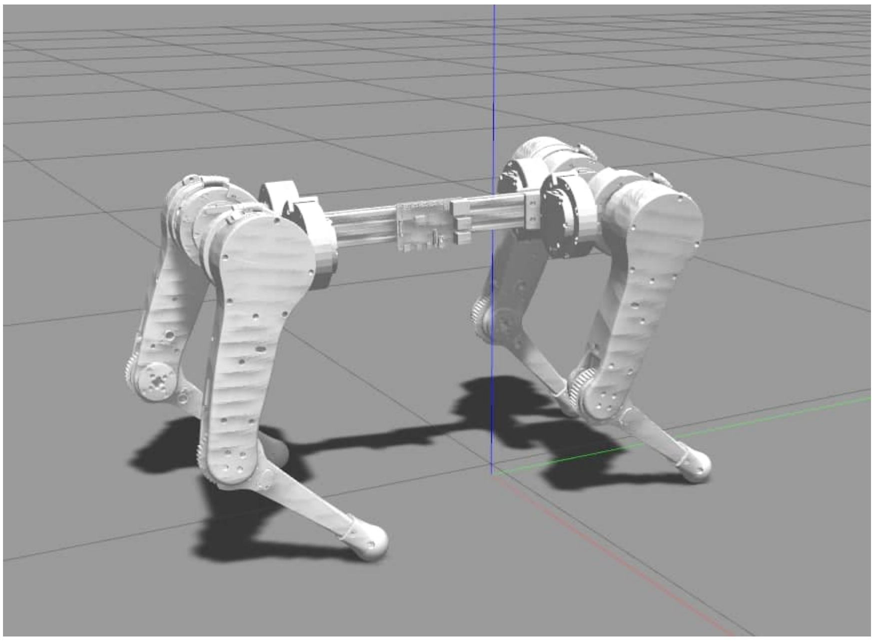


Figure 3.4: Quadruped model in Gazebo

Assembly of quadruped robot. After experiments are done, the full robot body should be assembled including electronics, motors, aluminium bar (the actual body of the robot) and 3D printed parts. The real robot is needed to check the programs that have been written for simulation.

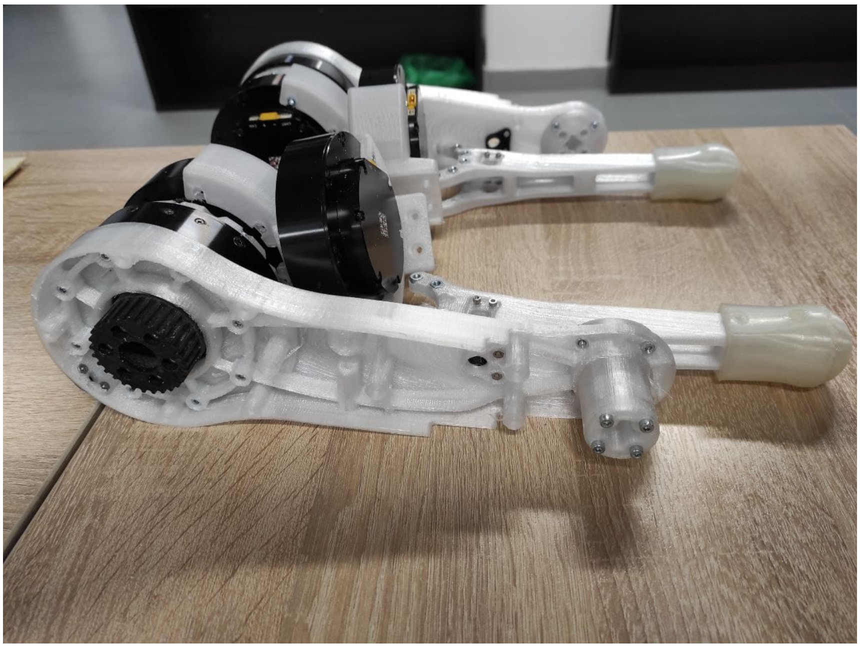


Figure 3.5: Assembly of quadruped robot legs made of plastic

Control experiments. Control experiments are done both in simulation and on the real robot. The following control experiments are interesting for me: CoM pose control when feet are fixed on the ground; force balance control, walking and jumping, etc.

3.2 Design of knee joint

There are at least 2 possible solutions for knee joint that connects thigh and calf of the leg [18]–[21].

The first one is the use of parallelogram linkage as shown in Figure 3.6.

The design implies the two bars of the same length that are connected to each other in such way that the linkage looks like a parallelogram and allows turning the second link. There are several variations of such solution: four bar linkage, six bar linkage, etc. This solution has an advantage that the lift it has is really small and it allows transmitting a big torque. But the main disadvantage is that the angle of rotation is limited because of the construction.

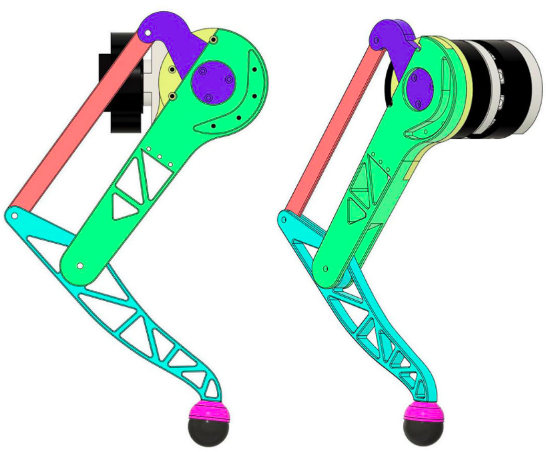


Figure 3.6: Parallelogram linkage [22]

The second possible solution is the use of drive belt connection as shown in Figure 3.7. It has an advantage that it doesn't limit the angle of rotation so much and by using a belt I can increase the transmitted torque in *reduction ratio* times. But it has a bigger lift because of the belt which depends on its stiffness and I need to use 2 gears to bear a belt that need a big accuracy of making so it might be expensive.

Figure 3.7: Belt linkage. Assembling of Mini Cheetah [23]

3.3 Proprioceptive Actuation

Nowadays, the most advanced quadrupeds are made in Boston Dynamics (Spot, BigDog, Cheetah) and Unitree Robotics (A1).

As for Russia, in 2018, the specialists of the Russian Foundation for Advanced Research have created a series of domestic four-legged robots on their own initiative [24].

Departments and companies usually prefer to purchase such robots from foreign development companies for their own tasks and further development.

The quadrupeds are designed to emulate various locomotion capabilities such as walking, running, and jumping. Such behaviors involve repeated high impacts followed by short ground contact times.

Figure 3.8 shows the example of the detailed design of the quasy-direct actuator.

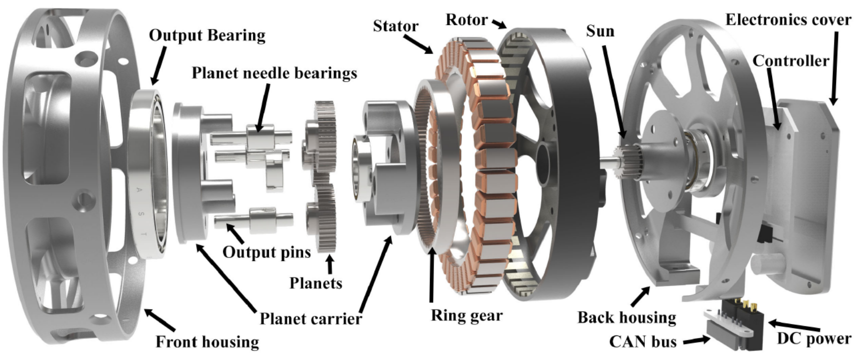


Figure 3.8: Exploded view of the actuator [25]

The motor is specifically designed to maximize the saturation torque for a given mass, where the major design trade-off is between ohmic loss and the saturation torque. The saturation torque density of the custom designed motor could be up to $55\frac{Nm}{kg}$.

Given the large torque capacity of the motor, the gear ratio should be chosen to meet the normal GRF generation requirements for running at a range of needed speeds.

The structure of the leg should also be designed to minimize mass and leg inertia, and thus maximize the impact mitigation factor. Bending stress is minimized in the leg structure by distributing tensile forces to the tendons.

So, there is a new actuator paradigm for high-speed running robots, which shows next:

- A prototype leg model manages impact and open-loop force control for legged machines with use of quasy-direct actuators;
- promising performance.

Geometric scaling analysis indicates that increasing the gap radius benefits torque density, which plays a critical role in system energetics for locomotion.

A successful implementation of the actuator design principles is shown to allow for high-force proprioception to deliver desired forces through contact with only motor current sensing.

These results encourage broader adoption of proprioceptive actuators to manage physical interaction for emerging applications in robotics. And that's why I use it in the thesis project.

3.4 Quadruped Robot Control

3.4.1 Control of Dynamic Quadruped Robots

Having necessary actuators, the control software architecture and components should be presented. Due to the the mechanical design of most quadruped robots there are control strategies enabled for dynamic locomotion to manage physical interaction with the environment. By design, the robot has light limbs with low inertia as compared to the overall body. For this reason, the control model can be simplified to ignore the effects of the legs for planning ground reaction forces (GRF) from the stance feet [25].

The overall architecture of the system is depicted in the block diagram in Fig. 3.9.

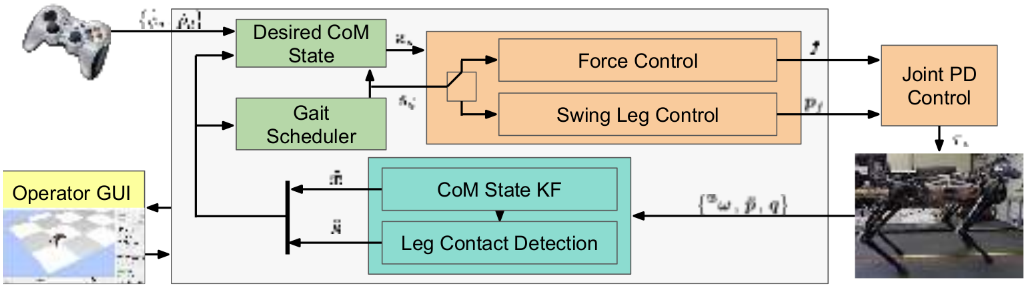


Figure 3.9: The user sends velocity and turning commands as well as general tunable parameters to the main computer. The main Cheetah control computer is composed of three main parts: higher-level planning (green), leg and body control (red), and state estimation (blue). The force and position commands are sent to the micro controllers for each leg that transfer the motor command to the robot [25].

The control model is given by

$$\underbrace{\begin{bmatrix} \boldsymbol{I}_3 & \dots & \boldsymbol{I}_3 \\ [\boldsymbol{p}_1 - \boldsymbol{p}_c]\times & \dots & [\boldsymbol{p}_4 - \boldsymbol{p}_c]\times \end{bmatrix}}_{\boldsymbol{A}} \boldsymbol{F} = \underbrace{\begin{bmatrix} m(\ddot{\boldsymbol{p}}_c + \boldsymbol{g}) \\ \boldsymbol{I}_G \dot{\boldsymbol{w}}_B \end{bmatrix}}_{\boldsymbol{b}}, \tag{3.1}$$

where m and $\boldsymbol{I}_G$ are the robot's total mass and centroidal rotational inertia, $\boldsymbol{g}$ is the gravity vector and $\boldsymbol{p}_i, i \in 1, 2, 3, 4$ are the positions of the feet. The term $[\boldsymbol{p}_i - \boldsymbol{p}_c]\times$ is the skew-symmetric matrix representing the cross product $(\boldsymbol{p}_i - \boldsymbol{p}_c) \times \boldsymbol{F}$.

Based on this, it's possible to derive various types of dynamic controllers:

- Balance controller
- MPC

As I can see from the diagram 3.9, the input is provided by the user inputs, so a higher level path planner could be developed to allow autonomous operation in the world.

Also, it is possible to control feet primitives - $stance, swing$ which can represent 16 possible primitives in total because there are four feet.

So the desired foot position $p_{d,i}$ for foot i could be computed:

$$p_{d,i} = p_{0,i} + k(\dot{p}_c - \dot{p}_{d,c}) \tag{3.2}$$

where $p_{0,i}$ is the default state.

Thesis research is linked to these researches due to the aim of using quasy-direct actuators to dynamically control quadruped robot made by our own.

The main aim of the research is to get knowledge

- how to control quadruped robots;
- understand dynamic behaviour of gates;
- assemble our own version of the quadruped robot implying lots of tests and refactoring;

3.4.2 Forward Kinematics

Forward kinematics allows to determine the position and orientation of end effector by knowing robot's joint variables. Usually, this problem is easily solved by attaching coordinate frames to each link of the robot and express the relationships among them as homogeneous transformations.

Here, I usually make an assumption that each joint has a single DoF (degree-of-freedom), so, the action of each joint can be described by a single real number:

- revolute joint: rotation angle

- prismatic joint: displacement.

With each joint i I associate a joint variable denoted by q_i:

$$q_i = \begin{cases} \theta_i, & \text{if joint } i \text{ is revolute} \\ d_i, & \text{if joint } i \text{ is prismatic} \end{cases} \tag{3.3}$$

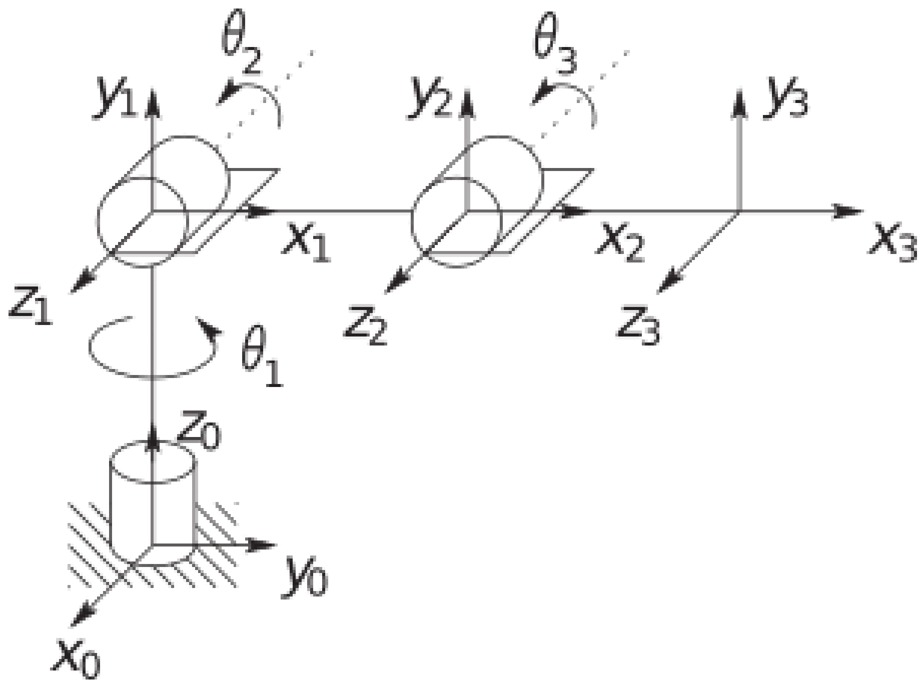


Figure 3.10: Coordinate frames attached to the manipulator [26]

To perform forward kinematics analysis, I attach a coordinate frame rigidly to each link (Fig. 3.10).

Supposing that I know homogeneous transformations between each frames, I can find a transformation matrix T_j^i.

$$T_j^i = \begin{cases} A_{i+1}A_{i+2}\ldots A_{j-1}A_j, & \text{if } i \leqslant j \\ I, & \text{if } i = j \\ (T_i^j)^{-1}, & \text{if } j > i \end{cases} \tag{3.4}$$

where A_i is the homogeneous transformation matrix giving the position and orientation of $o_i x_i y_i z_i$ with respect to (wrt) $o_{i-1} x_{i-1} y_{i-1} z_{i-1}$ [26].

3.4.3 Inverse Kinematics

In general, inverse kinematics problem is about finding the joint variables by knowing the end-effector position.

In more general sense, the problem can be stated as follows. Given a homogeneous transformation

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3) \tag{3.5}$$

find a solution(s) of the equation

$$\mathbf{T}_n^0(q_1, \ldots, q_n) = \mathbf{H} \tag{3.6}$$

where

$$\mathbf{T}_n^0(q_1, \ldots, q_n) = \mathbf{A}_1(q_1)\ldots\mathbf{A}_n(q_n) \tag{3.7}$$

specifies the forward kinematics of an n-degree-of-freedom manipulator.

There are several approaches of inverse kinematics problem solution [27]:

- geometrical approach

- numerical (analytical) approaches

3.4.4 Proportional-derivative control

The PD compensator is one of the most common types of controller used in many robotics systems. The general form of PD controller $\mathbf{U}$ is

$$\mathbf{U} = \mathbf{K}_P\mathbf{E} + \mathbf{K}_D\dot{\mathbf{E}} \tag{3.8}$$

where:

- $\mathbf{K}_P, \mathbf{K}_D$ are proportional and derivative gain matrices,
- $\mathbf{E}$ is the tracking error matrix.

Also, PD controller has to be optimally tuned. It means that I must choose such optimal gains so that it is possible to achieve the desired performance [28]. Also, for quadrupeds it's customary to use PD control with some low-level control, such as MPC, and high-level control like reinforcement learning control [29], etc.

3.4.5 QP-based locomotion control

The general centroidal dynamics of the quadruped robot can be written as:

$$\ddot{p}_{body} = \frac{\sum_{i=1}^{4} f_i}{m} - g \tag{3.9}$$

$$\mathbf{I}\dot{w} = \sum_{i=1}^{4} p_i \times f_i \tag{3.10}$$

where $\ddot{p}_{body}$ is the robot base linear acceleration, f_i is the GRF on each foot, g is gravity vector and m is the robot's mass, $\mathbf{I}$ is the robot's inertia tensor and w is the angular velocity of the robot's body.

I ignore the Coriolis force $w \times (\mathbf{I}w)$ since for bodies with small angular velocities it doesn't contribute a lot to the robot's dynamics [29], [30].

Also, for small values of roll and pitch angles, the angular velocity could be approximated by:

$$w = \mathbf{R}_z(\psi)\dot{\Theta} \tag{3.11}$$

where ψ is yaw angle, $\mathbf{R}_z$ is the rotation matrix about z-axis on yaw angle.

The inertia tensor in the world frame can be written as:

$$\mathbf{I} = \mathbf{R}_{\mathcal{B}}\mathbf{I}\mathbf{R}^T \tag{3.12}$$

And it could be approximated for small roll and pitch angles by:

$$\hat{\mathbf{I}} = \mathbf{R}_z(\psi)_{\mathcal{B}}\mathbf{I}\mathbf{R}_z(\psi)^T \tag{3.13}$$

where ${}_{\mathcal{B}}\mathbf{I}$ is the inertia tensor in body coordinates.

So, the linearized dynamics can be written as:

$$\ddot{q} = \mathbf{M}f - \begin{bmatrix} g \\ 0_3 \end{bmatrix} \tag{3.14}$$

where

$$\mathbf{M} = \begin{bmatrix} 1_3/m & \dots & 1_3/m \\ \mathbf{R}_z^T {}_{\mathcal{B}}\mathbf{I}^{-1}[p_1]_\times & \dots & \mathbf{R}_z^T {}_{\mathcal{B}}\mathbf{I}^{-1}[p_4]_\times \end{bmatrix} \tag{3.15}$$

In further, given a target robot base pose q_d and velocity $\dot{q}_d$, I can use PD control to compute the desired acceleration

$$\ddot{q}_d = k_p(q_d - q) + k_d(\dot{q}_d - \dot{q}) \tag{3.16}$$

And then it's used to construct a QP to find foot forces minimazing the $\ddot{q}_d - \ddot{q}$ error while respecting the friction and the contact configuration constraints

$$\begin{aligned} \min_{f} \quad & ||\ \mathbf{M}f - \tilde{g} - \ddot{q}_d\ ||\ \mathbf{Q}+\ ||\ f\ ||\ \mathbf{R} \\ \text{s.t.} \quad & f_{z,i} \geq f_{z,min}\ \textit{if}\ P_{t,i}\ \textit{is}\ \textbf{Stance} \\ & f_{z,i} = 0\ \textit{if}\ P_{t,i}\ \textit{is}\ \textbf{Swing} \\ & -\mu f_x \leq f_z \leq \mu f_x \\ & -\mu f_y \leq f_z \leq \mu f_y \end{aligned} \tag{3.17}$$

where $\mathbf{Q}, \mathbf{R}$ are diagonal matrices that adjust weights in the cost function, Swing and Stance are leg primitives states.

After that, I convert those foot forces into motor torques:

$$\tau = \mathbf{J}^T f \tag{3.18}$$

where $\mathbf{J}$ is the feet Jacobian matrix w.r.t. motor states.

Chapter 4

Implementation

Here I explain the methodology steps in more detailed way with graphs, equations, e.t.c. This section shows how the methodology is implemented for the thesis purposes.

4.1 Experimental Evaluation of a Single Leg

In this section, I will give a brief description of the experimental stand developed to verify the laws of feedback control, plan trajectories taking into account contact, and plan contact sequences for cyclic motions. Also, it allows us to write a framework for actuators and check the assembly stability. The stand consists of just one leg fixed on the vertical rails. The leg has two links and two joints. As actuators, quasi-direct actuators are used here.

For the experiment carried out in this section, the leg is fixed on a system of rails developed for this purpose, allowing it to move vertically, that is, to make jumps. The stand is shown in Figure 3.2.

Prior to the verification, the main components of the four-legged robot are tested, including the impact resistance tests. The body parts and foot are 3D printed for later debugging and possible improvements if problems are identified.

	tmotor AK80-9	tmotor AK80-6
Voltage, V	24	24
Rated torque, Nm	9	6
Peak torque, Nm	18	12
Rated current, A	12	12
Peak current, A	24	24
K_m, Nm/a	0.091	0.091
K_v, RPM/V	100	100
R, mOhm	170±5	170±5
L, uH	57±10	57±10

Table I: TMOTOR actuators characteristics

Tmotor AK80-9 and Tmotor AK80-6 (quasi-direct actuators) are used in the leg's joints. The table I shows their main characteristics. Rotation from the first link to the second one is transferred via a belt drive with a gear ratio of 1.5 (Fig. 4.1).

Figure 4.1: 3D model of quadruped leg

4.1.1 Control system of motors

The motor control includes two levels: upper and lower. The lower level is implemented by the motors manufacturer and is left unchanged. The upper level depends on the experiment being carried out.

Low level control scheme provided by TMOTORS you can see in figure 4.2.

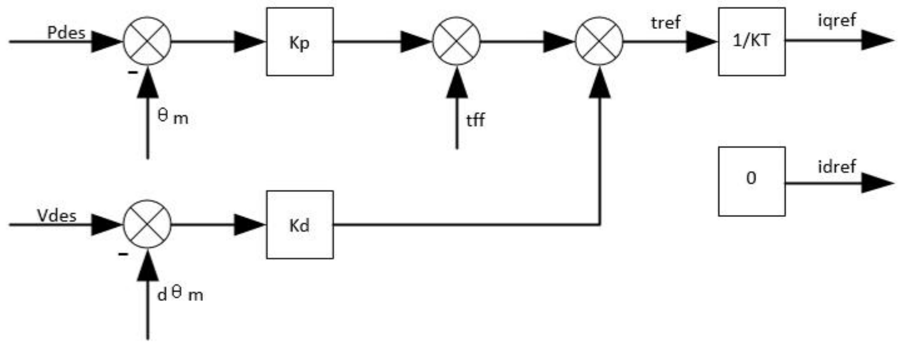


Figure 4.2: Low level control scheme of TMOTOR AK80-(6,9). θ_m - rotation angle, $d\theta_m$ - rotation velocity, $\mathbf{t}_{ref}$ - torque (readable); $\mathbf{i}^q_{ref}$, $\mathbf{i}^d_{ref}$ - currents on q and d-axis (readable)

CAN control commands comprise:

- $\mathbf{P}_{des}$ - desired rotation angle;
- $\mathbf{V}_{des}$ - desired rotation velocity;
- $\mathbf{t}_{ff}$ - feed-forward torque;
- $\mathbf{K}_p$, $\mathbf{K}_d$ - proportional and derivative coefficients.

Considering the scheme above, I note that it is possible to use position control, speed control, torque control or even their super-positions.

Proportional-differential control is implemented to perform preliminary tests and investigate the operatability of contact sequence planning systems. The desired torque is determined on the upper level computer of the ACS (automatic control system) and is sent to the lower level via the CAN protocol and the written python interface, and at the same time the angles of rotation and rotation speeds of the motors are monitored.

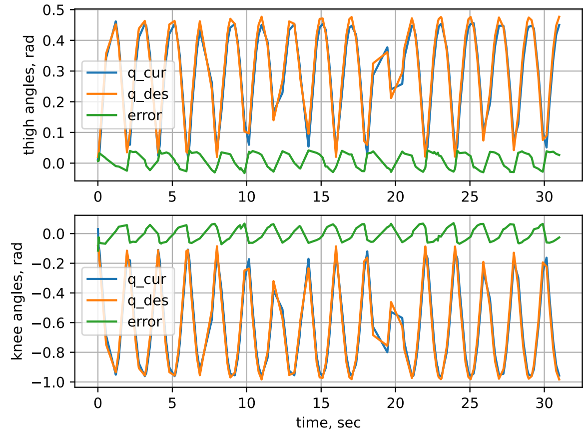
4.1.2 Experiments

Of all the experiments that have been done, I give a brief description of two:

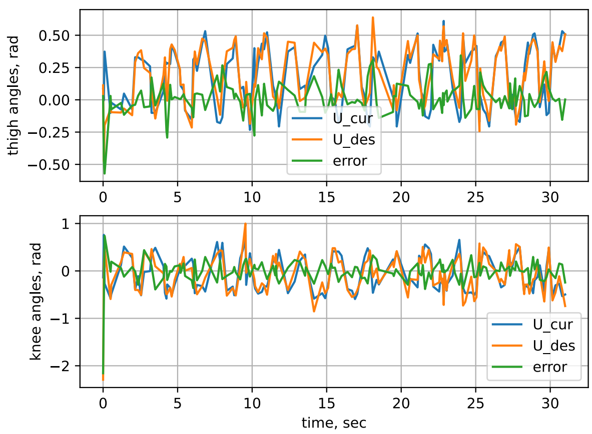
- execution of movements with a planned but varying frequency of gain and loss of contact as well as changing paths of links between acquisitions of contact;
- testing the stand for the execution of movement when applying moments that are on the border of admissible controlled influences.

Let's consider the first one.

Also note that $\mathbf{q}_{des}$ is calculated based on the given trajectory of the foot $\mathbf{x}_0 + A\sin(wt)$. Knowing the position of the trajectory point at time t, I solve the inverse kinematics problem (4.1.4) to find the desired angles of rotation.

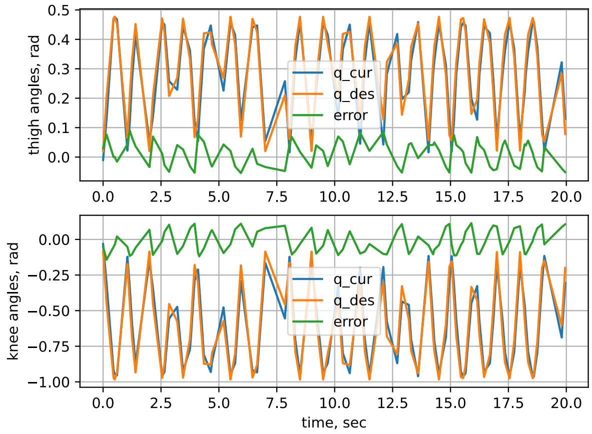


(a) Rotation angles

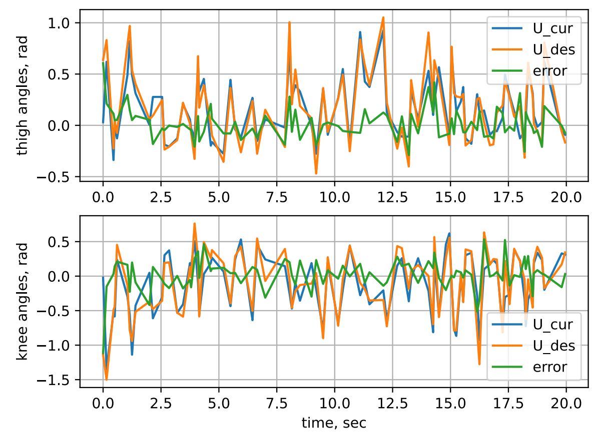


(b) Torques

Figure 4.3: Results of first experiment. A = 0.075 m, w = 3.14 sec^{-1}

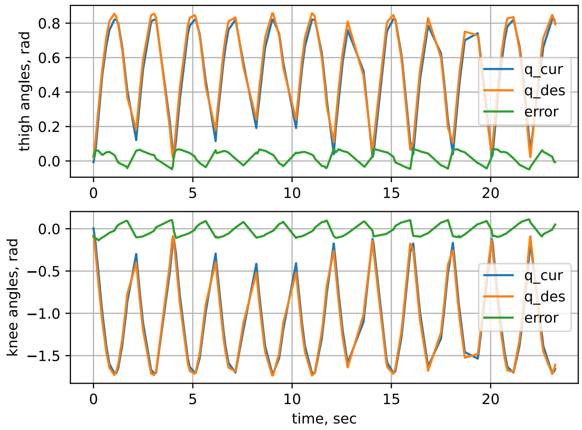


(a) Rotation angles

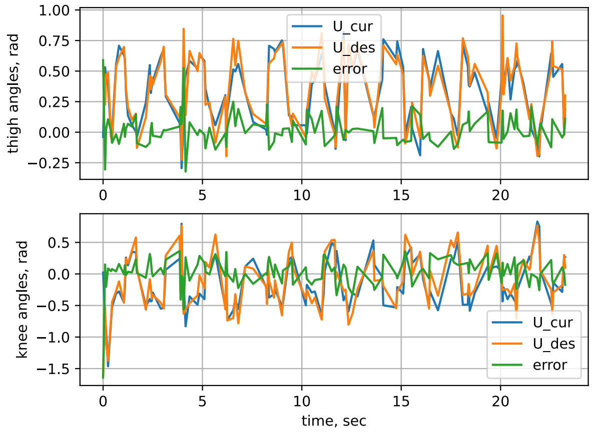


(b) Torques

Figure 4.4: Results of second experiment. A = 0.025 m, w = 6.28 sec^{-1}



(a) Rotation angles



(b) Torques

Figure 4.5: Results of third experiment. A = 0.075 m, w = 3.14 sec^{-1}

As you can see from the graphs above, at the same frequency of the reference trajectory with increasing amplitude, the tracking error also increases. Also, the graphs are made with an interval between saving info equal to 0.2 sec not to interrupt the main control, GUI,etc. Based on the graphs, for a reliable and accurate solution of the tracking problem - as for the case when the tracking task is formed for the foot trajectory, as well as for controlling the gait of the robot itself - a simple proportional-differential controller is enough (RMSE is equal to 0.02-0.05 rad) for our goals, but to make it more precise, it is necessary to calculate the dynamics of the robot, legs, identify the system and, based on the obtained calculations, apply more reliable and accurate control methods, such as optimization approaches for balancing the robot and its jump. The video of experiments you can find here (link) and here (link).

4.1.3 Forward kinematics derivation

Robot has 4 legs. But only 2 of them are individual: left forward one and left back one. Others are just copies of these two. So, you can see the kinematic scheme of the robot in the fig 4.6.

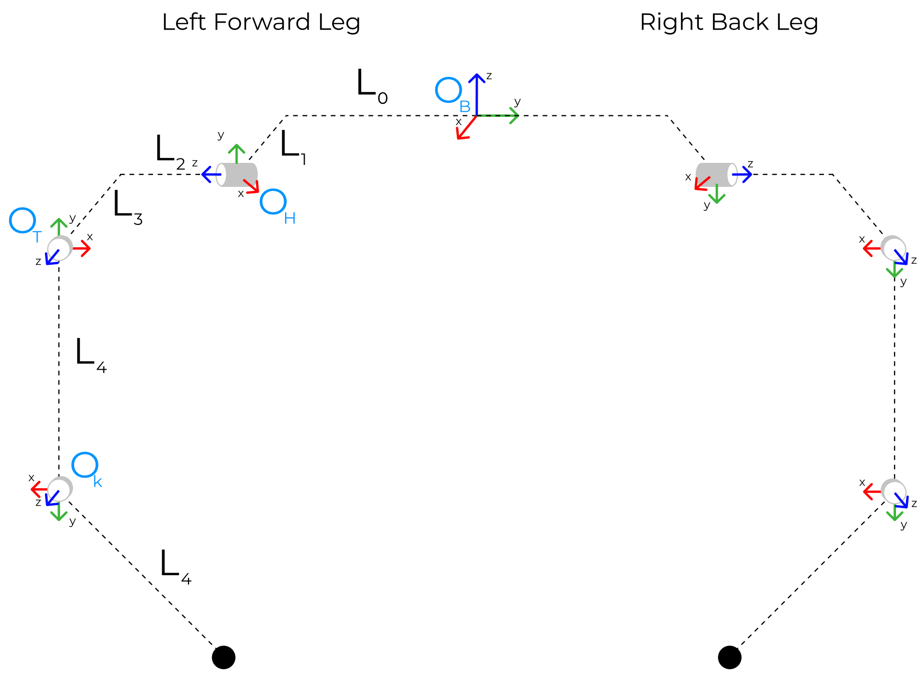


Figure 4.6: Kinematics scheme of the quadruped (only two legs are shown).

Considering that I have next rotation angles in each leg:

- hip - α
- thigh - β
- knee - γ

I can derive FK for each leg easily:

$$\boldsymbol{T}_{LF} = \boldsymbol{T}_{base} \cdot \boldsymbol{R}_z(\alpha) \cdot \boldsymbol{T}_z(L_2) \cdot \boldsymbol{T}_x(L_3) \cdot \boldsymbol{R}_y(\frac{\pi}{2}) \cdot \boldsymbol{R}_z(\beta) \cdot \boldsymbol{T}_y(-L_4) \cdot \boldsymbol{R}_z(\pi) \cdot \boldsymbol{R}_z(\gamma) \cdot \boldsymbol{T}_y(L_5) \quad (4.1)$$

for the left forward leg. And for left back leg also:

$$\boldsymbol{T}_{LB} = \boldsymbol{T}_{base} \cdot \boldsymbol{R}_z(\alpha) \cdot \boldsymbol{T}_z(L_2) \cdot \boldsymbol{T}_x(L_3) \cdot \boldsymbol{R}_z(\beta) \cdot \boldsymbol{T}_y(L_4) \cdot \boldsymbol{R}_z(\gamma) \cdot \boldsymbol{T}_y(L_5) \quad (4.2)$$

where $\boldsymbol{T}_{base}$ is the transformation matrix from CoM of the robot to the i-th leg.

4.1.4 Inverse kinematics derivation

The IK for each leg is the same and is geometrically derived.

The structure of the legs correspond to the so-called elbow manipulator (fig. 4.7).

Figure 4.7: Elbow manipulator with shoulder offset(leg) [26].

For the left arm configuration, as wee see from Figure 4.8,

$$\theta_1 = \phi - \alpha \tag{4.3}$$

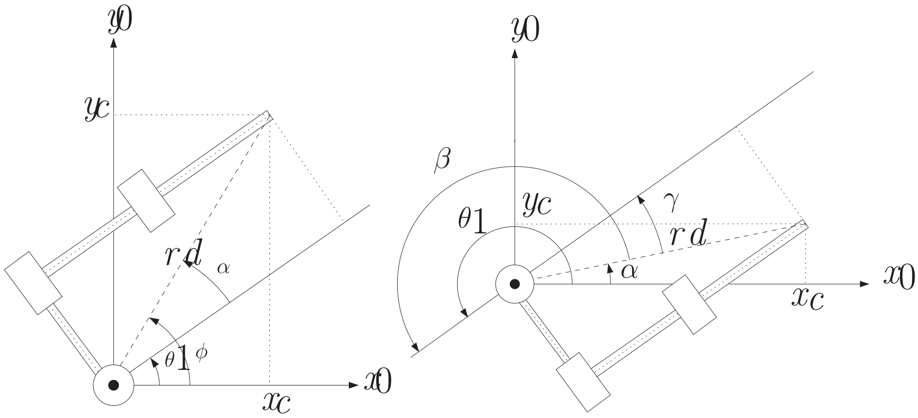


Figure 4.8: Left arm (left) and right arm (right) configurations for a leg [26]

in which

$$\phi = Atan2(x_c, y_c) \tag{4.4}$$

$$\alpha = Atan2(\sqrt{r^2 - d^2}, d) = Atan2(\sqrt{x_c^2 + y_c^2 - d^2}, d) \tag{4.5}$$

The second solution, for the right arm configuration is given by

$$\theta_1 = Atan2(x_c, y_c) + Atan2(-\sqrt{r^2 - d^2}, -d) \tag{4.6}$$

Also, it should be noted that

$$\begin{aligned} \theta_1 &= \alpha + \beta \\ \alpha &= Atan2(x_c, y_c) \\ \beta &= \gamma + \pi \\ \gamma &= Atan2(\sqrt{r^2 - d^2}, d) \end{aligned} \tag{4.7}$$

which imply that

$$\beta = Atan2(-\sqrt{r^2 - d^2}, -d) \tag{4.8}$$

To find the angles θ_1, θ_2 for the leg let's consider the plane formed by the second and third links as shown in Figure 4.9.

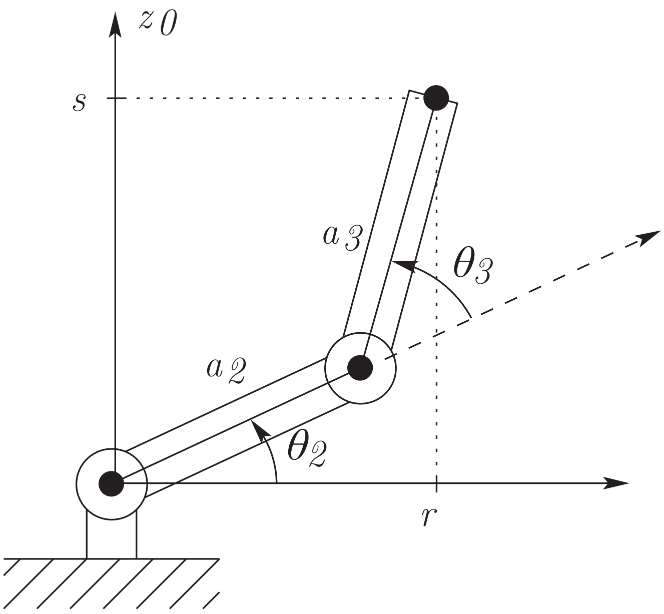


Figure 4.9: Left arm (left) and right arm (right) configurations for a leg [26]

So, I can use the cosines' law to obtain

$$\begin{aligned}\cos\theta_3 &= \frac{r^2 + s^2 - a_2^2 - a_3^2}{2a_2a_3} \\ &= \frac{x_c^2 + y_c^2 - d^2 + (z_c - d_1)^2 - a_2^2 - a_3^2}{2a_2a_3} := D\end{aligned} \tag{4.9}$$

since $r^2 = x_c^2 + y_c^2 - d^2$ and $s = z_c - d_1$. Hence, θ_3 is given by

$$\theta_3 = Atan2(D, \pm\sqrt{1 - D^2}) \tag{4.10}$$

The two solutions for θ_3 correspond to the elbow down and elbow up position, respectively. Similarly, θ_2 is given as

$$\begin{aligned}\theta_2 &= Atan(r, s) - Atan2(a_2 + a_3c_3, a_3s_3) \\ &= Atan2(\sqrt{x_c^2 + y_c^2 - d^2}, z_c - d_1) - Atan2(a_2 + a_3c_3, a_3s_3)\end{aligned} \tag{4.11}$$

4.2 Assembly of quadruped robot

The main parts of the robot such as leg bodies, knees and gears were 3D printed. The printed parts is shown in Figure 4.10.

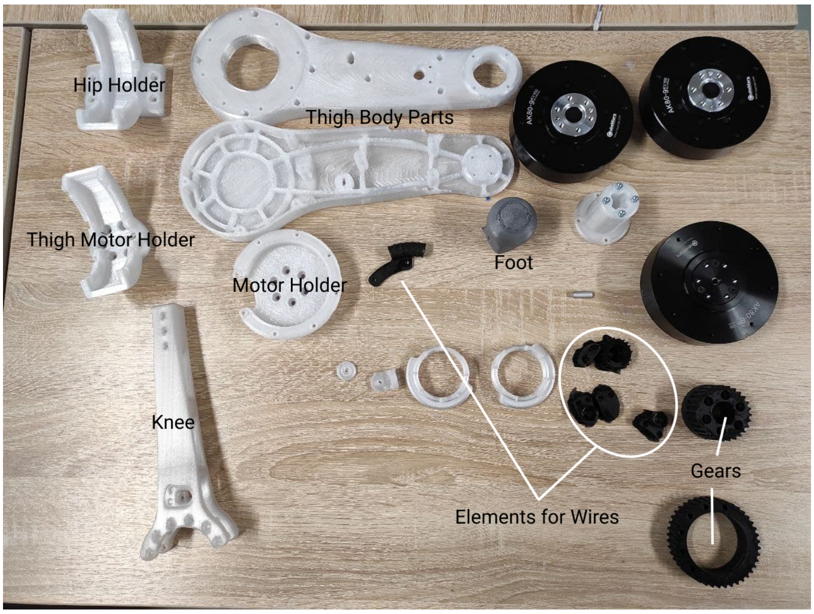


Figure 4.10: 3D printed parts of robot's leg

In Figure 4.11 is shown the leg that's almost assembled. In order to link the thigh and knee the belt is used which gives the reduction ratio equal to 1.5. Also, it should be noticed that the belt is elastic. To keep it strained the tensioner is used which is placed in the middle of the thigh body.

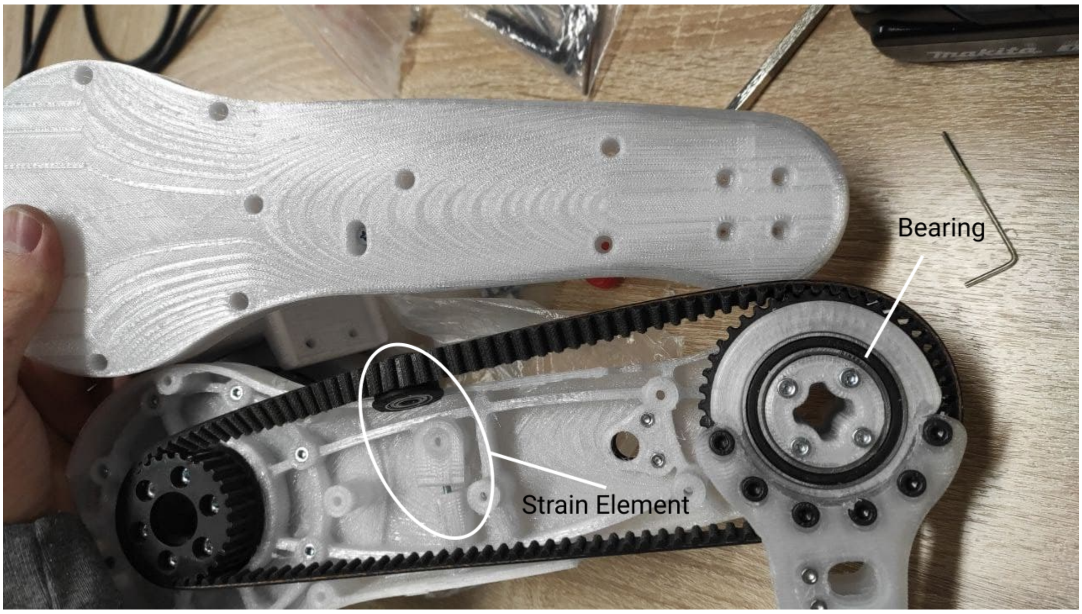


Figure 4.11: The linkage between thigh and knee of robot's leg

The assembled quadruped legs are shown in Figure 4.12.

Figure 4.12: Assembled quadruped legs

4.3 Electronics and system architecture

All the control code is run on the raspberry Pi 4B which is connected to the custom pcb which is capable for legs control (Figure 4.13).

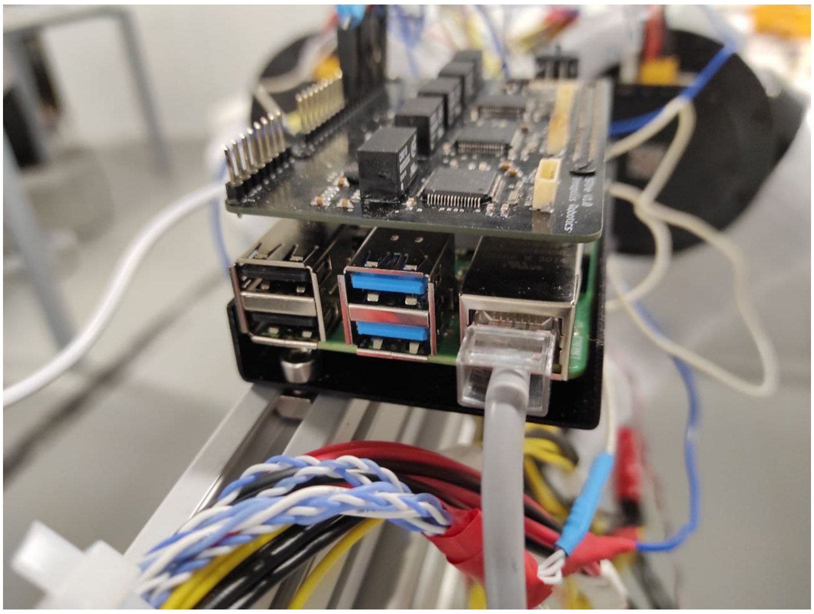


Figure 4.13: Raspberry Pi 4B (down) connected to the custom controller (up).

The Raspberry Pi 4B is connected to the controller via SPi. The controller consists of 3 CAN buses which are controlled by three STM32F446 microcontrollers. This scheme allows controlling 4 legs (12 actuators) at the same time and even connect additional sensors. The controller pcb is based on the one that is built in [31] with some modifications. Mainly, the PCB is needed because Raspberry Pi doesn't originally support CAN interface.

4.4 Simulation of quadruped robot

Also, it should be noted that the simulation of the robot in Gazebo is done. The simulation is mainly needed to test some hypothesises and after that transfer it to the real robot.

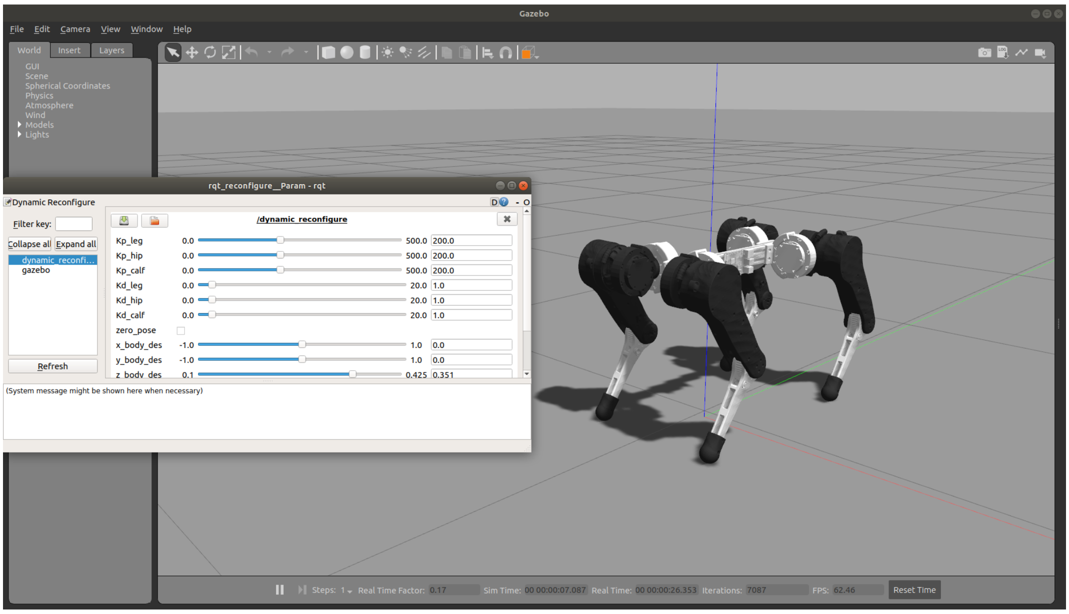


Figure 4.14: Quadruped Robot in Gazebo Simulator

For example, here (link) you can see the video of torso rotation control that we've implemented both for the real robot and simulation.

The corresponding graphs you can see in Figures 4.15 and 4.16

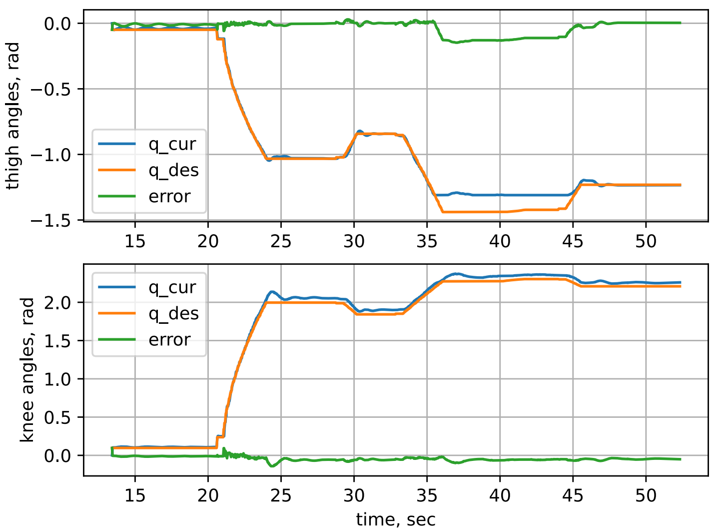


Figure 4.15: Quadruped Robot in Gazebo Simulator. The left back leg graphs

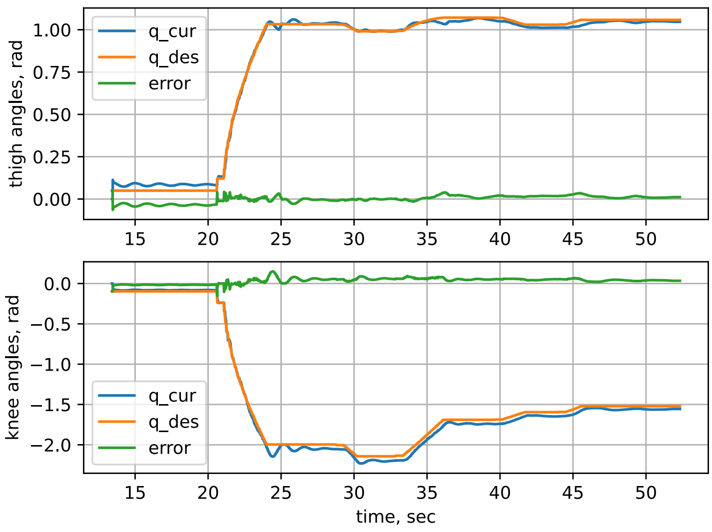


Figure 4.16: Quadruped Robot in Gazebo Simulator. The right forward leg graphs

The RMSE of joint angles is in range 0.015-0.045 rads.

4.5 Control Experiments

4.5.1 Orientation and Position Control of Robot CoM Using PD Control

Here I have next goals:

- P_{des}^{CoM} - desired position (x, y, z) of robot's center of mass with respect to the world
- R_{des}^{CoM} - desired rotation (roll, pitch, yaw) of robot's CoM wrt the world.

To achieve the goals, I use PD effort control for each i-th leg:

$$\boldsymbol{U^i} = \boldsymbol{K}_P(\boldsymbol{P}^i_{des} - \boldsymbol{P}^i_{cur}) + \boldsymbol{K}_D(\boldsymbol{V}^i_{des} - \boldsymbol{V}^i_{cur}) \tag{4.12}$$

where:

- $\boldsymbol{U}^i$ - effort vector of size three by one for i-th leg.
- $\boldsymbol{K}_P, \boldsymbol{K}_D$ - proportional and derivative gain matrices
- $\boldsymbol{P}^i_{des}, \boldsymbol{P}^i_{cur}$ is desired and current position vectors of size three by one for i-th leg, correspondingly. It's the foot position wrt the robot CoM.
- $\boldsymbol{V}^i_{des}, \boldsymbol{V}^i_{cur}$ is desired and current velocity vectors of size three by one for i-th leg, correspondingly. It's the foot velocity wrt the robot CoM.

So, to use this PD controller, I firstly calculate the desired positions of robot's feet regarding desired P^{CoM}_{des} and R^{CoM}_{des}.

Also, I assume that I know the positions $\boldsymbol{P}^i_0$ of feet wrt the world.

In order to find $\boldsymbol{P}^i$, I use next equation:

$$\boldsymbol{P}_i = (\boldsymbol{T}^0_B)^{-1} \cdot \boldsymbol{P}^i_0 \tag{4.13}$$

After all, I put all calculated parameters into PD controller, obtain control vectors $\boldsymbol{U}$ for each leg and send them to the motors via CAN bus.

The result of the implementation you can see on the video (link).

The code of the thesis project is presented in GitHub repository.

Chapter 5

Conclusion

As a result of this work, the following actions were performed.

The first step was the design of the quadruped robot with regarding other existent state-of-the-art quadrupeds. Several types of bodies were analysed and, finally, the body was designed and assembled with 3D printed parts and aluminium profile as a body.

The second step was the design of the electronics to allow controlling the motors. For main board, I chose Raspberry Pi 4B. But it originally didn't have in-build CAN interface. So, I designed a special SPIne PCB that allows us to control the motors. Also, the PCB allows connecting additional sensors which should improve the results and expand the robot's capabilities in further.

The third step was the experiments to do the strength analysis of the assembled robot's legs. The IK and FK problems were solved to execute tracking and regulation tasks both in Cartesian and joint spaces. The jumps with varying amplitude were performed. The conclusion was that such design was suitable for the project tasks and motors could handle both tracking and regulation tasks.

The fourth step was to write several control techniques for torso rotation, jumping, push-ups, etc. The code was written for robot's torso rotation with some simplifications. Also, to increase the frequency of control, the code was split into 3 subprocesses: GUI for control, control, data logging. The next steps are the digging into the quadruped gaits and implementation of various locomotion techniques (QP-based, NN-based, etc.).

Regarding initial hypotheses, it should be said that the dynamic model of quadruped is truly complex and that's why it should be linearized. If not, it can be resolved but it will be a time-consumption task.

Quasy-direct actuators are quite fitable for the quadruped development and meet all the requirements. Besides that, the motors which were used in the project, had a problem with zero angle set. The set zero angle was reset after turning off the motors. The next step is the resolving of this problem.

The shift between the real robot and the one in simulation could be high because of the wrong parameters identification, simulation ODE, solvers parameters, etc. The next steps here are the identification of the robot's parameters and reducing the shift between the real robot and the one in simulation.

Bibliography cited

[1] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 2464–2470.

[2] M. Bajracharya, J. Ma, M. Malchano, A. Perkins, A. A. Rizzi, and L. Matthies, "High fidelity day/night stereo mapping with vegetation and negative obstacle detection for vision-in-the-loop walking," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 3663–3670. DOI: 10.1109/IROS.2013.6696879.

[3] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of hyq–a hydraulically and electrically actuated quadruped robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011.

[4] M. Raibert, "Bigdog, the rough-terrain quadruped robot," vol. 17, Jul. 2008, pp. 10 822–10 825, ISBN: 9783902661005. DOI: 10.3182/20080706-5-KR-1001.01833.

[5] H. Ko, H. Yi, and H. Jeong, "Wall and ceiling climbing quadruped robot with superior water repellency manufactured using 3d printing (uniclimb)," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 4, pp. 273–280, Jul. 2017. DOI: 10.1007/s40684-017-0033-y.

[6] B. Dynamics, *Do You Love Me?* https://www.youtube.com/watch?v=fn3KWM1kuAw, [Online; accessed 10-february-2021], 2020.

[7] W. Bosworth, S. Kim, and N. Hogan, "The mit super mini cheetah: A small, low-cost quadrupedal robot for dynamic locomotion," in *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, 2015, pp. 1–8.

[8] S. Kitano, S. Hirose, A. Horigome, and G. Endo, "Titan-xiii: Sprawling-type quadruped robot with ability of fast and energy-efficient walking," *Robomech Journal*, vol. 3, no. 1, pp. 1–16, 2016.

[9] H. Miura, I. Shimoyama, M. Mitsuishi, and H. Kimura, "Dynamical walk of quadruped robot (collie-1)," in *Int. Symp. Robotics Research*, Cambridge, MA: MIT Press, 1985, pp. 317–324.

[10] H. Kimura, I. Shimoyama, and H. Miura, "Dynamics in the dynamic walk of a quadruped robot," *Advanced Robotics*, vol. 4, no. 3, pp. 283–301, 1989.

[11] D. J. Hyun, S. Seok, J. Lee, and S. Kim, "High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the mit cheetah," *The International Journal of Robotics Research*, vol. 33, no. 11, pp. 1417–1445, 2014.

[12] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick, *et al.*, "Autonomous spot:

Long-range autonomous exploration of extreme environments with legged locomotion," *arXiv preprint arXiv:2010.09259*, 2020.

[13] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.

[14] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 6295–6301.

[15] H. Zhang, S. Starke, T. Komura, and J. Saito, "Mode-adaptive neural networks for quadruped motion control," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–11, 2018.

[16] C. Hewson and D. W. Stewart, "Internet research methods," *Wiley StatsRef: Statistics reference online*, pp. 1–6, 2014.

[17] M. Sokolov, R. Lavrenov, A. Gabdullin, I. Afanasyev, and E. Magid, "3d modelling and simulation of a crawler robot in ros/gazebo," in *Proceedings of the 4th International Conference on Control, Mechatronics and Automation*, 2016, pp. 61–65.

[18] J. G. Nichol and K. J. Waldron, "Biomimetic leg design for untethered quadruped gallop," in *Proceedings of the 5th International Conference on Climbing and Walking Robots*, 2002, pp. 49–54.

[19] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "Bigdog, the rough-terrain quadruped robot," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 10 822–10 825, 2008.

[20] J. T. Kim, J. San Cho, B.-Y. Park, S. Park, and Y. Lee, “Experimental investigation on the design of leg for a hydraulic actuated quadruped robot,” in *IEEE ISR 2013*, IEEE, 2013, pp. 1–5.

[21] S. Seok, A. Wang, M. Y. Chuah, D. J. Hyun, J. Lee, D. M. Otten, J. H. Lang, and S. Kim, “Design principles for energy-efficient legged locomotion and implementation on the mit cheetah robot,” *Ieee/asme transactions on mechatronics*, vol. 20, no. 3, pp. 1117–1129, 2014.

[22] J. Kim, T. Kang, D. Song, and S.-J. Yi, “Design and control of a open-source, low cost, 3d printed dynamic quadruped robot,” *Applied Sciences*, vol. 11, no. 9, p. 3762, 2021.

[23] B. Katz, *BuildIts In Progress:2019*, https://build-its-inprogress.blogspot.com/2019/, [Online; accessed 15-february-2021], 2020.

[24] BBC, *Russia decided to create a fighting robot dog*, https://www.bbc.com/russian/news-37284810, [Online; accessed 21-march-2021], 2016.

[25] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, “Mit cheetah 3: Design and control of a robust, dynamic quadruped robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2245–2252.

[26] M. W. Spong, S. Hutchinson, M. Vidyasagar, *et al.*, *Robot modeling and control*. wiley New York, 2006, vol. 3.

[27] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[28] H. Wu, W. Su, and Z. Liu, “Pid controllers: Design and tuning methods,” in *2014 9th IEEE Conference on Industrial Electronics and Applications*, IEEE, 2014, pp. 808–813.

[29] X. Da, Z. Xie, D. Hoeller, B. Boots, A. Anandkumar, Y. Zhu, B. Babich, and A. Garg, "Learning a contact-adaptive controller for robust, efficient legged locomotion," *arXiv preprint arXiv:2009.10019*, 2020.

[30] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1–9.

[31] B. G. Katz, "A low cost modular actuator for dynamic robots," Ph.D. dissertation, Massachusetts Institute of Technology, 2018.