NTNU
Kunnskap for en bedre verden

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 [10].

Table of Contents

List of Figures

List of Tables

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 [13], robots like Axel [5] and Reachbot [6] 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 [1]. 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 [1].

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 [7] [8] 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) [11] 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 [9], which weighs 15g and takes of 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

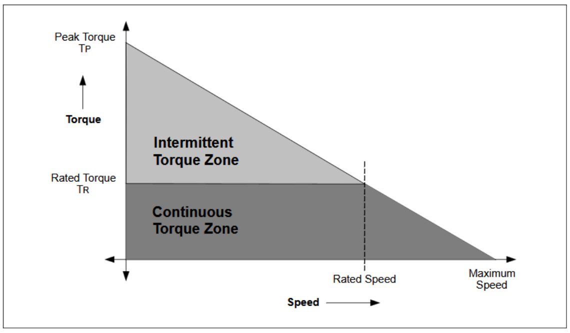


Figure 1: Torque-speed characteristics of a BLDC motor. TODO: Replace with an image that isn't stolen, and that we are allowed to use.

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, τ_{in} is the input torque, τ_{out} is the output torque, w_{in} is the input velocity, and w_{out} is the output velocity [4].

$$w_{out} = \frac{w_{in}}{N} \tag{1}$$

$$\tau_{out} = N\tau_{in} \tag{2}$$

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 [4]. 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 [4].

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

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 [4]. 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 [12].

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

2.2 Spring-Damper Systems

2.3 Kinematics, Jacobians, and Virtual Work

2.3.1 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 5. Axes and joint angles corresponding to the expression in equation 5 can be seen in figure 2.

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} l_i \cos(q_i) \\ \sum_{i=1}^{n} l_i \sin(q_i) \end{bmatrix} \tag{5}$$

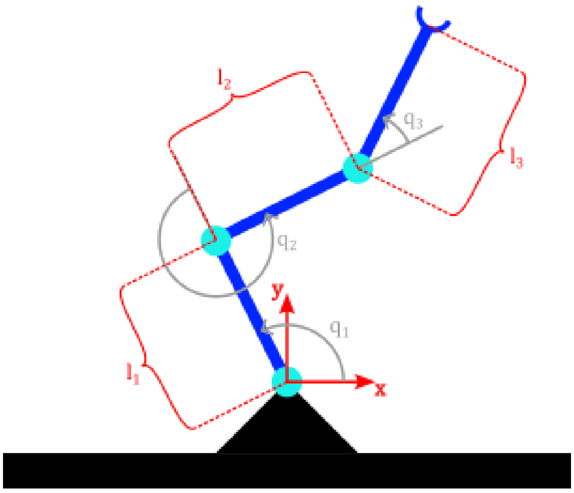


Figure 2: Illustration of a 3 link robotic link arm in $\mathbb{R}^2$ with n links.

2.3.2 Jacobian Matrix

As described in section 2.3.1, 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 6, infinitesimal changes in variables δy and δx can be described by means of the partial derivative [2]. If this is compared to the definition of the jacobian in equation 7, 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 8. The limit of an infinitesimal change over an infinitesimal time interval is a derivative, and thus by dividing each side in equation 8 by δt, one arrives at the expression in equation 9, by which the jacobian can be used to map joint velocities to end-effector velocities.

$$\delta y = \frac{\partial y}{\partial x} \delta x \tag{6}$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_n} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_n} \end{bmatrix} \tag{7}$$

$$\delta \mathbf{x} = \mathbf{J} \delta \mathbf{q} \tag{8}$$

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \tag{9}$$

2.3.3 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 10.

$$\text{power at the joints} = (\text{power to move the robot}) + (\text{power at the end-effector}) \tag{10}$$

As the power used to move the robot approaches zero,

Finally found northwestern book that says what I need: [4].

Consider a robotic manipulator with n joints, each with a joint angle q_i and a joint torque τ_i. The position of the end effector for such a system is given by equation 5, and thus the formula in equation 9 can be used to map joint velocities to end effector velocities.

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 Simscape 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 can typically consist of a physical block, or joint blocks that lie between the physical bodies they are supposed to connect. 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 Robot Body, Legs and Joints

Since the purpose of the Simscape model is not to develop a complicated full degree of freedom feedback controller, nor to optimize every small detail of the design, a simplified model was selected.

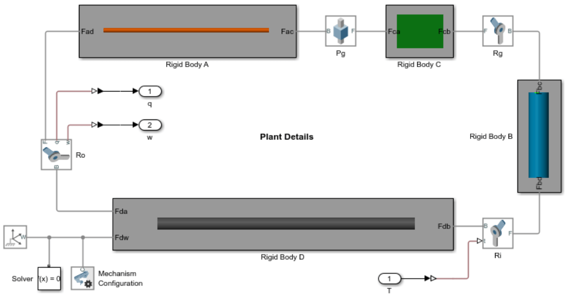


Figure 3: A typical Simscape block diagram.

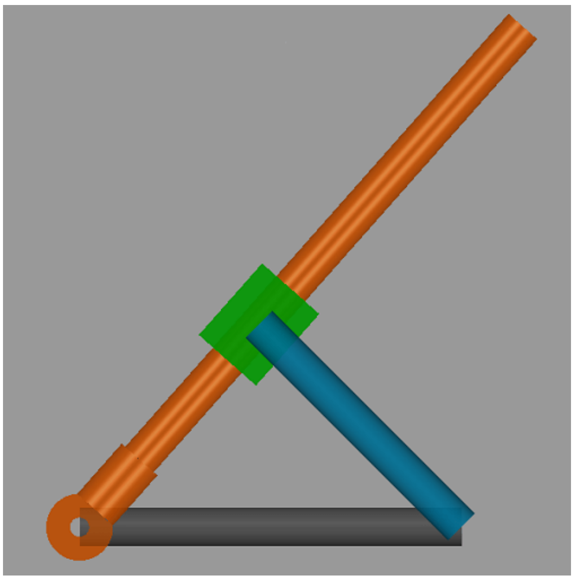


Figure 4: A visualization of the model in figure 3.

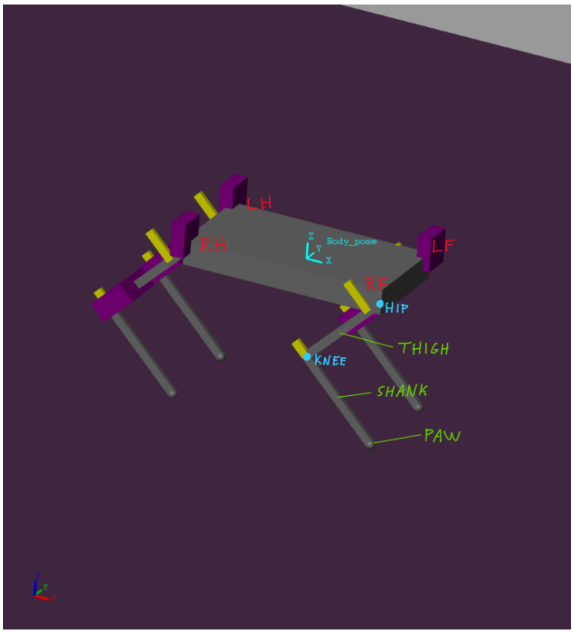


Figure 5: Naming conventions for the parts of the robot, as well as forwards direction definition. TODO: name spines

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 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 θ_1 and θ_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.

TODO: We also need to mention the motors, as well as what they weigh, what the legs weigh, that the legs are aluminum, how we got the main body mass, etc.

3.3 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 TODO. TODO: add theory
- **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.

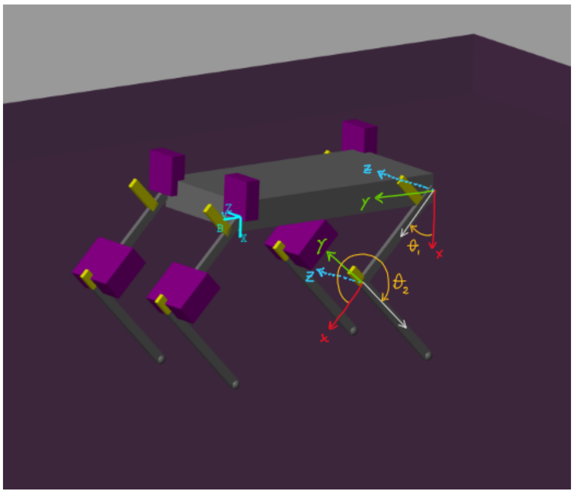


Figure 6: Angle conventions for the robot body.

Thes were both modeled according to standard simscape way, have a simscape block diagram screenshot here, TODO.

3.4 Solver selection

Mention that ode23s, and that ode15s created energy, ie. it was unphysical.

4 Link Length Optimization

4.1 Motivation

The robot must have reasonable jumping performance to be suitable for training and demonstrating the future RL control policies. One of the main design aspects that determines the jumping performance is the length of the thigh and shank links. The link lengths impact the robots mass, mass distribution, and acts as constraints for initial jumping poses and spring compression. Which in turn affect how jumps are performed and thus jumping performance. To find the optimal link lengths, a grid search was performed. To do this, a simplified model of the robot was created in Simscape, the details of this are covered in the section 3.

4.1.1 Pendulum Modeling

The pendulum used in the motor friction tests consists of an aluminum rod of length $l_{\text{arm}} = 0.19$ meters and a ballast mass $m_{\text{ballast}} = 0.301$ kg attached at a distance $r = 0.08$ meters from the pivot. The total mass of the arm is $m_{\text{arm}} = 0.034$ kg. The pendulum is modeled as a rigid body rotating about the motor shaft with a moment of inertia I given by:

$$I = \frac{1}{3} m_{\text{arm}} l_{\text{arm}}^2 + m_{\text{ballast}} r^2$$

The equation of motion for the pendulum, considering only viscous friction, is:

$$I\ddot{\theta} + b\dot{\theta} + (m_{\text{arm}} \frac{l_{\text{arm}}}{2} + m_{\text{ballast}} r) g \sin(\theta) = 0$$

where:

- θ is the angular displacement (positive counterclockwise, zero at vertical down position)
- $\dot{\theta}$ and $\ddot{\theta}$ are the angular velocity and acceleration, respectively
- b is the viscous damping coefficient
- $g = 9.81\,\mathrm{m/s^2}$ is the acceleration due to gravity

4.1.2 Linear Regression Derivation

Rearranging the equation for linear regression purposes:

$$I\ddot{\theta} + (m_{\text{arm}}\frac{l_{\text{arm}}}{2} + m_{\text{ballast}}r)g\sin(\theta) = -b\dot{\theta}$$

This can be expressed in the form:

$$Y = X\beta$$

where:

- $Y = -I\ddot{\theta} - (m_{\text{arm}}\frac{l_{\text{arm}}}{2} + m_{\text{ballast}}r)g\sin(\theta)$,
- $X = \dot{\theta}$,
- $\beta = b$.

The angular velocity $\dot{\theta}$ and acceleration $\ddot{\theta}$ are computed using centered finite differences:

$$\dot{\theta}_i = \frac{\theta_{i+1} - \theta_{i-1}}{2\Delta t}$$

$$\ddot{\theta}_i = \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{(\Delta t)^2}$$

where Δt is the time step between measurements.

The linear least squares solution for β is given by:

$$\beta = (X^T X)^{-1} X^T Y$$

This yields the viscous damping coefficient b.

4.2 Initial Pose Calculation

4.3 Symmetric versus Asymmetric legs

4.4 Grid Search

5 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.

5.1

5.2

6 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.

7 Robot Control

7.1 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.

8 Results

9 Discussion

10 Conclusion

Bibliography

[1] 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.

[2] Olav Egeland and Jan Tommy Gravdahl. *Marine Cybernetics*. For ordering, visit http://www.marinecybernetics.com or contact info@marinecybernetics.com. Trondheim, Norway: Marine Cybernetics AS, 2002. URL: http://www.marinecybernetics.com.

[3] 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.

[4] Kevin M. Lynch and Frank C. Park. *MODERN ROBOTICS: MECHANICS, PLANNING, AND CONTROL*. Cambridge University Pres, 2017. ISBN: 978110715630.

[5] 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.

[6] 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.

[7] 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.

[8] Jørgen Anker Olsen and Kostas Alexis. *Martian Lava Tube Exploration Using Jumping Legged Robots: A Concept Study*. 2023. arXiv: 2310.14876 [cs.RO]. URL: https://arxiv.org/abs/2310.14876.

[9] 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.

[10] 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).

[11] 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.

[12] 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.

[13] Mahboubeh 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;
}
```

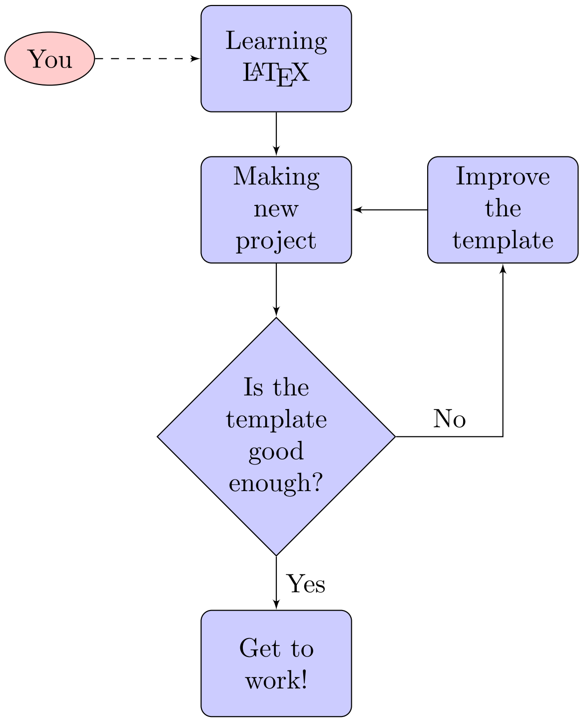
⋮

```
# This is a comment
print('Hello world from Python!')
print('I am using the "rrt" style to print this code in beautiful colors')
```

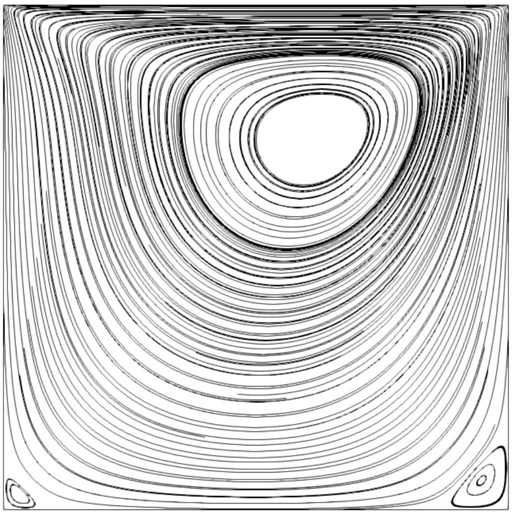
⋮

```
% Content of HelloWorld.m
disp('Hello World from Matlab!')
```

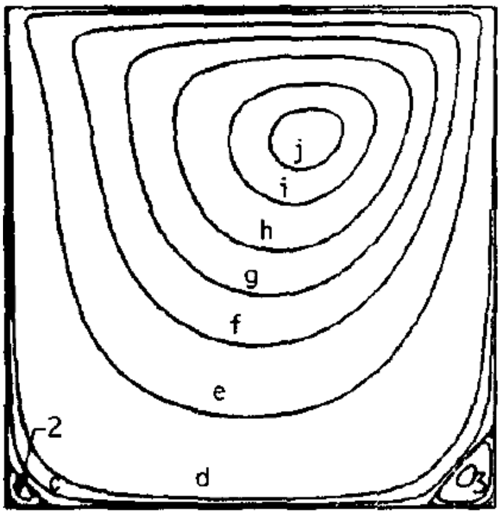
B Flow Chart Example



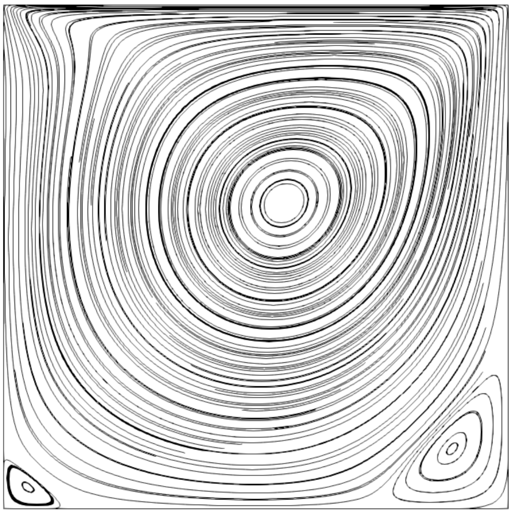
C Sub-figures Example



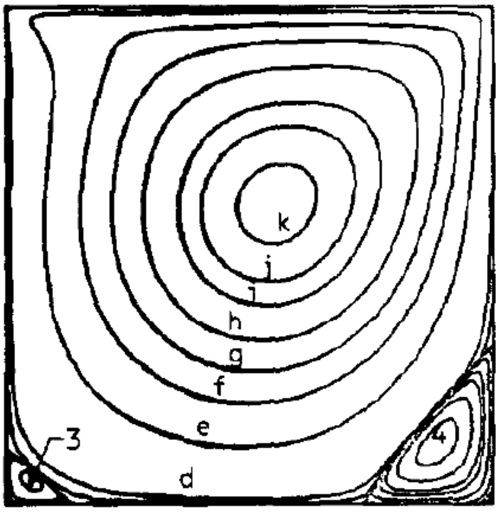
$Re = 100$ - Exercise



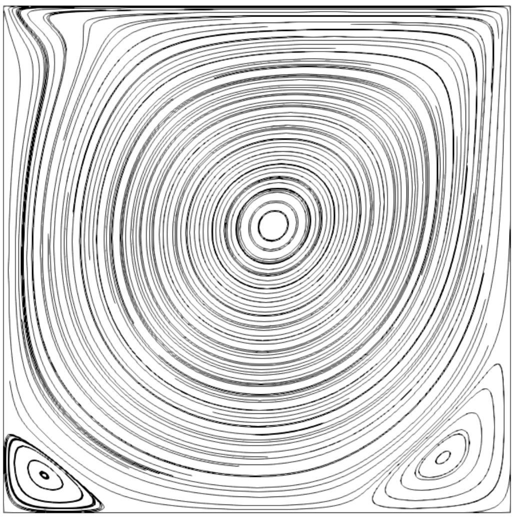
$Re = 100$ [3]



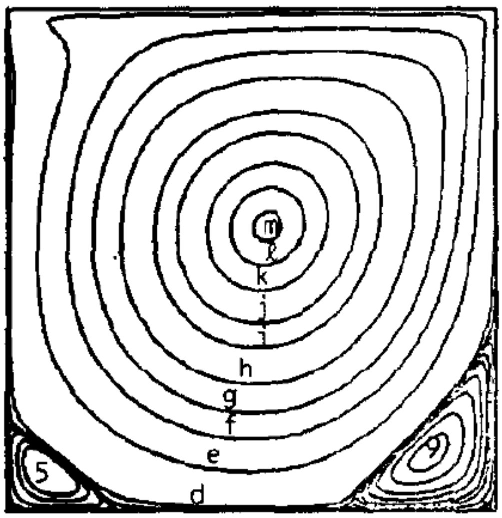
$Re = 400$ - Exercise



$Re = 400$ [3]



$Re = 1000$ - Exercise



$Re = 1000$ [3]

Figure 7: Streamlines for the problem of a lid-driven cavity.