



DEPARTMENT OF ENGINEERING CYBERNETICS

TTK4550 - SPECIALIZATION PROJECT

---

# Design and Control of a Spring-actuated Jumping Quadruped in Earth Gravity

---

*Author:*

Johannes Ihle  
Daniel Rosmæl Skauge

*Supervisor:*

Prof. Dr. Kostas Alexis

*Co-supervisor:*

Jørgen Anker Olsen

Date

---

## Abstract

This project report presents our specialization project, which is the design and control of a quadruped, spring-actuated, etc.

Here I am trying to cite [13].

---

# Table of Contents

<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Scope . . . . .	2
1.3 Related Work . . . . .	2
<b>2 Theory</b>	<b>3</b>
2.1 Actuator Modeling . . . . .	3
2.1.1 DC Motor Model . . . . .	3
2.1.2 BLDC Motor Model . . . . .	3
2.1.3 Gear Transmission Friction Model . . . . .	3
2.2 Spring Modeling . . . . .	4
2.3 Spring-Damper Systems . . . . .	4
2.4 Kinematics, Jacobians, and Virtual Work . . . . .	4
2.4.1 Robot Kinematics . . . . .	4
2.4.2 Jacobian Matrix . . . . .	5
2.4.3 Force/Torque Mapping . . . . .	5
2.5 Numerical Solvers . . . . .	6
<b>3 Modeling and Simulation</b>	<b>6</b>
3.1 Simscape . . . . .	6
3.2 Rigid Body Components of the Robot Model . . . . .	6
3.3 Rigid Body Masses and Inertias . . . . .	8
3.4 Elastic Components: Springs . . . . .	10
3.5 Motor Modeling . . . . .	10
3.6 Solver selection . . . . .	11
<b>4 Robot Design</b>	<b>11</b>
4.1 The Single Vertical Manipulator Jump Model . . . . .	11
4.2 Actuation Method Selection . . . . .	12
4.3 . . . . .	13
<b>5 Robot Hardware</b>	<b>13</b>

---

<b>6</b>	<b>Robot Control</b>	<b>14</b>
6.1	RL Problem Description? . . . . .	14
<b>7</b>	<b>Results</b>	<b>14</b>
<b>8</b>	<b>Discussion</b>	<b>14</b>
<b>9</b>	<b>Conclusion</b>	<b>14</b>
	<b>Bibliography</b>	<b>15</b>
	<b>Appendix</b>	<b>16</b>
A	Hello World Example . . . . .	16
B	Flow Chart Example . . . . .	16
C	Sub-figures Example . . . . .	17

## List of Figures

1	Torque-speed characteristics of a BLDC motor. TODO: Replace with an image that isn't stolen, and that we are allowed to use. . . . .	3
2	Illustration of a 3 link robotic link arm in $\mathbb{R}^2$ with $n$ links. . . . .	5
3	A typical Simscape block diagram. . . . .	7
4	A visualization of the model in figure 3. . . . .	7
5	Naming conventions for the parts of the robot, as well as forwards direction definition. TODO: name spines . . . . .	8
6	Angle conventions for the robot body. . . . .	9
7	Extension spring implementation in Simscape. As can be seen, the spring is connected between the appropriate points (output frames) of the thigh and shank spine, and in parallel to the knee motor model, which contains the knee joint. . . . .	10
8	Torque-speed characteristics of the motor. . . . .	11
9	The manipulator corresponding to a vertical one leg jump. . . . .	12
10	Vertical Paw velocity as a function of knee angle. . . . .	13
11	Knee speed until takeoff with A80BHP-H motor. . . . .	14
12	Streamline results . . . . .	17

## List of Tables

2	Mass and inertia properties of the rigid bodies in the robot model. A list of the Eurepus robot's electronics can be found in [1]. . . . .	9
3	Masses and dimensions used in the main body mass calculation. . . . .	10

---

## Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
CPU	Central Processing Unit
DRL	Deep Reinforcement Learning
EKF	Extended Kalman Filter
ESKF	Error State Kalman Filter
GNC	Guidance, Navigation, and Control
INS	Inertial Navigation System
ML	Machine Learning
MOOS	Mission Oriented Operating Suite
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
USV	Unmanned Surface Vehicle

---

## 1 Introduction

### 1.1 Motivation

TODO: Shorten the below drastically, but mention the loss of traction in low g for wheeled robots, mentioned in the SpaceHopper paper.

The exploration of extraterrestrial environments represents one of the most demanding frontiers of robotic systems, requiring exceptional autonomy, resilience, and adaptability to navigate complex and unpredictable terrain. On Mars, wheeled rovers have proven their utility, with six successful deployments to date [18], robots like Axel [8] and Reachbot [9] have also been designed, tailored towards specific tasks. One such task that has received much attention in recent years, is the exploration of potential Martian and Lunar lava tubes [2]. These tubes are hollow caverns hypothesized to exist beneath the surface of Mars and the Moon, formed by ancient lava flows. They are of particular interest to astrobiologists and planetary scientists, as they could provide shelter from cosmic radiation and micrometeorites, as well as stable temperatures and access to subsurface water ice [2].

The exploration of such lava tubes present a unique challenge to robotic systems, as they are believed to be characterized by rough, uneven terrain, sharp rocks, and steep slopes. This could present a challenge to traditional wheeled rovers. Further, the motion of wheeled robots is limited to the ground plane, and thus, inherently, they do not utilize the lower gravity of extraterrestrial objects such as asteroids, the Moon and Mars. Jumping quadrupeds, on the other hand, inherently utilize the lower gravity of such objects, and in low earth gravity could potentially jump to heights of several meters TODO: CITE. This could allow them to traverse obstacles that would be insurmountable to wheeled rovers, such as steep slopes, large rocks, and gaps in the terrain.

While recent years have seen great progress in the development of quadruped robots, most quadrupeds still struggle with jumping in earth gravity TODO: CITE. Since, additionally, low gravity environments are very hard to replicate on earth, it is difficult to test hardware and control algorithms intended for low gravity jumping quadrupeds. Jumping also includes high velocity impacts, making damage to the often expensive hardware likely. This motivates the main goal of this project, which is to develop a design for a small, lightweight, and low-cost jumping quadruped robot. The robot’s low weight is intended to reduce the risk of damage during testing, and the low cost to make it more accessible to researchers, as well as reduce the cost of potential damage. Special emphasis is placed on being able to jump long distances, without losing the general utility of the quadruped form factor, such as the ability to walk on rough terrains, flexibly adjust body pose, and potentially carry scientific payloads.

---

## 1.2 Scope

As described in the Motivation section, section 1.1, the main goal of this project is to develop a design for a small, lightweight, and low-cost jumping quadruped robot. The work presented in this report is part of a specialization project, TTK4550 - Engineering Cybernetics, Specialization Project TODO: CITE, pursued at the Norwegian University of Science and Technology (NTNU), as a preparation for a master's thesis. So while the scope of the specialization project is limited to the development of a design, the overall goal is for the design to be used as the basis for a master's thesis, where the robot will be built and tested. The master's thesis will also include the development of control algorithms for the robot, which is not included in this report.

More precisely, the scope of this project is limited to the following:

- Developing a simplified simulation for the robot in MATLAB/Simulink, to be used for verification and evaluation of various design choices.
- Choosing a specific method of actuation, such as motors, parallel torsional springs, parallel extension springs, or a combination of these.
- Identifying key hardware components, such as motors and springs.
- Designing a CAD model for a single leg of the robot. The leg must adhere to geometric and mechanical constraints such as:
  - Accommodating chosen springs and motors.
  - Being easily manufacturable using 3D printing and machining TODO: WHAT?
  - Sturdiness, ie. being able to withstand the forces and impacts of jumping.

## 1.3 Related Work

The problem of robotic jumping in earth and low gravity environments has been studied by several researchers, with various approaches taken. One unique example is the Olympus robot [10] [11] developed by NTNU's ARL (Autonomous Robots lab), which uses a 5-bar linkage spring assisted leg to jump. The robot weighs TODO kg, is capable of jumping to heights of up to TODO meters in earth gravity, and has been tested in simulated low gravity environments. Another example is the 600g robot RAVEN (Robotic Avian-inspired Vehicle for multiple ENvironments) [14] developed at EPFL, which uses its bird-inspired 2 DOF multifunctional legs to jump rapidly into flight, walk on the ground, and hop over obstacles and gaps similar to the multimodal locomotion of birds. Notable for RAVEN is its geared BLDC motors, which wind up embedded torsional springs, which then assist in jumping. Apart from the different topology of the legs and springs, the concept is quite similar to that of Olympus. The RAVEN robot can jump TODO (26 cm) cm in earth gravity. A third example is the Grillo robot [12], which weighs 15g and takes off at velocities of about 30 body lengths per second, ie. 1.5m/s.

---

## 2 Theory

### 2.1 Actuator Modeling

#### 2.1.1 DC Motor Model

#### 2.1.2 BLDC Motor Model

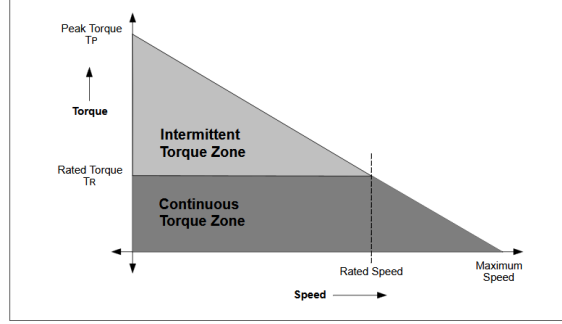


Figure 1: Torque-speed characteristics of a BLDC motor. TODO: Replace with an image that isn't stolen, and that we are allowed to use.

#### 2.1.3 Gear Transmission Friction Model

Although electric motors for robotics are available for a wide power range, they are often too high speed and low power to do any useful work. For this reason, it is often necessary to use a gear transmission to increase the torque and reduce the speed of the motor.

In the presence of a geared transmission, assuming no power loss, the output torque and velocity of a geared motor are given by equation 1 and equation 2, respectively, where  $N$  is the gear ratio,  $\tau_{in}$  is the input torque,  $\tau_{out}$  is the output torque,  $w_{in}$  is the input velocity, and  $w_{out}$  is the output velocity [6].

$$w_{out} = \frac{w_{in}}{N} \quad (1)$$

$$\tau_{out} = N\tau_{in} \quad (2)$$

In reality, however, there is always some power loss in the transmission. A common way to model this is to use friction model consisting of a viscous friction term and a Coulomb friction term [6]. The viscous friction term is proportional to the velocity of the transmission, and the Coulomb friction term is a constant friction torque that must be overcome before the transmission starts moving. The total friction torque is the sum of these two terms, as seen in equation 3. It is also possible to drop one or the other of these terms, depending on the application [6].

$$\tau_{friction} = b_{viscous}\dot{\theta} + b_{coulomb}\text{sign}(\dot{\theta}) \quad (3)$$

In addition to friction, heavily gearing motors can lead to a very high apparent rotor inertia. If one looks at equation 4, it is clear that the apparent rotor inertia is proportional to the square of the gear ratio [6]. This can lead to a very high apparent rotor inertia, which can often be problematic to robotic applications. This is especially the case for cases with contact forces, as the high apparent rotor inertia can lead to very stiff and damaging collisions [17].

$$K = \frac{1}{2}I_{rotor}(G\dot{\theta})^2 = \frac{1}{2}I_{rotor}G^2(\dot{\theta})^2 = \frac{1}{2}I_{apparent}(\dot{\theta})^2 \quad (4)$$

---

## 2.2 Spring Modeling

Springs are mechanical devices that store and release energy when subjected to displacement. There are two main types of springs: extension springs and torsion springs.

Extension springs are designed to operate with a tension load, meaning they extend as the load is applied. The force exerted by an extension spring is proportional to the displacement from its equilibrium position, following Hooke's Law, which is given by:

$$F = -kx \quad (5)$$

where  $F$  is the force exerted by the spring,  $k$  is the spring constant, and  $x$  is the displacement from the equilibrium position. The potential energy stored in an extension spring is given by:

$$U = \frac{1}{2}kx^2 \quad (6)$$

Torsion springs, on the other hand, are designed to operate with a rotational or twisting load. They exert a torque that is proportional to the angular displacement from their equilibrium position. The torque generated by a torsion spring is given by:

$$\tau = -k\theta \quad (7)$$

where  $\tau$  is the torque,  $k$  is the torsion spring constant, and  $\theta$  is the angular displacement. The potential energy stored in a torsion spring is given by:

$$U = \frac{1}{2}k\theta^2 \quad (8)$$

Both types of springs are widely used in various mechanical systems to provide force or torque, absorb shock, and store energy.

## 2.3 Spring-Damper Systems

## 2.4 Kinematics, Jacobians, and Virtual Work

### 2.4.1 Robot Kinematics

Consider a robotic link arm existing in  $\mathbb{R}^2$  consisting of  $n$  links, each with a length  $l_i$  and a joint angle  $q_i$ . The position of the end-effector is given by the vector  $\mathbf{x} = [x, y]^T$ , where  $x$  and  $y$  are the coordinates of the end-effector in the global coordinate system. Using simple trigonometry, the position of the end-effector can be expressed as a function of the joint angles and link lengths as seen in equation 9. Axes and joint angles corresponding to the expression in equation 9 can be seen in figure 2.

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n l_i \cos(q_i) \\ \sum_{i=1}^n l_i \sin(q_i) \end{bmatrix} \quad (9)$$



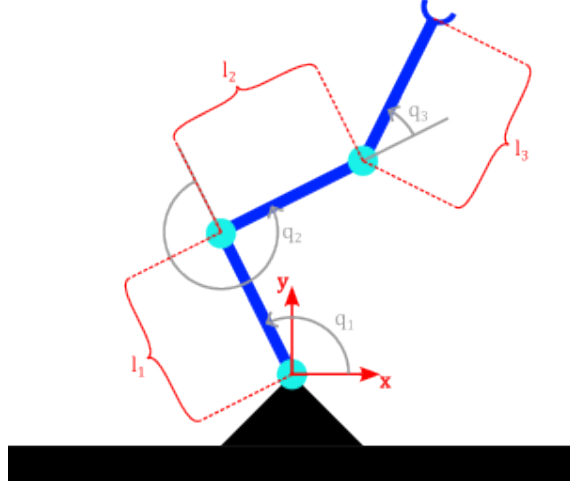


Figure 2: Illustration of a 3 link robotic link arm in  $\mathbb{R}^2$  with  $n$  links.

### 2.4.2 Jacobian Matrix

As described in section 2.4.1, the position of the end-effector can be expressed as a function of the joint angles and link lengths. In robotics, it is often useful to express the relationship between infinitesimal changes in the joint angles and the resulting change in the end-effector position. As can be seen in equation 10, infinitesimal changes in variables  $\delta y$  and  $\delta x$  can be described by means of the partial derivative [3]. If this is compared to the definition of the jacobian in equation 11, it is clear that the jacobian matrix  $\mathbf{J}$  can be used to map infinitesimal changes in joint angles to changes in the end-effector position, as illustrated in equation 12. The limit of an infinitesimal change over an infinitesimal time interval is a derivative, and thus by dividing each side in equation 12 by  $\delta t$ , one arrives at the expression in equation 13, by which the jacobian can be used to map joint velocities to end-effector velocities.

$$\delta y = \frac{\partial y}{\partial x} \delta x \quad (10)$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \end{bmatrix} \quad (11)$$

$$\delta \mathbf{x} = \mathbf{J} \delta \mathbf{q} \quad (12)$$

$$\dot{\mathbf{x}} = \mathbf{J} \dot{\mathbf{q}} \quad (13)$$

### 2.4.3 Force/Torque Mapping

Consider a general robotic manipulator, such as the one illustrated in figure 2, but with an arbitrary amount,  $n$ , of joints and links. Using the principle of conservation of power, one arrives at the formulation found in equation 14.

$$\text{power at the joints} = (\text{power to move the robot}) + (\text{power at the end-effector}) \quad (14)$$

As the power used to move the robot approaches zero,

Finally found northwestern book that says what I need: [6].

---

Consider a robotic manipulator with  $n$  joints, each with a joint angle  $q_i$  and a joint torque  $\tau_i$ . The position of the end effector for such a system is given by equation 9, and thus the formula in equation 13 can be used to map joint velocities to end effector velocities.

## 2.5 Numerical Solvers

Although a detailed review of numerical solvers and the associated theory is beyond the scope of this theory section, it is worth mentioning that numerical simulation of contact dynamics is particularly challenging. This is due to the discontinuities and high-frequency events that occur during contact. These dynamics often require the use of stiff solvers to accurately capture the rapid changes in forces and velocities [5][15]. Stiff solvers are designed to handle problems with widely varying timescales, ensuring stability and accuracy in the simulation of contact events. Without the use of stiff solvers, simulations can become unstable or fail to converge, leading to inaccurate results. Examples of stiff solvers include the well known ode15s and ode23s solvers in MATLAB, which are specifically designed to handle stiff ordinary differential equations [16].

## 3 Modeling and Simulation

For the purpose of doing design verification and optimization, a simplified model of the robot was created. The model was created in Simscape, a physical modeling toolbox integrated with MATLAB/Simulink.

### 3.1 Simscape

Simscape is a simulation tool that allows you to rapidly create models of physical systems within Mathworks' MATLAB/Simulink environment. With Simscape, physical systems are built by interconnecting blocks representing physical components, such as rigid bodies, joints and springs in a block diagram. The blocks are parameterized by physical properties, such as mass, inertia, and damping. Simscape automatically generates the equations of motion for the system, which can be solved numerically to simulate the system's behavior. Like you can do with Simulink without Simscape, you can also add ordinary Simulink blocks, including Matlab Function blocks, to the model. Simscape is also compatible with Simulink's multiple numerical solvers, such as ode15s, ode45, and ode23s (TODO? Is it ode23t?).

An example of a typical SimScape block diagram can be found in figure 3. A visualization of the corresponding model can be seen in figure 4. As one can see, each element in a block diagram is typically either a rigid body, or joints connecting the various rigid bodies. Since a given body has multiple possible locations that a joint could be connected to, as well as axes it can act on, blocks can export different frames, with different origins and orientations, depending on the desired position and orientation of the joint. For example, for a block representing the robotic equivalent of a thigh, natural output frames would be the ones with origins at the top and bottom of the thigh, with a select axis aligned with the desired knee or hip axis of rotation. TODO: We want to use something other than the tutorial here, but our robot model is too split into submodels of submodels, so it loses the intuition you get from a "flatter" model. Will fix later.

### 3.2 Rigid Body Components of the Robot Model

Since the purpose of the Simscape model is not to facilitate the development of a complicated full degree of freedom feedback controller, nor to optimize every small detail of the design, a simplified model was selected. This model consists of a main body with four legs, each of which with two degrees of freedom. A visualization of the model, as well as an overview of the body's naming conventions can be found in figure 5. An overview of the body's angle conventions can be found in figure 6. Note the absence of a hip abduction/adduction joint. This is because the model's

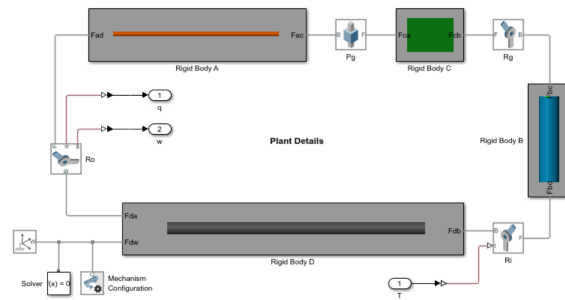


Figure 3: A typical Simscape block diagram.

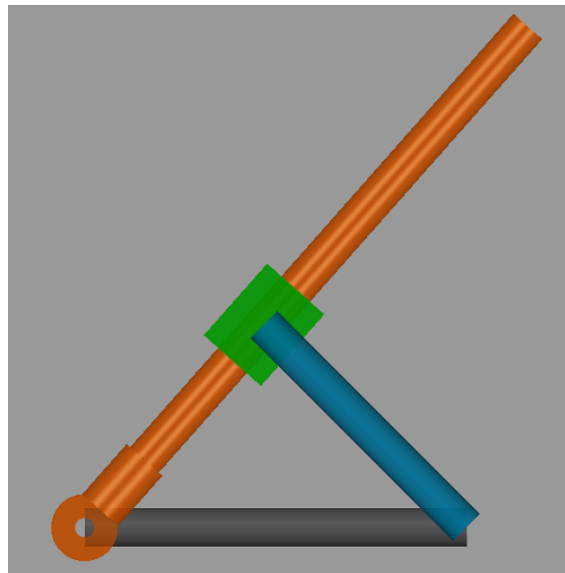


Figure 4: A visualization of the model in figure 3.

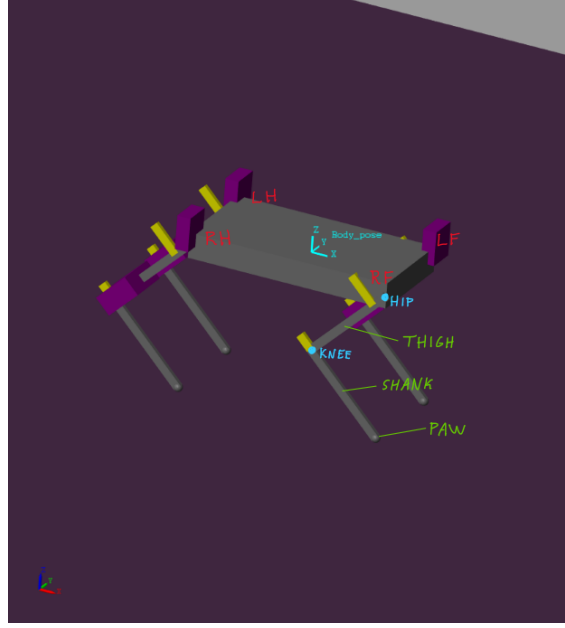


Figure 5: Naming conventions for the parts of the robot, as well as forwards direction definition. TODO: name spines

main purpose is to verify the design for jumping in the sagittal (forward-backward and upwards-downwards) plane, and the hip abduction/adduction joint is not necessary for this purpose.

Regarding the naming conventions presented in figure 5, note especially the naming of the different legs corresponding to location on the body, namely RH (Right Hind), RF (Right Front), LH (Left Hind), and LF (Left Front). Note also the naming of the joints hip (HIP) and knee (KNEE). If you see the angle conventions in figure 6, you can see that the angles of these joints correspond to the angles  $\theta_1$  and  $\theta_2$  respectively. Note that an orientation of zero degrees for the hip joint corresponds to the leg pointing straight downwards, and an orientation of zero degrees for the knee joint corresponds to the shank pointing in the same direction as the thigh.

TODO: Add body coordsys in both figures? So I can explain that positive rotation for both sides corresponds to positive rotation about the body y axis in nominal position.

As can be seen in both figure 5 and figure 6, in addition to the main body and legs colored in grey, the robot model also contains large purple blocks. These blocks represent motor masses, and their mass can be adjusted to represent different motors.

### 3.3 Rigid Body Masses and Inertias

In Simscape, the mass and inertia properties of a rigid body can be specified by the user or automatically calculated based on the body's geometry and material properties. Hybrid solutions are also possible, where the user specifies some properties and Simscape calculates the rest [7].

In the case of this model, a summary of the origin of the mass and inertia properties of the rigid bodies can be found in table 2. Exceptions are the properties of the main body and the paws, which will be specified in more detail in the two next paragraphs. For the parts whose mass and inertia are calculated by the geometry, this geometry will be specified in detail whenever the model is used, for example, section ref: TODO, the link-length optimization section, will specify the geometry assumed during the optimization.

The mass properties of the main body are based on the hardware components used by the Eurepus robot constructed by Maurer and El Agroudi [1]. Since early in the design process a rough estimate was needed for the robot body mass, and it seemed likely that the electronic solution, apart from

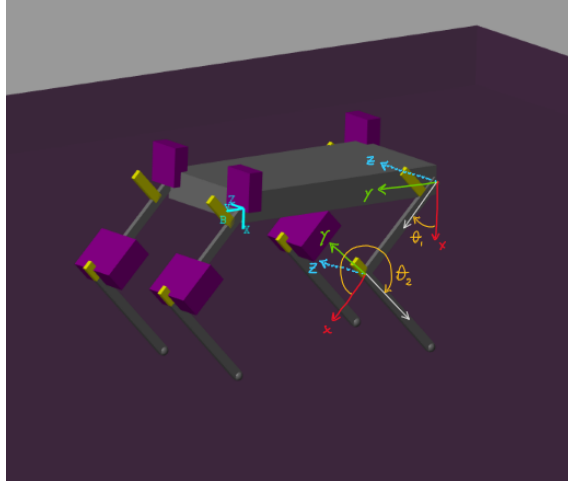


Figure 6: Angle conventions for the robot body.

Component	Mass	Density (kg/m <sup>3</sup> )	Inertia	Geometry
Main Body	See section 3.3	See section 3.3	See section 3.3	Rectangular Prism
Thigh	From geometry	2700 (Aluminium 6061)	From geometry	Rectangular Prism
Shank	From Geometry	2700 (Aluminium 6061)	From Geometry	Rectangular Prism
Law	From Geometry	2700 (Aluminium 6061)	From Geometry	Rectangular Prism
Hip Motor	Actual motor mass	From Geometry	From Geometry	Rectangular Prism
Knee Motor	Actual motor mass	From Geometry	From Geometry	Rectangular Prism
Thigh Spine	From Geometry	2700 (Aluminium 6061)	From Geometry	Rectangular Prism
Shank Spine	From Geometry	2700 (Aluminium 6061)	From Geometry	Rectangular Prism
Paw	See section 3.3	See section 3.3	See section 3.3	Sphere

Table 2: Mass and inertia properties of the rigid bodies in the robot model. A list of the Eurepus robot’s electronics can be found in [1].

the motors, would be similar to the Eurepus robot, an approximate mass of the main body was calculated based on the mass of some of the Eurepus robot’s electronics plus an approximate amount of Nylon body material, four motors, and a random chosen microcontroller (MCU) mass. The formula used for the approximate of the main body mass can be found in equations 15 to 17. The masses corresponding to variables in equations 15 to 17 can be found in table 3. The motor mass chosen in table 3 corresponds to the mass of the agfrc A20BHM motor, this motor was chosen because it is unlikely that the hip abduction/adduction motor will be any heavier than this, for details, see section 4 or robot hardware? TODO.

The geometrical, mass and inertia properties of the paws differ based on which of two scenarios we intend to simulate. The first scenario is the normal jumping scenario, in which the paw mass is simply set to 1000 kg/m<sup>3</sup>, though choosing the actual density of rubber would maybe be more appropriate. The dimensions of the paw are in this scenario simply chosen so that the diameter coincides exactly with  $max(\text{shank width, shank height})$ . The second scenario was meant to test the leg motor’s potential ability to do in air attitude stabilization like done in [1]. This scenario will be described in more detail in section TODO.

$$m_{plate} = \rho_{nylon} \cdot V_{plate} = \rho_{nylon} \cdot l_{plate} \cdot w_{plate} \cdot h_{plate} \quad (15)$$

$$m_{eurepus\_electronics} = m_{battery} + m_{I2C} + 12 \cdot m_{ADC} + m_{PWM\_driver} \quad (16)$$

$$m_{main\_body} = m_{plate} + m_{eurepus\_electronics} + m_{MCU} + 4 \cdot m_{motor} \quad (17)$$

Variable	Description	Value
$\rho_{\text{nylon}}$	Density of Nylon	1520 kg/m <sup>3</sup>
$l_{\text{plate}}$	Length of the plate	10 cm
$w_{\text{plate}}$	Width of the plate	6 cm
$h_{\text{plate}}$	Height of the plate	1.67 cm
$m_{\text{battery}}$	Mass of the battery	27 g
$m_{\text{motor}}$	Mass of one motor	20 g
$m_{\text{I2C}}$	Mass of the I2C module	5.1 g
$m_{\text{ADC}}$	Mass of one ADC module	2.4 g
$m_{\text{PWM\_driver}}$	Mass of the PWM driver	8.5 g
$m_{\text{MCU}}$	Approximate mass of some microcontroller	30 g
$m_{\text{main\_body}}$	Resultant mass of the main body	332 g

Table 3: Masses and dimensions used in the main body mass calculation.

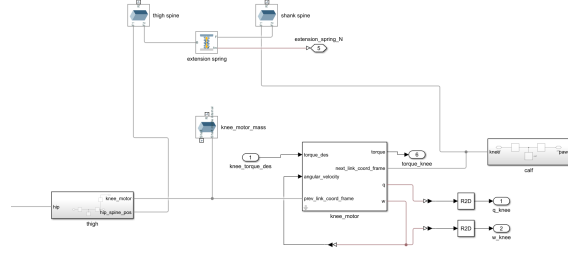


Figure 7: Extension spring implementation in Simscape. As can be seen, the spring is connected between the appropriate points (output frames) of the thigh and shank spine, and in parallel to the knee motor model, which contains the knee joint.

### 3.4 Elastic Components: Springs

In addition to the model’s many rigid bodies, we also implemented two different forms of spring based passive actuation, namely:

- **A torsional spring** acting in parallel with the knee joint, as illustrated in figure TODO. This spring is at zero extension when the knee joint is at zero degrees, and applies a torque that is proportional to the knee joint angle, as covered in section 2.2.
- **An extension spring** acting in parallel with the knee joint, attached to the shank and thigh spine, as illustrated in TODO: add figure illustrating spines/add spine description in main body figure. The force generated by the extension spring is proportional to the displacement of the shank from the thigh, as covered in section 2.2.

The torsion spring was implemented in the model using Simscape’s prismatic joint option to add spring stiffness. The extension spring, on the other hand, was connected in parallel to the joint using Simscape’s natural block diagram functionality, as seen in figure 7.

### 3.5 Motor Modeling

To be able to more accurately judge the jumping capability of different motors, a motor model in the form of a torque speed curve was implemented. The BLDC torque-speed curve presented in figure 1 is characterized by four parameters, namely the stall torque, the operating torque, the rated speed, and the maximum speed. Since motor suppliers contacted (primarily agf-rc and T-motor) were unable to provide most of the desired parameters, we chose a motor model with only two parameters, namely stall torque and maximum speed. The torque-speed model thus became a simple model where torque decreases linearly from stall torque, at speeds smaller than or equal to

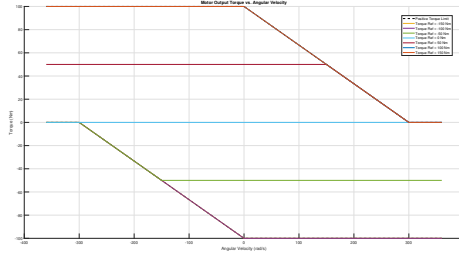


Figure 8: Torque-speed characteristics of the motor.

zero, to zero at speeds greater than or equal to maximum speed. The model is identical for negative velocities and torques, but with opposite signs. An example of the relation between desired torques and achieved torques for a given motor max speed can be seen in figure 8.

### 3.6 Solver selection

As described in section 2.2, the potential energy of a loaded spring can be easily calculated. Similarly, the potential energy due to gravity of a robot at the peak of a jump can also be determined. For the reasons discussed in section 2.5 we chose a stiff numerical solver, initially, we chose the ode15s solver. However, it was eventually observed that simulations using ode15s occasionally resulted in jump trajectories where the gravitational potential energy at the robot’s peak height exceeded the combined potential energy of the four fully loaded springs by a factor of 2, even for jumps with only passive (spring) actuation. To address this inaccuracy, we experimented with different numerical solvers and ultimately selected the ode23s solver. This solver provided accurate simulations without artificially generating excess energy, and it performed efficiently.

## 4 Robot Design

This is where we explain ”overall” design. It’s motivation, etc. Also where we cover dimensioning, link-lengths, etc. We explain here our parameter sweep to find optimal design. We also explain here our kinematics-script that we used to derive the required stall-torque for our motor given springs and link-lengths.

### 4.1 The Single Vertical Manipulator Jump Model

In the absence of air resistance, which is negligible for a robot like this during a jumping maneuver like this, the factor that determines jumping distance is body velocity at takeoff. To keep things simple, in the following will be given an example where the sole goal is to maximize vertical jump height. Further, only a one leg robot with a 2 DOF leg with equal link lengths is considered.

For a robot such as the one described above, the position of the paw relative to the body can be described by the standard 2 link manipulator equation as seen in equation 18, in accordance with the theory in section 2.4.1. For this simple robot, it is assumed that the optimal jump is one where the body center of mass and paw position move in opposite directions, with velocities strictly along the up/down y axis. This example is illustrated in figure 9. As is obvious from the figure and from the kinematics, for such a jump  $\theta_1 = -\frac{(\theta_2 + \pi)}{2}$ , and  $\dot{\theta}_2 = -2\dot{\theta}_1$ . If this is combined with the jacobian for the end effector, given in equation 4.1, one can plot the vertical velocity of the paw as a function of the knee angle  $\theta_2$ , this is shown in figure 10. This is done using the jacobian as in equation 13.

As can be seen in figure 10, the joint velocity of the leg much more readily translates to body velocity when the knee is crouched. Without giving a detailed derivation, the intuition meant to

be gained from figure 10 is that how quickly the leg joints are accelerated can be just as important as the speed it is accelerated to. This has an important interpretation for the choice of motors, in the sense that, if a given motor is unable to accelerate the leg joints quickly enough, there is little sense in looking for a faster motor, unless it is also stronger.

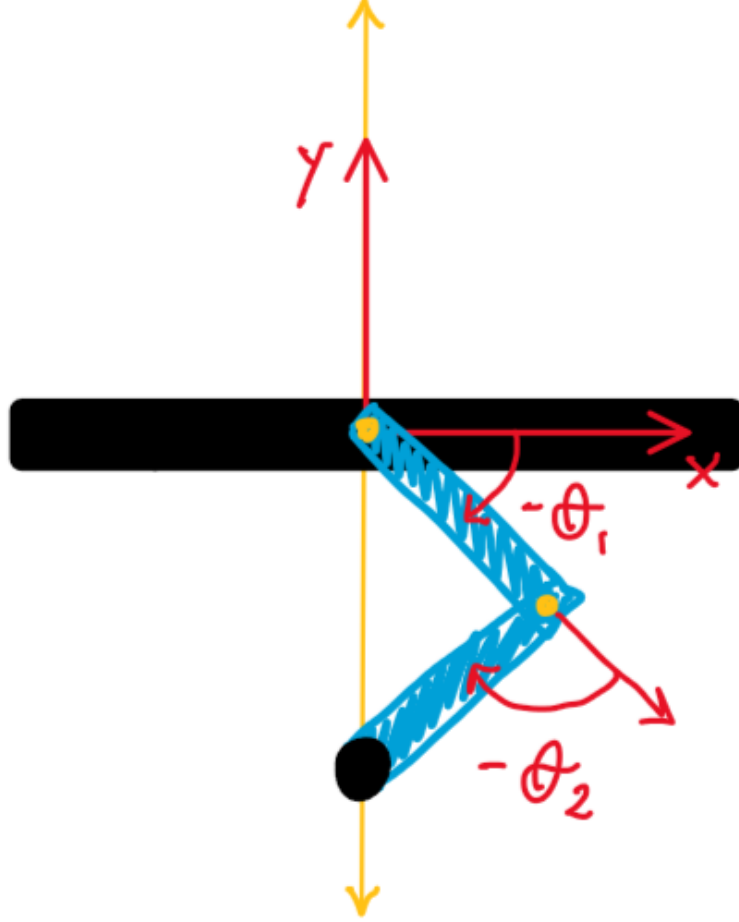


Figure 9: The manipulator corresponding to a vertical one leg jump.

$$\begin{aligned} x_{\text{end}} &= L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \\ y_{\text{end}} &= L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \end{aligned} \tag{18}$$

$$J = \begin{bmatrix} -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) & -L_2 \sin(\theta_1 + \theta_2) \\ L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) & L_2 \cos(\theta_1 + \theta_2) \end{bmatrix}$$

## 4.2 Actuation Method Selection

One of the most central decisions that had to be made for the robot was the choice of actuation method. The two main options considered were to use motors alone, or to use a combination of motors and springs.

Initially, experiments were done utilizing motors alone. Due to the need for a light and strong motor, a series of AGF-RC motors were chosen. These motors were chosen due to their high torque to weight ratio. While extensive testing was done, the jumping performance was so far from acceptable that not too much detail will be presented here. The best jumping performance



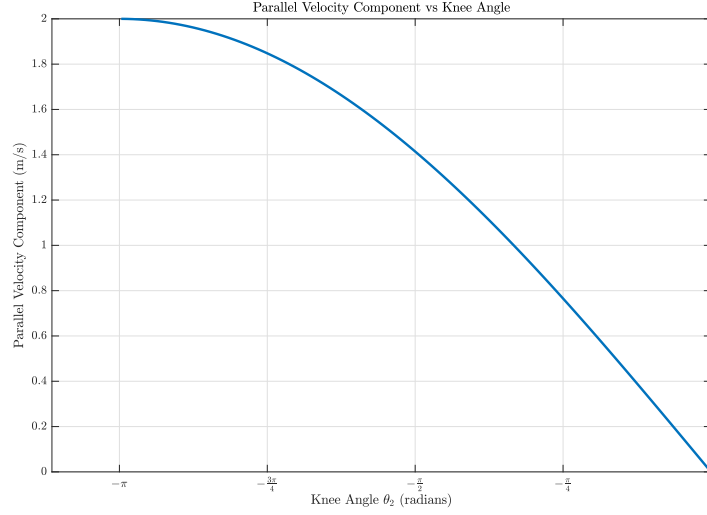


Figure 10: Vertical Paw velocity as a function of knee angle.

was achieved with the A80BHP-H motor from AGF-RC. This motor has a stall torque of 18.5 kg/cm, a weight of 80 grams, and a maximum velocity of 2000 degrees per second. To test jumping performance, the robots dimensions were gradually increased, body length as well as link lengths. This gradual increase was terminated when the body length reached 30cm and the thigh and shank links reached 20cm each, for a total leg length of 40cm. The search was terminated here because the robot was still not jumping satisfactorily, despite the robots dimensions having reached a point where they were considered excessive. Primarily because the body mass was not being scaled with the general dimensions of the robot, and because the jump height was also not increasing at a commensurate rate with the increase in robot size. Ie., even though the total center of mass displacement was increasing, the center of mass jump height compared to the total length of the legs was not increasing, and the robot was thus not jumping higher. The achieved velocity of the knee joint until takeoff can be seen in figure 11. As can be seen, the motor torque drops quickly, and is not able to reach a high enough speed to achieve a good jump. In fact, the peak body center of mass height reached is 70cm, despite a total leg length of 40cm. As can be seen, the motor is too slow to reach a high enough joint velocity for a good jump, it is therefore unlikely that a faster but weaker motor would perform much better. Efforts were made to test motors with both higher torques and higher speeds, but no motors were found with similar specs without being significantly heavier. TODO: Should I show some motors that re objectively worse, as an example?

For this reason, the decision was made to explore the use of springs in combination with motors. As mentioned earlier, two methods of spring actuation were explored, namely extension springs and torsional springs.

### 4.3

## 5 Robot Hardware

This is where we discuss actual hardware, ie. component selection, materials, CAD, etc.

So here we specify motors and how they match specs, while referring to the design section to explain why we want these motors. We should here have a list of existing motors. The design section can link to this list to motivate its own choice. It will be circular, but that's okay.

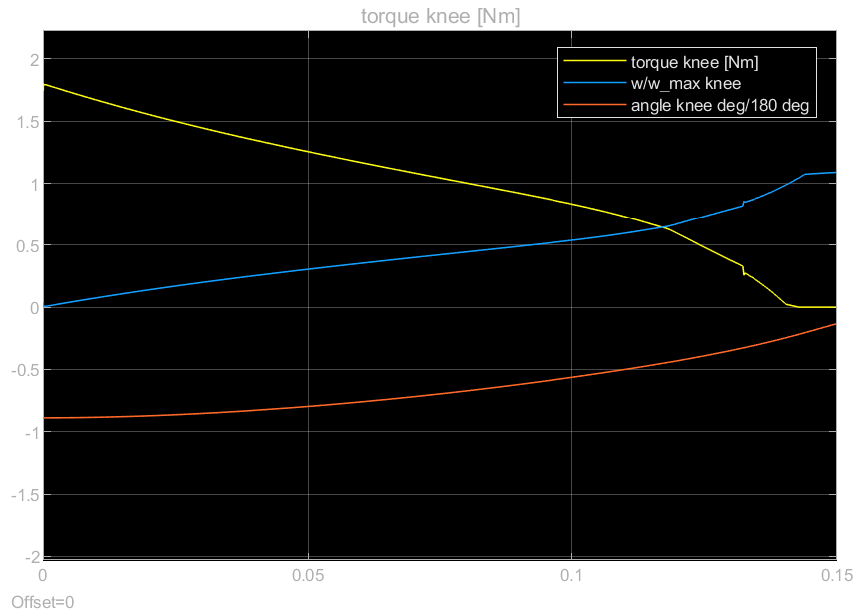


Figure 11: Knee speed until takeoff with A80BHP-H motor.

## 6 Robot Control

### 6.1 RL Problem Description?

This is not where we explain the goal of the thesis, ie. "we want to jump". This is just where we explain the RL problem. That we want to \* jump \* land \* etc. is something we describe earlier, for instance in Introduction/Scope/Problem Description.

## 7 Results

## 8 Discussion

## 9 Conclusion

---

## Bibliography

- [1] Tarek El-Agroudi and Finn Gross Maurer. *In-Flight Attitude Control of a Quadrupe in Low-Gravity Environments using Deep Reinforcement Learning*. 2024.
- [2] Andrew Daga et al. *Lunar and Martian Lava Tube Exploration as Part of an Overall Scientific Survey*. White Paper submitted to the Planetary Sciences Decadal Survey 2013-2022. 2022.
- [3] Olav Egeland and Jan Tommy Gravdahl. *Marine Cybernetics*. For ordering, visit <http://www.marinecybernetics.com> or contact [info@marinecybernetics.com](mailto:info@marinecybernetics.com). Trondheim, Norway: Marine Cybernetics AS, 2002. URL: <http://www.marinecybernetics.com>.
- [4] U. K. Ghia, K. N. Ghia and C. T. Shin. ‘High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method’. In: *Journal of Computational Physics* 48.3 (1982), pp. 387–411. DOI: 10.1016/0021-9991(82)90058-4.
- [5] Ernst Hairer, Christian Lubich and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*. Vol. 31. Springer Science & Business Media, 2006.
- [6] Kevin M. Lynch and Frank C. Park. *MODERN ROBOTICS: MECHANICS, PLANNING, AND CONTROL*. Cambridge University Press, 2017. ISBN: 978110715630.
- [7] MathWorks. *Getting Started with Simscape*. <https://se.mathworks.com/help/simscape/getting-started-with-simscape.html>. Accessed: 2024-12-15. 2024.
- [8] I. A. Nesnas et al. ‘Axel and Duaxel Rovers for the Sustainable Exploration of Extreme Terrains’. In: *Journal of Field Robotics* 29.4 (2012), pp. 663–685.
- [9] Stephanie Newdick et al. ‘Designing ReachBot: System Design Process with a Case Study of a Martian Lava Tube Mission’. In: *2023 IEEE Aerospace Conference*. IEEE, 2023, pp. 1–9.
- [10] Jørgen Anker Olsen and Kostas Alexis. ‘Design and Experimental Verification of a Jumping Legged Robot for Martian Lava Tube Exploration’. In: *2023 21st International Conference on Advanced Robotics (ICAR)*. 2023, pp. 452–459. DOI: 10.1109/ICAR58858.2023.10406863.
- [11] Jørgen Anker Olsen and Kostas Alexis. *Martian Lava Tube Exploration Using Jumping Legged Robots: A Concept Study*. 2023. arXiv: 2310.14876 [cs.R0]. URL: <https://arxiv.org/abs/2310.14876>.
- [12] Umberto Scarfogliero, Cesare Stefanini and Paolo Dario. ‘Design and Development of the Long-Jumping ”Grillo” Mini Robot’. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, pp. 467–472. DOI: 10.1109/ROBOT.2007.363830.
- [13] John Schulman et al. *Proximal Policy Optimization Algorithms*. en. arXiv:1707.06347 [cs]. Aug. 2017. URL: <http://arxiv.org/abs/1707.06347> (visited on 12th Nov. 2024).
- [14] W.D. Shin, H.V. Phan, M.A. Daley et al. ‘Fast ground-to-air transition with avian-inspired multifunctional legs’. In: *Nature* 636 (2024), pp. 86–91. DOI: 10.1038/s41586-024-08228-9.
- [15] David E Stewart. ‘Rigid-body dynamics with friction and impact’. In: *SIAM review* 42.1 (2000), pp. 3–39.
- [16] Inc. The MathWorks. *MATLAB ODE Suite*. The MathWorks, Inc. Natick, MA, 2024. URL: <https://www.mathworks.com/help/matlab/ref/ode15s.html>.
- [17] 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/TRO.2016.2640183.
- [18] Mahboubbeh Zarei and Robin Chhabra. ‘Advancements in Autonomous Mobility of Planetary Wheeled Mobile Robots: A Review’. In: *Frontiers in Space Technologies* 3 (2022). ISSN: 2673-5075. DOI: 10.3389/frspt.2022.1080291.

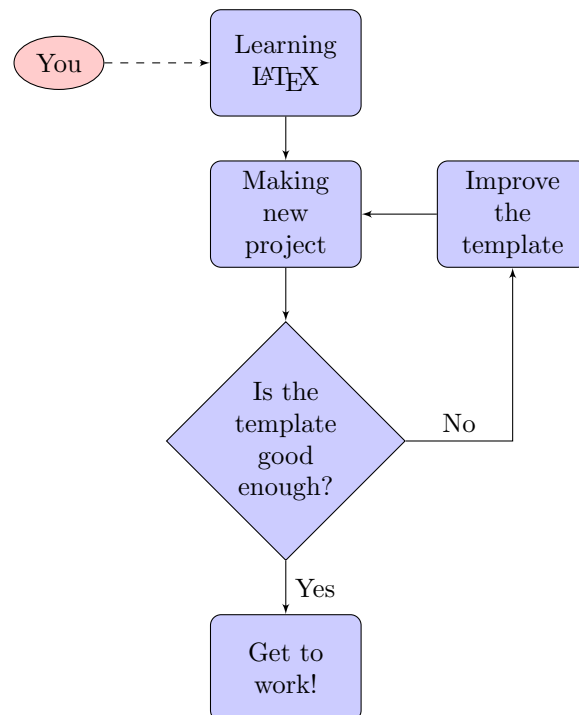
---

## Appendix

### A Hello World Example

```
int main {  
    // This is a comment  
    std::cout << "Hello World from C++!" << std::endl;  
    std::cout << "I am using the default style to print this code in beautiful colors. Since the t  
    return 0;  
}  
  
:  
:  
  
# This is a comment  
print('Hello world from Python!')  
print('I am using the "rrt" style to print this code in beautiful colors')  
  
:  
:  
  
% Content of HelloWorld.m  
disp('Hello World from Matlab!')
```

### B Flow Chart Example



## C Sub-figures Example

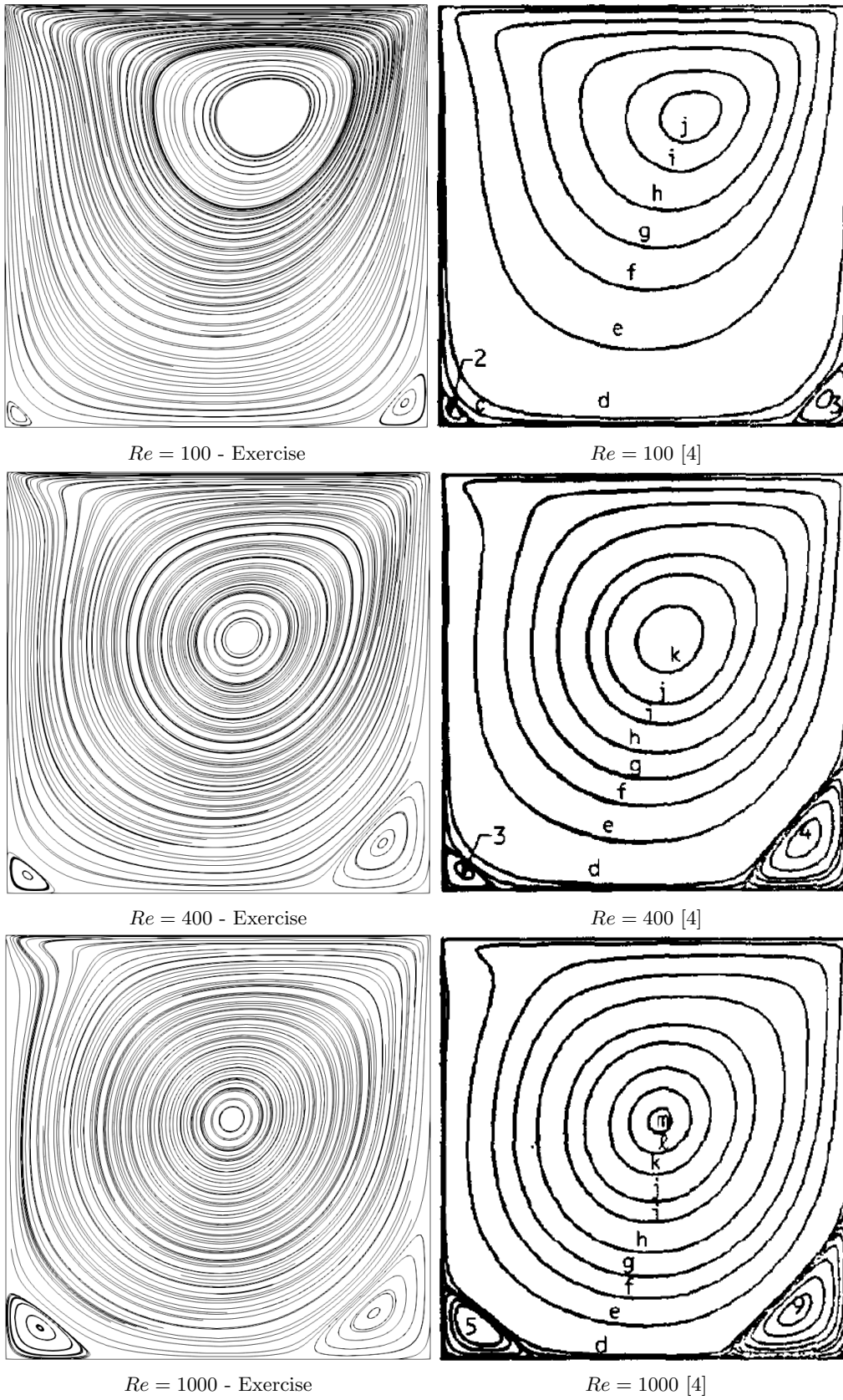


Figure 12: Streamlines for the problem of a lid-driven cavity.