

Machine and behaviour co- design of a powerful minimally actuated hopping robot

J.J.M. Driessen

Master of Science Thesis



Machine and behaviour co-design of a powerful minimally actuated hopping robot

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering at Delft
University of Technology

J.J.M. Driessen

Friday 30th October, 2015

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of
Technology



The work in this thesis was supported by Istituto Italiano di Tecnologia (IIT). Their cooperation is hereby gratefully acknowledged.



Copyright © BioMechanical Engineering (BMechE)
All rights reserved.

Cover Art: Gattoni, Ugo. *Panic*. 2012. Caravan Palace album poster.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
BIOMECHANICAL ENGINEERING (BMECHE)

The undersigned hereby certify that they have read and recommend to the Faculty of
Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis
entitled

MACHINE AND BEHAVIOUR CO-DESIGN OF A POWERFUL MINIMALLY ACTUATED
HOPPING ROBOT

by

J.J.M. DRIESSEN

in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE MECHANICAL ENGINEERING

Dated: Friday 30th October, 2015

Supervisor(s):

Prof.Dr. R. Featherstone

Dr.ir. H. Vallery

Reader(s):

Prof.Dr.ir. M. Wisse

Dr. S.H. Hossein Nia Kani

Abstract

This thesis presents the first steps of a design study on the robust hopping and balancing robot Skippy. The purpose of the overall design study is to challenge a new design approach for robots with high physical performance. This approach comprises an iterative study of machine (physical system) and behaviour (action strategy) co-design. Skippy is maximally simple in that it has only two actuators to control itself in 3D, and it is to be made from COTS components. It should be able to hop up to heights of 4 m and in addition be able to balance, do acrobatic manoeuvres and survive its crashes.

The goal of this thesis is to design a realistic action strategy and mechanism for Skippy in 2D to hop up to 4 m that is compliant with Skippy's required additional abilities and physical limitations. In 2D, Skippy only requires one actuator.

A study that uses the new design approach is presented in this thesis. The initial model of Skippy assumes a perfect actuator and simplified transmissions. The thesis proceeds step-by-step to a more realistic model. The physical system is designed to consist of linkage mechanisms and two non-linear passive elastic elements. The action strategy is designed such that Skippy acts in saturation. Saturation is determined by various physical limits such as the limited nut velocity of the driveline's ball screw and electrical limitations of the motor's armature. The system's behaviour is improved by a redesign of the physical system that moves the cause of the saturation, which increases the mechanical power output of the motor. This redesign includes tuning of inertial, dimensional and stiffness parameters. It is shown that Skippy is able to reach its target height while having basic control over lift-off momenta, which is required for performing acrobatic manoeuvres. In addition, Skippy's balancing performance has been investigated. By adjusting the inertial and dimensional parameters that favour balance, the jumping height is reduced to 3.8 m. This suggests that it is important to take additional abilities, like balancing, at an early stage into consideration for design decisions of the iterative design process.

Step-by-step introduction and alteration of parts and their parameters in alternation with restrictions and secondary requirements have allowed for well-understood system behaviour and have built towards a more realistic system that works.

Table of Contents

| | |
|--|------------|
| Acknowledgements | vii |
| 1 Introduction | 1 |
| 1-1 Problems in legged robotics | 1 |
| 1-2 Introducing Skippy | 3 |
| 1-3 Goal | 4 |
| 1-4 Approach | 4 |
| 1-5 Thesis Outline | 5 |
| 2 From idea to model | 7 |
| 2-1 Setting the research scope | 7 |
| 2-1-1 Morphology | 7 |
| 2-1-2 Energy and dimension calculations | 9 |
| 2-1-3 Transmission: mechanisms and components | 10 |
| 2-1-4 Passive ankle | 13 |
| 2-1-5 Requirements | 15 |
| 2-2 A first 2D model of Skippy | 16 |
| 2-2-1 Dynamic model description | 16 |
| 2-2-2 Thrust conditions | 18 |
| 2-2-3 Interim results: locked ankle | 19 |
| 2-2-4 Ankle spring profile | 20 |
| 2-2-5 Knee torque profile: thrust and steering | 20 |
| 2-2-6 Results | 22 |
| Concluding Remarks | 23 |

| | | |
|----------|--|-----------|
| 3 | Inverse dynamics approach | 25 |
| 3-1 | Knee torque profiles | 25 |
| 3-1-1 | Parabolic thrust profile | 26 |
| 3-1-2 | Continuous fuzzy transition | 29 |
| 3-1-3 | Results | 30 |
| 3-2 | Driveline modelling | 30 |
| 3-2-1 | Linear actuation kinematics | 31 |
| 3-2-2 | Spring and driveline inertia | 36 |
| 3-2-3 | Inverse dynamics control | 40 |
| 3-2-4 | Results | 41 |
| | Concluding Remarks | 42 |
| 4 | Feedforward approach | 45 |
| 4-1 | Applying rotor torques | 45 |
| 4-2 | Electrical limitations | 47 |
| 4-2-1 | Electric circuit modelling | 47 |
| 4-2-2 | Current saturation | 48 |
| 4-2-3 | Voltage saturation | 49 |
| 4-2-4 | Results | 50 |
| 4-3 | Velocity limitations | 50 |
| 4-3-1 | Nut velocity saturation | 51 |
| 4-3-2 | Results | 52 |
| 4-4 | Energy analysis | 53 |
| 4-4-1 | Ball screw friction | 53 |
| 4-4-2 | Energy flow analysis and results | 56 |
| 4-5 | Steering analysis | 57 |
| 4-5-1 | Simplified steering strategy | 57 |
| 4-5-2 | Interim steering results | 58 |
| 4-5-3 | Bilinear spring profile parameters | 58 |
| 4-5-4 | Results | 60 |
| 4-6 | Energy-based model improvements | 61 |
| 4-6-1 | Increase mechanical work | 62 |
| 4-6-2 | Improving energy conversion | 66 |
| 4-6-3 | Results | 67 |
| | Concluding Remarks | 69 |

| | | |
|----------|---|------------|
| 5 | Balancing | 71 |
| 5-1 | Physical ability to balance | 71 |
| 5-1-1 | Velocity gain | 72 |
| 5-1-2 | Simplified balancing model | 73 |
| 5-1-3 | Results | 73 |
| 5-2 | Verification | 75 |
| 5-2-1 | Controller selection | 75 |
| 5-2-2 | Electrical Limitations | 76 |
| 5-2-3 | Results | 77 |
| 5-3 | Improving balancing performance | 79 |
| 5-3-1 | Torso Mass distribution | 79 |
| 5-3-2 | Swivel actuation | 81 |
| 5-3-3 | Results | 84 |
| | Concluding Remarks | 85 |
| 6 | Discussion | 87 |
| 7 | Conclusion | 91 |
| A | Quasi-static momentum analysis | 93 |
| B | Hybrid dynamics and loop closure | 97 |
| C | Energy flows | 101 |
| | Bibliography | 107 |
| | Glossary | 111 |
| | List of Acronyms | 111 |

Acknowledgements

In June 2014, on Friday the thirteenth, the night of a full moon and in the country where Frankenstein had created his fully autonomous humanoid, I had attended the very first seminar on Skippy, held by professor Roy Featherstone. The subject grasped my attention and within a year's time I started working on this rather mind-boggling prodigy, which—by the way—did not yet exist. Primary results of the project are captured in this thesis. I am thanking Roy for his daily supervisions during my three month's visit in IIT, Genoa (Italy), for all the associated arrangements, and for all his assistance and unceasing support after my return to Delft. I would also like to thank Martijn Wisse of Delft University of Technology (TU Delft) and my father, for the feedback they have provided during the finalization of my work. Lastly, I would like to thank Heike Vallery, my supervisor at TU Delft, for her assistance during the initial stages of the project.

Delft, University of Technology
Friday 30th October, 2015

J.J.M. Driessen

To my loving oma Femmy

“Everything should be made as simple as possible, but not simpler.”

— *Albert Einstein*

Chapter 1

Introduction

This thesis presents an iterative work of machine and behaviour co-design that is the first step towards the realization of the yet theoretical robot Skippy. The project's overall purpose is to contribute to the legged robotics community by showing how to design and control robots with new levels of energetic and athletic motor skills that exceed the motor skills of any robot at the present time, and that even slightly outperform able-bodied humans. In reaching this overall goal, a new design approach is pursued, which comprises an iterative study of machine and behaviour co-design. With only two actuators, Skippy is to be able to perform a huge variety of powerful and athletic motions that have not been accomplished before. The physical design process in combination with the motion action strategy are exploited to its fullest to approach the limits of what is physically possible with commercial off-the-shelf (COTS) components. Key throughout the project is keeping the system as simple as possible and taking little steps at a time that permit no deficiency of the understanding of underlying system behaviour throughout the iterative design process. The idea behind Skippy and the design study as a whole are a response to the problems of poor physical performance faced in robotics today.

1-1 Problems in legged robotics

Research on legged robots has a variety of purposes. For one, legged robots are employable for tasks where wheeled robots are not or less applicable. The principle that legged machines have a variety of advantages over wheeled machines is well established; this axiom is supported by nature itself: galloping cheetahs, clambering mountain goats and acrobatic monkeys. Direct applications for legged robots would include extraterrestrial and Arctic *exploration* (Yoshida and Wilcox, 2008), jungly and rocky *transportation* (Buehler et al., 2005) and *industry* (Billingsley et al., 2008) and hardly passable or hazardous *search and rescue* environments (Murphy et al., 2008; Saranli et al., 2001). A secondary (more indirect) reason for studying legged robotics is a biomedical one. Legged robotics research is in many cases inspired by nature with the purpose of improving our medical understanding of humans and animals. Obtained knowledge is used for building or improving exoskeletons, prostheses and

orthoses and rehabilitation (de Boer, 2012; Karszen, 2013). Finally, legged—of which mainly humanoid—robots do well in entertainment, which consists of many big industries (theme parks, movies, commercials) (Kemp et al., 2008). Nonetheless, very few autonomous legged robots have yet been built that are successfully applied in any of these fields.

We can analyse some of the problems in legged robotics by making a distinction between versatility, energy efficiency and robustness (de Boer, 2012). Robustness can furthermore be divided in behavioural robustness and physical robustness, both related to the survivability of the robot. By behavioural robustness is meant the ability of robots to cope with disturbances from the environment, also known as disturbance rejection (Hobbelen and Wisse, 2007). Physical robustness refers to physical fragility, the ability of the robot to physically survive impacts with the environment, i.e. without breaking components. Where animals excel in all three versatility, energy efficiency and robustness, no robot has yet been built that excels in more than one. We believe that this is partly due to over-complication and neglect of the importance of the physical design.

Most prominent robots, such as Honda’s commercial ASIMO (Hirose and Masato, 2007), focus on versatility and have gained popularity for their wide variety of actions: from walking stairs, to running, to unscrewing bottles and serving food, most of which are limited to static balancing. These complex systems are often very bulky and have many actuators. The projects are big and expensive. Physical design and control are taken apart and the robots as a whole, although ingenious and appealing to one’s imagination, are no longer comprehensible nor efficient. Little is required to bring the robots out of balance and a single impact could break it. We generally seem to face problems related to robustness in large robots, not only in stiff robots like ASIMO. Even though designed to be more bionic and behaviourally robust, complex robots such as TU Delft’s TULip, Boston Dynamics’ PETman and IIT’s HyQ are still required to be carried by a harness.

A branch of robotics that *does* take into account the importance of physical behaviour, is one that bases its studies on passive dynamic modelling, such as passive dynamic walking (McGeer, 1990) or spring-loaded inverted pendula (Schwind and Koditschek, 1997; Seyfarth et al., 2002). Design of simple robots from a physical point of view that takes into account the robot’s natural dynamics has led to very cheap, understandable, and energy-efficient walking machines (Collins et al., 2005), such as Cornell’s Ranger (Karszen, 2007). However, these limited and fragile robots have shown little to no practical use, as they are able to perform only one particular action (such as walking) under specific guarded conditions. The robots lack both versatility and robustness. Initially, physical design had its primary focus on energy efficiency. Over the years, the focus has slightly been shifting towards behavioural robustness, such as in TU Delft’s Phides where non-linear parallel springs have been used to improve disturbance rejection (Karszen, 2013). The problem of robustness has been addressed by other roboticists. ATRIAS, a relatively new robot, is built with the aim of addressing both robustness and energy efficiency (Grimes and Hurst, 2012). However, no results have yet been published that live up to the objective of being fully autonomous.

Overall, we find very few successful legged robots that are both reasonably versatile and robust, such that they are sufficiently capable of operating in full autonomy. One of the legged robots that is still by many considered as most successful is Raibert’s original work (Raibert, 1986) from back in the 1980s. His work is still cited by many, and progress in the development of legged robots is slow (Silva, 2007). A few more recent successful works

include various robots from Boston Dynamics, such as the tethered quadruped Cheetah and the quadrupeds Big Dog and Spot (Boston Dynamics, 2015). The latter two are extremely powerful but also very heavy and energy inefficient, requiring heavy combustion engines to operate.

1-2 Introducing Skippy

The desire to build a robot that is both robust and versatile, in combination with the observation of problems related to over-complication and neglect of the importance of physical design, lay the foundations for a new robot named Skippy, introduced by Featherstone (Featherstone, 2014). Skippy is an exceptionally athletic but very simple hopping and balancing robot, and is designed to address the following:

- **Complexity** Skippy is maximally simple in that it only has two actuators, which is the minimum number of actuators required to achieve 3D balancing and hopping (Azad, 2014). The reduced amount of actuators reduces cost and increases reliability. More importantly, a robot this simple can be designed in greater detail and optimized to a greater degree than complex multi-actuated humanoids. Furthermore, Skippy consists of only COTS components, which contributes to the simplicity of the robot.
- **Performance** Skippy is challenged with a wide variety of required activities, from physically most demanding activities to delicate balancing, and as such reflect the desire for versatility and performance. Skippy should be able to make high powerful hops up to a height of 4 m^[1] while doing athletic manoeuvres, including somersaults and twists.
- **Robustness & Safety** It is observed that legged robotics is mainly dominated by risk management. Many robots are fragile, dangerous and prone to self-destruction due to their fragile design and high mass. Skippy is to be both behaviourally and physically robust. It should be lightweight; with a target mass of only 2 kg, it should not be harmful to the environment and it allows for designs that enable Skippy to survive its own crashes. After a crash, Skippy should be capable of standing up autonomously. It is to be able to operate in full autonomy, implying that it is also not to be allowed to be tethered or mounted to an external harness.

Comparisons

Note that many hopping robots have already been developed. Raibert's hopping machines (Raibert, 1986) are most notable as they are relatively simple hopping and balancing machines. Skippy uses only two actuators, for which it is fundamentally different from Raibert's 3D hopper, which used three actuators. Some of Raibert's hoppers make somersaults, but do not jump higher than half a meter (measured from the centre of mass (CoM)). Raibert's original 3D hopper reached a top speed of 2.2 m s^{-1} and his biped reached a top speed of 4.3 m s^{-1} . Skippy has to reach a *vertical* velocity of approximately $\sqrt{2 \times 9.81 \times 4} \approx 9 \text{ m s}^{-1}$

^[1]The physical ability of a system with an estimated weight of 2 kg to hop 4 m high is the result of a few lines of back-of-the-envelope calculations that assume mechanical energy storage and near-optimal energy flow. This is further elucidated in section 2-1-2. Together with other non-trivial requirements, this property highlights the necessity of paying attention to the physical behaviour of Skippy during the design project.

to jump up to 4 m. Festo's Bionic Kangaroo can jump up to 0.4 m (Festo, 2014). Compared to Skippy's 4 m high hop, the energy-to-weight ratio differs a factor of $4/0.4 = 10$. To jump up to a height of 4 m, Skippy has to be capable of accelerating its mass up to 9 g (see section 2-1-3), which is equivalent to carrying nine times its own mass. The strong quadrupedal robots from Boston Dynamics are not capable of carrying more than once their own mass.

A few robots exist that arguably outperform the dynamic capabilities of Skippy. Boston Dynamics' SandFlea is capable of hopping up to 8 m, but is only able to do this a few times^[iii] as it makes use of a compressed gas cylinder to make its hops (Boston Dynamics, 2012). Moreover, it is a wheeled robot. Boston Dynamics' quadrupedal Cheetah has been measured to reach velocities of 13 m s^{-1} . However, it is tethered and its (horizontal) velocity is the result of a long lasting horizontal acceleration, whereas Skippy has to accelerate to a (vertical) velocity of 9 m s^{-1} in a tenth of a second^[iii].

1-3 Goal

At present, Skippy is merely an idea with some underlying theories that support the physical possibility of the requirements stated. The execution of the idea and its theories are yet to be carried out. The goal of this thesis is to design a realistic action strategy and mechanism for Skippy to hop up to 4 m that is compliant with Skippy's required additional abilities and limitations. These include balancing abilities, controlling lift-off momenta and actuator and driveline limitations.

1-4 Approach

Because the requirements are so extreme and the problem so broad, it is not practical to pick an existing physical device and design an action strategy accordingly. The requirements demand physical system properties that are compliant with the plan of action, such as the requirement of mechanical energy storage to make a 4 m high hop with a system of 2 kg. The idea of machine and behaviour co-design is key throughout the thesis. To keep an understanding of the system's behaviour and its capabilities, we tackle the design progress iteratively. Initially, the scope of the possibilities is severely reduced by thorough reasoning and careful decisions to obtain the simplest version of a robot model capable of achieving what is believed to be most challenging, which is hopping up to a height of 4 m controllably. The model or action strategy is then expanded with new parts and parameters to obtain a larger set of solutions. Subsequently, new restrictions and requirements are introduced that shrink the set of solutions. Exploring the design path that is governed by these simplifications, expansions and restrictions could lead to a dead end. The iterative design study is then governed by backtracking of poor decisions and use of hindsight to make improved design decisions. The step-by-step approach allows us to have full overview of system behaviour and build towards a more realistic system that works.

^[iii]No specific requirement on Skippy's operating time is included, but it should not be restricted to do only a few hops.

^[iii]This assumes a stroke (acceleration distance) of 0.45 m, as further explained in section 2-1-2

Due to the time limitations imposed by the length of a thesis project, the studies presented in this thesis are limited to 2D and focus mainly on the capability of controllably hopping up to a height of 4 m. The 2D version of Skippy uses only one actuator, in contrast to the full 3D version with two actuators.

1-5 Thesis Outline

The thesis is outlined as follows. Chapter 2 provides yet undocumented preliminary studies on Skippy that set a basis for the thesis. Chapter 3 continues with the model defined in chapter 2, expanding it to be more realistic, and attempts to find new solutions through an inverse dynamics approach to perform the launch phase of the highest hop to reach 4 m controllably. Expanding the model introduces complications—caused by the aggressive motor torque command that violates driveline limitations—which are solved in chapter 4 by changing the action strategy, which is to be based on driveline limitations. With the ability of thrusting the system up to a height of 4 m controllably, other required capabilities of Skippy are to be tackled. Chapter 5 focusses on one of these abilities: balancing. The requirements for balancing deviate most from those for hopping. Other required additional abilities are not treated extensively because they did not fit in one thesis project. Results are included at the end of nearly every section in the thesis, and conclusions are drawn at the end of each chapter. Chapter 6 discusses the approach (iterative design and machine and behaviour co-design), by analysing how it has been beneficial to the obtained results and what its pitfalls are. In addition, limitations and appurtenant recommendations of the research are presented. A conclusion is subsequently found in chapter 7.

Chapter 2

From idea to model

This chapter takes the theory and idea of (3D) Skippy as defined in section 1-2 and works towards a comprehensible 2D model of Skippy that is used as a basis for the remainder of the thesis. Section 2-1 summarizes relevant previous and partly undocumented studies that were done in collaboration with R. Featherstone. The section works towards delimitations and requirements to set the research scope, for which it is essential to understand that the thesis contributes to the first steps of a large design project within this scope. Based on the scope, section 2-2 defines the basic model of Skippy used for dynamic simulations throughout the thesis. The model is planar and it assumes an ideal actuator. It is based on the original model that was provided by R. Featherstone. The chapter ends with a revamped dynamics study that uses that model, which aims to investigate the capability of Skippy to make 4 m high hops controllably.

2-1 Setting the research scope

This section sets the research scope by defining and describing Skippy's morphology in section 2-1-1 and delimitations in sections 2-1-2 to 2-1-4. Section 2-1-2 explains basic calculations that form the basis of Skippy's requirements and section 2-1-3 explains about the selection of mechanisms and components. Section 2-1-4 introduces an additional joint: a passive ankle, which is introduced to obtain desired lift-off momenta to make controllable jumps. Section 2-1-5 lists quantifiable requirements of Skippy, based on the morphology and delimitations mentioned in the previous sections, and the demands mentioned in section 1-2. This list of requirements applies to the entire Skippy design project and is not specifically composed for the studies described in this thesis, which comprise only a small part of that project.

2-1-1 Morphology

This section describes the morphology of Skippy, and summarizes the primary reasoning behind that morphology. The morphology has been determined prior to the thesis project.

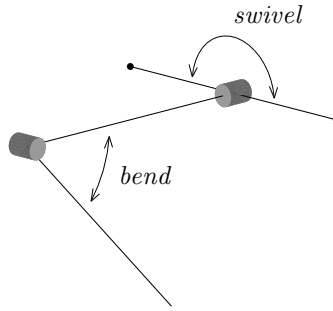


Figure 2-1: 3D morphology of Skippy, showing the actuated ‘bend’ and ‘swivel’ joints

The simplest model of Skippy consists of two revolute actuated joints and three main bodies, of which the morphology is depicted in figure 2-1.

In order to hop high, it is required to build momentum along the vertical axis. Ideally, this is achieved by matching the space of the actuator that does the ‘thrusting’ with the vertical axis using linear actuation. However, for balancing we require change of angular momentum about horizontal axes, orthogonal to the vertical axis. This implies that the linear thrust actuator cannot be used to maintain a balanced configuration, unless the thrust axis is significantly tilted (indicating a physical non-symmetry in the system), but this directly impairs the thrust performance. It has been shown that such a hopping mechanism (i.e. a non-symmetrical hopping mechanism with one linear actuator only) is indeed feasible for hopping in theory (Kuswadi et al., 2003). However, the study focusses mainly on hopping gaits and does not investigate balancing. In practice, researchers have lent towards usage of additional actuators, such as Raibert’s hopping machines (Raibert, 1986), which have an additional actuator for planar balancing and a third actuator for 3D balancing.

If we are restricted to use only two actuators to achieve 3D balancing and hopping—such as the case for Skippy—using linear actuation is possible but not effective. Revolute joints offer more (although non-perfect) solutions to achieve both balancing and hopping, owing to their non-linear mapping to the desired actuation space:

- With revolute joints, the orientation of bodies with respect to each other change during the motion. The changing non-linear dynamics of the robot during stance phase^[i] from touch-down to its lowest point, and then back up towards lift-off^[ii], contribute to build-up of all three components of momentum in the vertical plane: horizontal, vertical and angular. This build-up occurs at different ratios at different thrust-joint angles. This opens possibilities to have some control over the various momenta up to a degree, depending on the system configuration. Prismatic joints offer much less opportunity to do this.
- Revolute joints offer much better abilities to change the orientation of the robot during flight than prismatic joints.

For planar balancing and hopping, one revolute actuated joint suffices. We refer to the system with only this ‘bend’ or ‘knee’ joint as planar Skippy, which is similar to acrobot (Berkemeier

^[i]The stance phase is the phase the robot is in contact with the ground, in contrast to the flight phase, where the robot is flying.

^[ii]Lift-off is the point of transition between the stance and flight phase, the initial condition of jumping.

and Fearing, 1998). A second revolute actuator is added collinear to the bend plane (or sagittal plane) to achieve balancing in 3D, known as the ‘swivel’ joint. From bottom to top, the bodies of Skippy are referred to as leg, torso and swivel bar. The swivel bar acts as a flywheel because its CoM coincides with the axis of the swivel joint and thus the bend plane. Owing to this property and the collinearity of the swivel axis allow us to decouple the problems of controlling Skippy in 2D and 3D. The decoupled 2D system has only one actuator. This is an important property for the remainder of the thesis, which considers only 2D models that use only the bend actuator (i.e. planar Skippy).

2-1-2 Energy and dimension calculations

Back-of-the-envelope calculations and a search of COTS components form the basis of Skippy’s ambitious requirements (Featherstone, 2014), performed and provided by R. Featherstone. The calculations are explained in this section. A few of the suggested components are mentioned in section 2-1-3, as part of a study to Skippy’s transmission.

The target mass of Skippy was set to $m = 2$ kg, based on an iterative search of COTS components required for a dual actuated powerful hopping and balancing robot. The only type of actuation feasible for a robot as light as 2 kg is rotary electric motors. Hydraulic and pneumatic actuators (including equipment required to operate those actuators) and linear electric motors have a low power-to-weight ratio. With a 60 W or 90 W motor that is capable of being pulsed to 200 W, a stance phase that lasts 0.2 s is able to inject 40 J into the system. If we assume that Skippy is able to retrieve another 40 J from the previous hop—which is 50% of the total energy—then Skippy is able to jump up to a height of $80/m/g \approx 4$ m, with $g = 9.81 \text{ m s}^{-2}$ the gravitational acceleration.

The assumption of being able to store energy implies that Skippy is required to have an energy-storage spring. However, only 50% (instead of 100%) of the total energy required to reach a height of 4 m is assumed retrievable from the spring. The idea behind this is twofold. Firstly, it cannot be assumed that Skippy’s mechanical system is 100% efficient, i.e. some of the energy will be lost to heat during motion. Secondly, Skippy ought to be a powerful robot: it is to be able to jump up to 4 m in only a few hops. Skippy is not required to reach 4 m in a single hop, nor should Skippy take tens of hops before it reaches 4 m. Allowing tens of hops to reach a height of 4 m equates to designing the lightest actuated system with the most energy-efficient spring, and thus implies a preference of energy efficiency over power, which is undesired.

At the moment of lift-off, Skippy must have reached a velocity of $\dot{y} = \sqrt{2gh} = 8.86 \approx 9 \text{ m s}^{-1}$. Assuming that the launch phase takes $\Delta t = 0.1$ s, which is half the time of the stance phase, the average acceleration is $\ddot{y} = \dot{y}/\Delta t = 89 \text{ m s}^{-2} \approx 9g$. This corresponds to an average vertical ground reaction force (GRF) of $F_y = (9 + 1)gm \approx 200$ N. Correspondingly, we require a stroke of $\Delta y = 1/2\ddot{y}\Delta t^2 = 0.45$ m. The launch phase is the second phase of the stance phase between the lowest point of Skippy—where it has reached zero velocity—and lift-off. The landing phase is the first phase of the stance phase. The stroke is the distance Skippy’s CoM travels during the launch. The leg and torso should both be approximately half a meter long to realize a stroke of half a meter, to prevent Skippy from approaching a stretched (singular) configuration during the launch phase.

2-1-3 Transmission: mechanisms and components

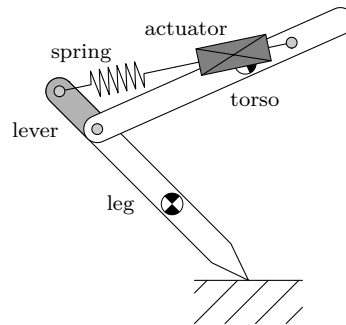


Figure 2-2: Depiction of planar Skippy with linear driveline and revolute joint. The part of the leg between the knee joint and linear series elastic actuation (SEA) (spring and actuator) is referred to as lever and drawn grey. This is not the final design of planar Skippy. The final design of planar Skippy has a four-bar linkage instead of a revolute joint at the knee (figure 2-3), and a foot that connects to the leg via a passive ankle joint (figure 2-4).

This section describes a short study on Skippy's transmission. A transmission is decided on and problems of the transmission are tackled. Additional design decisions follow from this study, including component selection.

Ball screw

From a short study prior to the thesis project on types of transmissions, it was found that a ballscrew in combination with a linkage mechanism at the bend joint was the best way forward. A few reasons contributed to the selection of a ballscrew as transmission:

1. The transmission should be able to transfer mechanical power over a distance that spans part of Skippy's torso. It is not desired to have most mass of the torso located at the bend joint. For jumping, we want to throw as much mass as possible in the air. With the mass at the far end of Skippy (i.e. towards the swivel bar), the stroke length will be longer and lift-off velocity will be higher. As such, it is undesired to have the relatively heavy motor close to the bend joint.
2. The transmission ought to be (nearly) backlash-free for purposes of control.
3. The transmission is desired to be energy efficient. Energy efficiency is not a main requirement, but we cannot afford to dissipate a lot of mechanical power to heat as we need it for Skippy to jump up to heights of 4 m.
4. The transmission should be as light as possible. Effectively, this aims for a transmission that: (1) consists of materials with a high specific strength^[iii], (2) is most efficiently loaded during the heaviest task (which is 'pulling' the knee to make a 4 m high jump) and (3) does not require a lot of pre-loading.

With respect to these points, ballscrews outdo all of the alternative transmissions that were considered. A (planetary) gear head offers only a proximal transmission and thus clashes

^[iii]Specific strength is a material's strength divided by its density

with the first point. It would have to be combined with a secondary transmission such as a belt, a chain, cables or Bowden cables. These types of multi-stage transmissions are difficult to make backlash-free or have low efficiency. Another disadvantage of pull-only transmissions is that they are weight inefficient, as never more than half of the transmission can be loaded at a time and they have to be pre-loaded. Instead of backlash-free gear heads, one could also consider harmonic drives, but these also have a low efficiency η , typically $\eta < 70\%$ (HDAG, 2014). In contrast, ball screws can be made with efficiencies higher than 95% (NSK, 2008; SBC, 2015), use a single rod to transfer both tensile and compressive forces and can be made backlash free.

The suggested transmission also has a few disadvantages, related to singularities of the mechanism depicted in figure 2-2 and more specific problems directly related to the ball screw. Both are discussed below.

Disadvantages of ballscrews are their low maximum speed, capped by the ball-return mechanism (Steinmeyer, 2015), and their fragility compared to for instance belt drives. This problem can be tackled using a series elastic element, which is also used for energy storage. The elastic element, referred to as main spring, will be connected to the nut of the ball screw and to a lever that drives the bend joint as depicted in figure 2-2 (with the ball screw and motor referred to as actuator). The spring, ball screw and motor form a linear series elastic actuation (SEA). The SEA is also referred to as the series elastic driveline.

A disadvantage of the actuation mechanism—which consists of a linear driveline and a lever—as depicted in figure 2-2, is the existence of singularities. Skippy’s leg ought to have a range of motion (RoM) of 180° , from a fully flexed to fully extended configuration. Figure 2-2 suggests that the lever has to rotate over the same range, i.e. it suggests a transmission ratio from the lever angle to knee angle of 1:1. However, the lever cannot be rotated over this range because it would cross a singular configuration. To prevent the lever from approaching a singular configuration, where the transmission ratio is extremely low, an input range of approximately 90° is desired. This problem is introduced by replacing the revolute joint at the knee in figure 2-2 for a four-bar linkage, which is described below.

Four-bar linkage

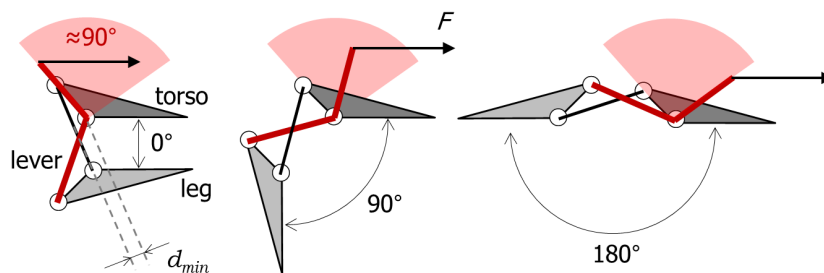


Figure 2-3: In contrast to a ‘regular’ revolute joint, the crossed-four-bar linkage or isogram creates an approximate 1:2 gear ratio and allows for a range of $\Delta\phi_{knee} > 180^\circ$. For flexing, the mechanism does not cross a singular configuration as long as d_{min} does not cross zero.

The desired transmission ratio between the angle of the lever and knee angle is approximately $\frac{90}{180} = \frac{1}{2}$. This transmission can be realized by using a four-bar linkage, such as the

one depicted in figure 2-3. Four-bar linkages have been used in robotic joints before (Khan et al., 2015; Pfeifer et al., 2012). The four-bar linkage developed by Khan, also referred to as a crossed-four-bar linkage or isogram, consists of two main bodies—which would be the leg and torso—and two links that connect to the leg and torso in a crossed configuration, closing the kinematic loop. The isogram mechanism is depicted in figure 2-3. One of the links is used as lever and rotates with approximately 90° to induce a 180° rotation between the leg and torso. Moreover, the crossed configuration makes this four-bar linkage very compact and thus very suitable for Skippy’s purpose, which is why we adopt it to be part of Skippy’s transmission.

Ball screw and motor calculations

This section expands on the back-of-the-envelope calculations in section 2-1-2 and the proposed transmission, which leads to a set of suggested dimensions and components, including the ball screw, motor and batteries.

Miniature ball screws from Steinmeyer with end cap deflectors, such as those of series 2412 (Steinmeyer, 2015), have been observed by other robotics researchers to survive rotational velocities over 6000 rpm, 33% higher than the maximum specified of 4500 rpm—which is already higher than most of its competitors.^[iv] With a linear travel of the tip of the lever of approximately $\Delta x_q = 10$ cm during the launch phase, the nut would be required to travel at an average velocity of $\Delta x_q / \Delta t = 0.10 / 0.10 = 1.0 \text{ m s}^{-1}$ without a spring. The ball screw’s spindle is available with leads of 2, 4, 5 and 8 mm, which corresponds to maximum speeds of 0.2, 0.4, 0.5 and 0.8 m s^{-1} respectively for an assumed maximum velocity of 6000 rpm. A ball screw with a lead of 8 mm can thus nearly produce the desired 1.0 m s^{-1} even without a spring, but leads as high as 8 mm require the motor to run at torques that are too high. With an estimated lever travel of $x_q = 10$ cm, the average force on the ball screw is approximately 0.9 kN, with estimated peaks that could reach up to twice this average. The corresponding torque for an $n = 8$ mm lead and $F = 2$ kN thrust force is $T = F n / 2 / \pi = 2.6 \text{ N m}$, more than twice the *stall* torque of Maxon Motor’s RE35 90 W 24 V motor^[v] brushed DC motor, whereas the torque for optimal power output is approximately at half the stall torque. The RE30 60 W and RE35 90 W series of Maxon Motor are capable of delivering sufficient power when they run at the right velocity and torque, so there is no need to step up to the heavier RE40 (150 W) series.^[vi] Instead, it was necessary to fix on a lead of 2 mm as this is the best choice for the torque-velocity characteristics of the motor.

Another possibility is changing the above assumed lever travel $\Delta x_q = 0.1$ by changing the lever length l_q (the length of the part marked grey in figure 2-2). If the lever rotates approximately

^[iv]We have also looked into ball screws of Maxon Motor and NSK, but their specifications regarding maximum transmissible force, speed and weight were lagging behind those of Steinmeyer.

^[v]Note that “90 W” in “RE35 90 W 24 V” refers to an approximation of the nominal mechanical power that it delivers for continuous operation at maximum efficiency and nominal voltage, 24 V. The maximum power the motor is capable of delivering is much higher, and approximately equal to the product of half the no-load velocity and half the stall torque, which for this motor is 245 W for 24 V

^[vi]Moreover, stepping up to heavier motors would require us to change our 2 kg budget, which was based under the assumptions of using a relatively light motor (section 2-1-2). Other series and brands of motors were also included in the comparison, but most of the optimal rotational velocities of other motors are unfavourably higher and the RE-series of Maxon Motor seemed to match our quality and price range best.

60° ^[vii] during Skippy's launch, then the lever length is approximately $l_q \approx \Delta x_q$. The maximum permissible absolute axial force on the Steinmeyer ball screw is $F_{n,max} = 3.1$ kN. Reducing the lever length further increases the maximum thrust force of approximately $F = 2$ kN on the ball screw. This can be done to an extent, but not by a factor of two. Reducing the lever length furthermore has the disadvantage of reducing the accuracy. Increasing the lever length is also advised against, because this increases the problem of limited transferable speed and furthermore increases the size of the joint, making it more vulnerable.

The lever length is assumed $l_q = 10$ cm for now, and together with a lead of 2 mm, this implies that the spring is going to play a vital role in achieving 4 m high jumps. It also implies that the spring has to have a relatively low stiffness when thrusting. To shorten the required spring elongation and to improve balancing performance—for which high bandwidth and thus high stiffness is preferred—the spring's load-displacement profile is likely to be regressive (also known as 'weakening'), implying that it is stiffer when unloaded than loaded.

The usage of 2×4 -cell lithium-polymer battery packs are assumed for motor torque-velocity calculations, with a maximum voltage of $8 \times 3.7 = 29.6$ V with an assumed voltage drop of 10% of this amount: $.9 \times 29.6 = 26.6$ V. Note that this is a pessimistic estimate; lithium-polymer batteries are known for their low voltage drops. The motors can thus be driven slightly above their nominal voltage of 24 V.

2-1-4 Passive ankle

Sections 2-1-2 and 2-1-3 have focussed only on Skippy's desired ability to make powerful jumps, which is governed by the build-up of vertical momentum during the stance phase. This section focusses on Skippy's desired ability to jump controllably, for which it is required to build up other desired components of momentum. For planar Skippy, these components of momentum are horizontal and angular momentum in the vertical plane. The desire to jump controllably leads to the introduction of a passive ankle joint.

To analyse angular momentum build-up during the launch phase, it is important to have an understanding of planar Skippy's non-symmetrical morphology and its mass distribution. Skippy is a non-symmetrical system due to its bend joint and its unevenly distributed mass. It was explained in section 2-1-2 that Skippy preferably has a lot of its mass at the far end of its body for thrust purposes. For balancing it is required to have some mass close to the knee as well. The mass of Skippy's toe (i.e. foot tip or contact point) is preferably as low as possible, because it will lose all its kinetic energy when it hits the ground during impact. The leg mass excluding toe can be higher, because its kinetic energy during impact will be absorbed by springs. In fact, the leg should have a mass to allow for reorientation during flight. Nevertheless, for thrust purposes, the torso (including swivel bar and its driveline) is going to be the heaviest component with an estimated mass of 90% of the whole system. The leg mass and inertia have a small but non-negligible effect on both balancing and thrust behaviour.

The torso and leg will counter-rotate during the launch phase, but since the torso is much heavier than the leg, it is expected that Skippy will build up a net positive angular and horizontal momentum. This theory is confirmed by a quasi-static analysis conducted in

^[viii] 60° corresponds to a knee rotation of 120° , approximately $\frac{2}{3}$ of the full 180° range

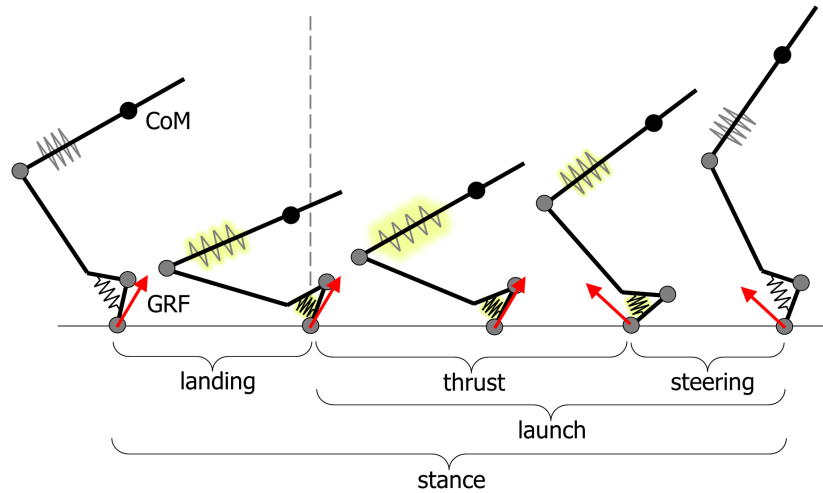


Figure 2-4: The proposal of a passive ankle to decompose the launch phase into a thrust and steering phase. The ankle unloads during the steering phase to bring back angular and horizontal momenta to zero. The images depict the loading of the main spring as well, but the lever and linear driveline are not drawn.

appendix A by R. Featherstone, and later in dynamic simulations, such as those conducted in section 2-2. Several solutions are explored to lower the build up of angular and horizontal momentum, of which some are mentioned in appendix A. Shortening the leg, which reduces momentum build-up, is *not* a feasible option for various reasons explained in appendix A but also because bringing the foot closer to the bend joint reduces the ability to do in-flight control. Initial conditions of the launch phase (i.e. when Skippy is at its lowest point) can be altered and are able to reduce the build-up of undesired momentum to some extent, in cooperation with a gravitational component. However, no strategy was found that sufficiently reduced the momentum build-up and secondly, none of the strategies were able to provide more control over lift-off momenta with the given system. Having control over momentum build-up is required to make different types of acrobatic manoeuvres.

Solutions were explored that included alteration of the physical system. A practical solution was found based on elongating the leg length during the end of the launch phase. The introduction of a passive ankle allows for prolonged contact with the ground, which is used to ‘steer’ Skippy to reduce momentum. The launch phase would then be decomposed in a thrust phase and steering phase. Build-up of vertical momentum by injecting maximum power is done during the thrust phase. During thrust, the ankle is loaded. When Skippy would lift-off without the passive ankle, the ankle will start to unload. The GRF will change direction and will point to the other side of Skippy’s CoM, depending on the action strategy used on the bend joint. While Skippy is approaching a stretched configuration, the bend joint is less capable of controlling vertical momentum, but more capable of controlling horizontal momentum. However, without the ankle, Skippy would prematurely lift-off and the bend joint does not have time to control for horizontal momentum. Moreover, during the steering phase, the unloading ankle spring will inject more kinetic energy in Skippy, such that also the steering phase contributes to build-up of vertical momentum, but to a lesser extent than the thrust phase.

The concept of the ankle joint is elucidated in figure 2-4 and tested by dynamic simulations

in section 2-2, where it is shown that the passive ankle joint works satisfactory as it allows control over planar lift-off momenta.

2-1-5 Requirements

Below follows a list of quantifiable requirements, based on the main requirements mentioned in section 1-2 and delimitations in sections 2-1-1 to 2-1-4. They are requirements that apply to the overall design project of 3D Skippy. This thesis describes a principal stage of that design project, and thus does not relate to all these requirements. The thesis investigates the design of a simplified planar version of Skippy with one actuator that could achieve the 4 m hop and would at least plausibly be capable of other behaviours as described by below list, such as balancing, even if the performance of these behaviours are not investigated in detail.

- **2 actuators** The desire for maximum simplicity is mainly reflected in the number of actuators that Skippy is allowed to have. This number is equal to two, which is the minimum number of actuators required to achieve 3D balancing and hopping.^[viii]
- **COTS components** Owing to the desire for simplicity and low-cost, Skippy is to be built from only commercial off-the-shelf (COTS) components, except for structural elements.
- **4 meter hop** Skippy should be able to hop up to heights of 4 m, measured as the distance between the COM at lift-off and the highest point it reaches in mid-air.
- **50% energy from previous hop** Skippy does not have to reach 4 m in a single hop; it is allowed to passively store some energy from the previous hop to hop higher. With 50% energy recovery, Skippy should be able to hop from 2 m to 4 m in an ideal case (where we assume 100% energy efficiency). In an unacceptably bad case, where 50% of energy is lost during stance phase, Skippy would require a hop of 4 m to reach another 4 m. In practice—where an efficiency between 50% and 100% is assumed—the result will be somewhere in between both cases, perhaps requiring a 3 m hop to reach 4 m. For now, no further assumptions or requirements are made regarding the efficiency of Skippy.
- **Control over lift-off momenta** In order to have a robot that is acrobatic or versatile in general, it is important to have control over lift-off momenta. This implies vertical momentum, but also horizontal and angular momentum. Control over lift-off momenta is achieved by the introduction of a passive ankle.
- **5° balance recovery** Related to the desired ability for Skippy to be a robust balancer, it is required that Skippy is able to recover balance for a range of 5° COM offset with respect to the contact point, for at least half the range of the bend joint (i.e. 90°).
- **180° bend RoM** Following from the desire of Skippy's autonomous ability to stand-up, its leg should be able to fully flex itself. Skippy should also be able to fully stretch itself to maximize its capabilities for balancing and acrobatic manoeuvres. These demands are the foundations for the requirement of a full 180° of the bend joint. A four-bar linkage is used to realize this.
- **two-arm swivel bar** Another requirement that follows from the desired ability of being able to stand up autonomously, is the shape of the swivel bar. As can be seen in

^[viii]Note that the number of actuators for planar Skippy is further reduced to only one actuator.

figure 2-1, the swivel bar consists of two arms, which is the minimum number of arms required to have its COM coincide with Skippy's sagittal plane. These two arms allow Skippy to make a three-point contact with the floor when standing up. Standing up could also be possible with a three-arm swivel bar, but this already significantly diminishes Skippy's compactness. Moreover, a two-arm swivel bar allows for a lighter design and for Skippy to reach lower (more flexed) positions without touching the ground.

- **Survive 4 meter fall** Skippy is a cheap and experimental platform. One of its targets is to make fast progress in robot hardware development and testing. For these reasons, it is important that Skippy survives all of its crashes from 4 m heights, whether by a sacrificial or permanent protection layer.
- **detachable leg** Skippy's leg should be detachable and replaceable by another (type of) leg, which is also related to Skippy being an experimental platform. Moreover, this allows us to put main focus on other parts of Skippy for the design process.
- **2 kg** Skippy is light-weight. The target weight of the robot is 2 kg.

2-2 A first 2D model of Skippy

This section introduces the basic dynamic model of planar Skippy that is built upon throughout the thesis. All models in the thesis are planar. The model with its parameters is described in section 2-2-1. Subsequent sections focus on dynamic simulations with this model and explain about the launch strategy to make Skippy jump with desired lift-off momenta.

2-2-1 Dynamic model description

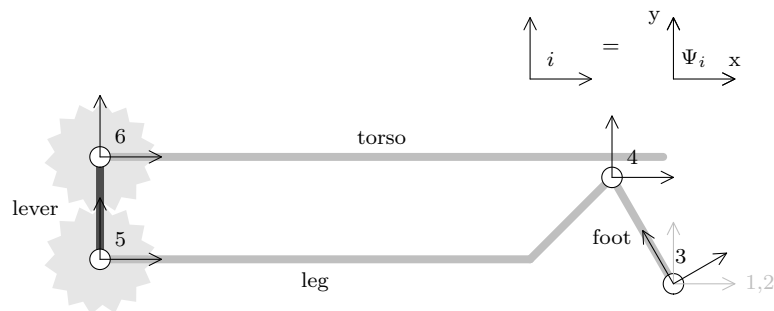


Figure 2-5: Zero configuration ($\mathbf{q} = \mathbf{0}$) of Skippy with all joint coordinate frames and a 1:1 geared knee joint. This diagram forms the basis of Skippy's planar dynamic model and is drawn to scale. Bodies are identified by name, and joints by number. The coordinate frames of the first two (fictitious) bodies coincide and are shown in pale grey.

To do simulation studies of Skippy, it is required to define a dynamic model. This section describes that model.

Skippy including the ankle is modelled as a planar mechanism with six joints and six bodies. Joints are numbered 1 to 6. Joints 3 to 6 are named toe, ankle, lower knee and upper knee respectively, as listed in table 2-1. Joints can be described as being either revolute (rotational) or prismatic (translational). The six joint positions are described by joint variables and are

elements of vector \mathbf{q} :

$$\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]^\top$$

Joint variables are either angles (in radians) or distances (in metres). Corresponding joint torques or forces are denoted $\boldsymbol{\tau} = [\tau_1 \cdots \tau_6]^\top$. Bodies are also numbered 1 to 6, and bodies 3 to 6 are named foot, leg, lever and torso respectively. Both joints and bodies are numbered sequentially from a fixed base, which is the ground, regarded as body 0. Joint i connects between body i and $i - 1$. Bodies 1 and 2 are fictitious massless bodies that hold together joints 1 to 3. Joints 1 and 2 are prismatic joints in the horizontal (x) and vertical (y) directions that determine the position of the foot. Imaginary actuators supply whatever forces are necessary to hold the toe at the ground, and to prevent the toe from slipping. The imaginary actuator forces are the horizontal and vertical GRF. Joints 1 and 2 thus provide an easy way of calculating the GRFs. q_3 and q_4 are rotational joints located at the foot's contact point and ankle, describing the foot and leg rotation, respectively. q_5 and q_6 describe the knee joint and emulate the isogram (four-bar linkage) explained in section 2-1-3, referred to as the lower and upper knee joint respectively. These are two revolute joints geared 1:1 and describe the rotation of the lever and torso respectively.

The knee angle is defined as $\phi_{knee} = q_5 + q_6$, implying a 1:2 gear ratio R_1 between the upper knee joint and knee:

$$R_1 = \frac{\partial q_6}{\partial \phi_{knee}} = \frac{1}{2} \quad (2-1)$$

The full four-bar linkage is not yet implemented because it is yet unknown how it should be dimensioned. The dimensioning mainly affects the transmission ratio. The emulation is a neutral solution and prevents us from having to add additional parameters to the model that have to be tuned. Similarly to the 1:1 geared joints, the isogram mechanism creates an approximate average gear ratio of 1:2 between the lever and knee angle (a lever rotation of 90° allows for a 180° knee rotation). For both the 1:1 geared joints and the isogram mechanism, the instantaneous point of rotation is not a fixed point relative to any of Skippy's joints. The path travelled by the instantaneous point of rotation—known as the centroid—is described by a circular arc and ellipse respectively for the 1:1 geared joints and isogram mechanism.

Model parameters are listed in table 2-1. Each body i has a mass m_i , rotational inertia about its CoM I_i and coordinates of the CoM $\mathbf{c}_i = [c_{i,x} \quad c_{i,y}]^\top$ expressed in its local frame Ψ_i . Ψ_i is embedded in body i and located at joint i if it is a revolute joint. As such, dimensional and inertial parameters of body i in Ψ_i are constant. $\mathbf{X}_i \in \text{SE}(2)$ is the coordinate transform from the parental frame of Ψ_i to Ψ_i . All bodies form a serial chain and are expressed relative to each other. As such, all parental frames are Ψ_{i-1} , i.e. $\mathbf{X}_i = (\Psi_{i-1} \rightarrow \Psi_i)$. Coordinate transforms are 3×3 matrices but can also be expressed as a 2D translation $\mathbf{r}_i = [r_{i,x} \quad r_{i,y}]^\top$ and rotation θ_i . Typically, the length l_i of body i equals $\|\mathbf{r}_{i+1}\|$. For $\mathbf{q} = \mathbf{0}$, all coordinate frames have the same global orientation except for Ψ_3 as can be seen in figure 2-5: the ankle is at rest at an angle of 60° with respect to the ground.

Most of Skippy's mass (90%) is located in the torso, which contains all the equipment and actuators, including that of the swivel bar. Rotational inertias of the torso and leg are chosen to represent values in between that of a slender beam and a point mass on a stick, as we prefer the mass to be well distributed, explained in sections 2-1-2 and 2-1-4. The position of the CoM of the torso is placed so as to achieve an approximate knee angle of $\phi_{knee} = 90^\circ$ by the end of the thrust phase, as further explained in section 2-2-2. The foot has a length of

$l_{foot} = 12$ cm and a corresponding stroke length of approximately 14 cm, which is sufficient to bring angular momentum back to zero as shown in section 2-2-6.

Featherstone's articulated-body algorithms (Featherstone, 2008c) with corresponding MATLAB software suite `Spatial_v2` (Featherstone, 2012b) are used for dynamic modelling. The dependency between q_5 and q_6 is implemented using a loop closure function $\gamma(\mathbf{y})$ and is defined as

$$\mathbf{q} = \gamma(\mathbf{y}) \quad (2-2)$$

which calculates \mathbf{q} explicitly. \mathbf{y} is the set of independent position variables. q_6 is chosen as independent position variable and q_5 as dependent position variable, so we have:

$$\mathbf{y} = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_6]^T$$

The equations of motion (EoM) including the loop closure function together with the enforced zero acceleration of the toe are defined using hybrid dynamics equations, which combines forward dynamics (FD) and inverse dynamics (ID). More information on the implementation of the hybrid dynamics and loop closure can be found in appendix B.

| i | joint name | body name | m_i [kg] | I_i [kg m ²] | \mathbf{c}_i [m] | \mathbf{r}_{i+1} [m] |
|-----|------------|-----------|------------|-----------------------------------|--------------------|------------------------|
| 3 | toe | foot | 0.05 | $m_1 0.04^2 = 8 \times 10^{-5}$ | [0 0.06] | [0 0.12] |
| 4 | ankle | leg | 0.15 | $m_1 0.17^2 = 4.3 \times 10^{-3}$ | [-0.25 -0.08] | [-0.50 -0.08] |
| 5 | lower knee | lever | 0 | 0 | [0 0.10] | [0 0.10] |
| 6 | upper knee | torso | 1.8 | $m_1 0.17^2 = 5.2 \times 10^{-2}$ | [0.38 0] | |

Table 2-1: Skippy parameters

2-2-2 Thrust conditions

This section describes a method for finding viable torque profiles for the active upper knee joint and passive ankle joint. It also provides and explains the initial conditions and thrust force profile.

The dynamic model of Skippy described in section 2-2-1 has two independent joints that do not connect to the ground: the ankle joint and the upper knee joint. However, only the upper knee torque joint is actuated. The ankle joint is passive. Nonetheless, the ankle torque τ_3 is not zero, because of the presence of an ankle spring. A spring load-displacement profile of the ankle torque τ_3 and an actuator torque profile for the knee torque τ_6 are yet to be defined. To find valid profiles for both, we propose a simulation study that is twofold: one simulation is done where the ankle joint (joint 4) is locked (i.e., held at a given fixed angle by means of an imaginary actuator) and one where it is spring-loaded and free to move. The first simulation considers only the thrust phase, whereas the second simulation considers both the thrust and steering phase. For the first simulation, a thrust force with a specified vertical component $f_{y,c}$ is applied at the foot while the acceleration of the ankle is kept zero so as to find thrust profiles for both the ankle and knee joint. These thrust profiles are used to define the ankle spring load-displacement profile and a torque profile for the actuated upper knee joint (q_6). The resulting profiles are then applied to the second simulation, where the ankle is free to move. For the thrust phase, the second simulation should have approximately the

same results as the first simulation if the ankle spring is sufficiently stiff, as a connection with a higher stiffness approaches a connection that is rigid (i.e. locked).

To do the first simulation, we have to specify the above mentioned $f_{y,c}$ and initial conditions of the system. A constant thrust force profile $f_{y,c} : \tau_2 = F_y = 200 \text{ N}$ —taken from section 2-1-2—is applied for 0.1 s to reach a thrust velocity of $v_y \approx 9 \text{ m s}^{-1}$, after which the simulation stops (i.e. no steering). The ankle is locked at $q_4(t = 0) = q_{4,0} = \pi/3 - 0.1 \text{ rad}$ which corresponds to the flexed configuration that the ankle should have during the thrust phase. The flexed configuration is chosen such that the ankle would never come close to the ground possibly causing an undesired collision. Skippy initially starts at its lowest point, where its knee is almost maximally flexed at $q_{5,0} = q_{6,0} = 0.1 \text{ rad}$ and its velocities are zero ($\dot{q}_0 = \mathbf{0}$). Here, and in the remainder of the thesis, the subscript $_0$ always indicates initial condition, i.e. $a_0 := a(t = 0)$ for any variable a .

With the leg and torso dimensions listed in section 2-2-1, Skippy reaches a non-stretched configuration by the end of the thrust phase. If Skippy would be too small, Skippy would reach a stretched configuration ($\phi_{knee} = 180^\circ$) prematurely which is to be avoided as steering would no longer be possible. At $t = 0.1 \text{ s}$, Skippy's CoM has travelled $\Delta y = 0.45 \text{ m}$, which corresponds to a knee angle of $\phi_{knee} \approx 90^\circ$ such that Skippy has sufficient travel left during the steering phase without having to approach the fully stretched configuration.

The initial foot angle $q_{3,0}$ is chosen to obtain a desired initial CoM angle ϕ_0 . ϕ is the angle between the ground (perpendicular to gravity) and the line that passes through the CoM and foot. $\phi = \frac{\pi}{2} \text{ rad}$ corresponds to a statically balanced configuration. This initial angle mainly affects the build-up of horizontal momentum, rather than angular momentum. ϕ_0 is set to a value through a few iterations of the simulation studies presented in subsequent sections (i.e. sections 2-2-3 to 2-2-6) to allow for the steering phase to bring *both* the angular and horizontal momenta back to zero at lift-off. An initial guess is $\phi_0 = \frac{\pi}{2} \text{ rad}$, which corresponds to the balanced configuration of the robot.

The results of the simulations presented in sections 2-2-3 to 2-2-6 (in which torque profiles are defined) use $\phi_0 = \pi/2 + 0.2 \text{ rad}$ (a 0.2 rad offset with respect to the balanced configuration). The preliminary simulation studies to set ϕ_0 are of little importance as they do not significantly change the dynamic behaviour of Skippy, and are thus not presented in the thesis.

2-2-3 Interim results: locked ankle

With the thrust conditions described in section 2-2-2 and the dynamic model by figure 2-5 and table 2-1 in section 2-2-1, thrust simulations of Skippy with a locked ankle have been performed. The purpose of this study is to extract torque profiles for the ankle and the upper knee joint.

Momenta during the thrust phase with the locked ankle are plotted in figure 2-6a, where it can be seen that positive angular and horizontal momenta are built up. The torque profiles that follow from the thrust force are displayed as dashed lines in figure 2-7a for the ankle and in figures 2-7c and 2-7d for the knee. The ankle and knee torques are used to devise a load-displacement profile $f_{\tau_4}(q_4)$ for the ankle spring and an actuator torque profile $f_{\tau_6}(q_6)$ for the upper knee respectively. The results are used to devise an ankle spring profile and knee thrust torque profile in sections 2-2-4 and 2-2-5 respectively.

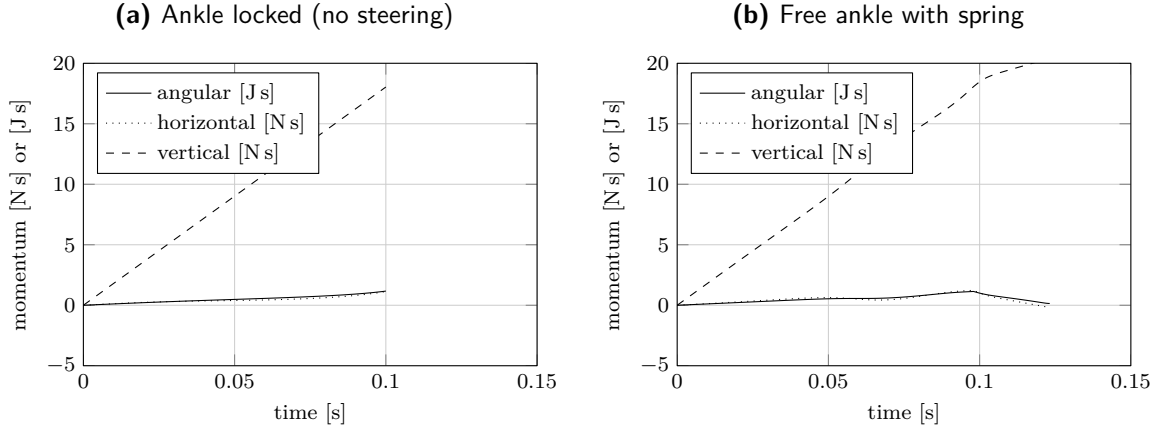


Figure 2-6: Linear and angular momenta. The ankle spring (right) brings non-vertical momenta back to zero.

2-2-4 Ankle spring profile

This section describes the desired behaviour of the ankle joint and—together with results presented in section 2-2-3—defines the ankle spring profile to obtain this behaviour.

The ankle is going to be fully compressed (thrust), at rest (balancing, lift-off) or in between these two states. When compressed or at rest, the spring should be relatively stiff such that it barely affects the system's dynamics or control bandwidth. During steering it will decompress over its full range, meaning that the stiffness can be much lower during this phase (it can almost be zero). The spring profile is thus stiff at both ends and compliant in between, from which follows that the ankle spring has a reversed S-shape (albeit monotonically increasing), as depicted in figure 2-8. A typical spring profile could be realized by making use of rubber end stops.

As can be seen from figure 2-7a, the minimum ankle torque is $\tau_{4,\min} = 14 \text{ N m}$, which is also the initial torque $\tau_{4,0}$. This implies that the spring steering or crossover torque τ_{co} —the torque around which the stiffness is very low—must be slightly lower than 14 N m and that $f_{\tau_4}(\pi/3 - 0.1) = 14 \text{ N m}$. The spring profile chosen is defined as follows:

$$f_{\tau_4}(q_4) : \tau_4 = \tau_{co} \tanh(c_1 q_4) + (\tau_{4,0} - \tau_{co}) 2^{c_2(q_4 - q_{4,0})} \quad (2-3)$$

with $c_1 = 10$ and $c_2 = 30$, two constants that affect the stiffness at rest and when fully flexed respectively. The initial angle is $q_{4,0} = \pi/3 - 0.1 \text{ rad}$ and the crossover torque is chosen at $\tau_{co} = 12 \text{ N m}$, 2 N m lower than $\tau_{4,\min}$.

2-2-5 Knee torque profile: thrust and steering

The knee torque profile consists of a thrust and steering profile, which are both described in this section. The thrust torque profile is defined based on results obtained in section 2-2-3.

The knee thrust torque τ_6 is nearly linear to q_6 , as can be seen in figure 2-7d. We have therefore defined the actuator torque as a linear ramp from the initial torque $\tau_{6,0}$ at $q_{6,0} =$

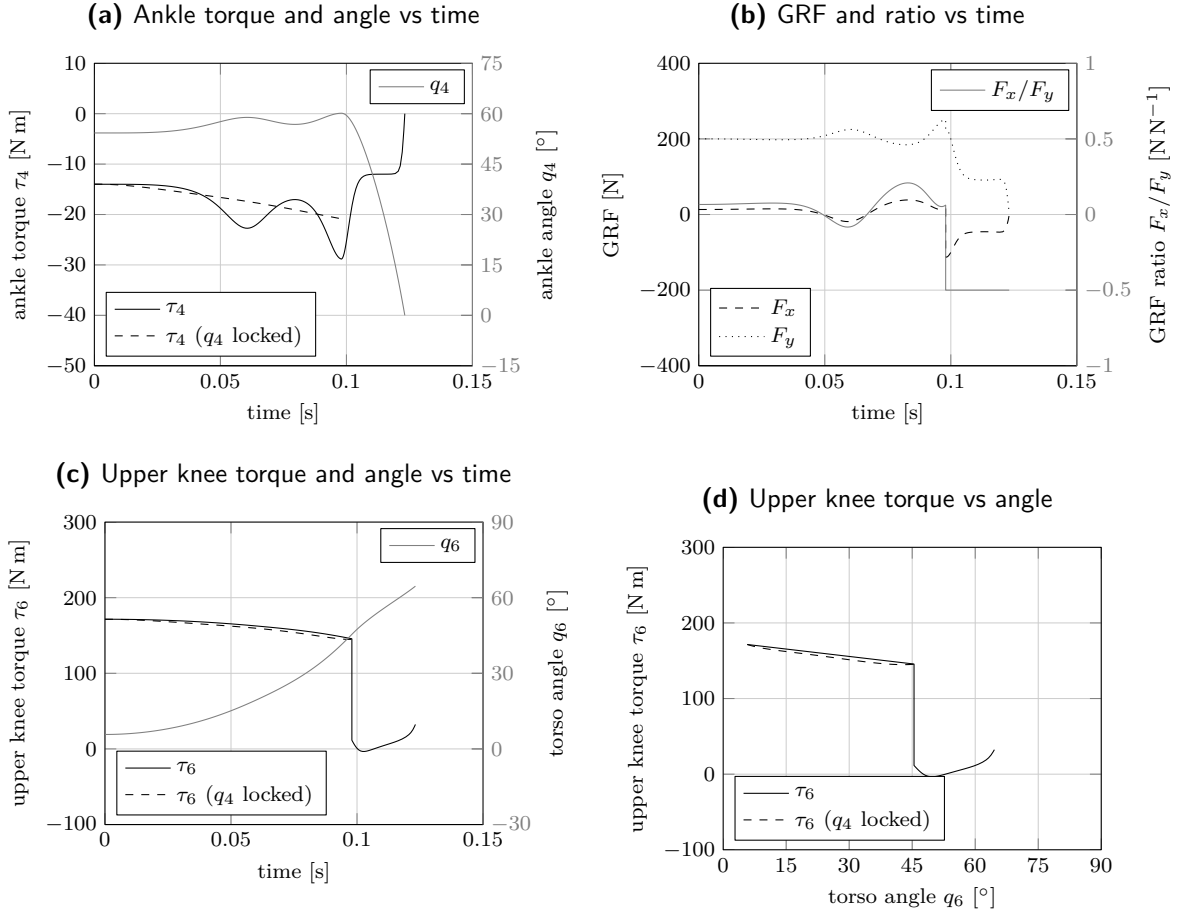


Figure 2-7: Ankle and knee joint torques and positions of the launch phase with both the thrust and steering strategy as described in the text, assuming a spring-loaded ankle. The difference between both strategies is shown clearly in figure 2-7b: during the thrust phase, F_y is kept close to 200 N regardless of the value of F_x ; during the steering phase, the ratio of these two forces is forced to be $\mu = -0.5$ for the purpose of lowering the horizontal and angular momentum.

0.1 rad (at $t_0 = 0$ s) to the torque that we measure at the end of the thrust phase $\tau_{6,f}$ at $q_{6,f} = 0.8$ rad ($t_f \approx 0.1$ s) from the simulation with the locked ankle in section 2-2-3. Note that $\tau_{6,0} = 171.5$ N m is directly calculated from the desired initial thrust force $F_y = 200$ N.

$$f_{\tau_6 t}(q_6) : \tau_6 = \tau_{6,0} - (\tau_{6,0} - \tau_{6,f}) \frac{q_6 - q_{6,0}}{q_{6,f} - q_{6,0}} \quad (2-4)$$

This thrust torque is held until Skippy's upper knee reaches an angle $q_{6,f} \simeq 0.8$ rad (corresponding to $\phi_{knee} \simeq 90^\circ$), which is the angle at which we switch to the steering phase. $q_{6,f}$ is retrieved from the simulation with the locked ankle where Skippy reached an upward velocity of $\dot{y} = 9$ m s⁻¹, approximately at $t = 0.1$ s, but slightly earlier. From this point onwards, the upper knee steering torque profile $f_{q_6,s}(\mathbf{q}, \dot{\mathbf{q}})$ is computed so as to put the direction of the GRF on the edge of an imaginary friction cone with slope $\mu = F_x/F_y$, effectively adjusting the hybrid dynamics. μ can take any value approximately between -1 and 1 to prevent slipping, but should be negative to correct for the positive angular and horizontal momenta that are

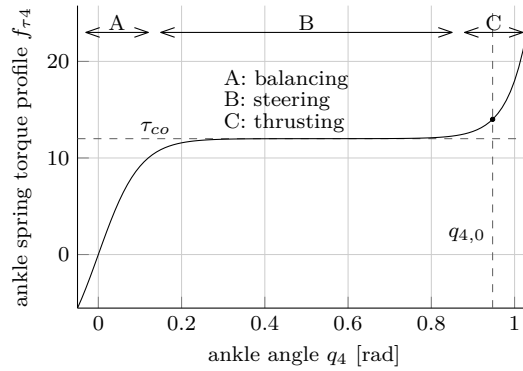


Figure 2-8: Load-displacement profile of ankle spring: stiff at both ends for balancing and thrust and compliant in between for steering.

built up during the thrust phase.

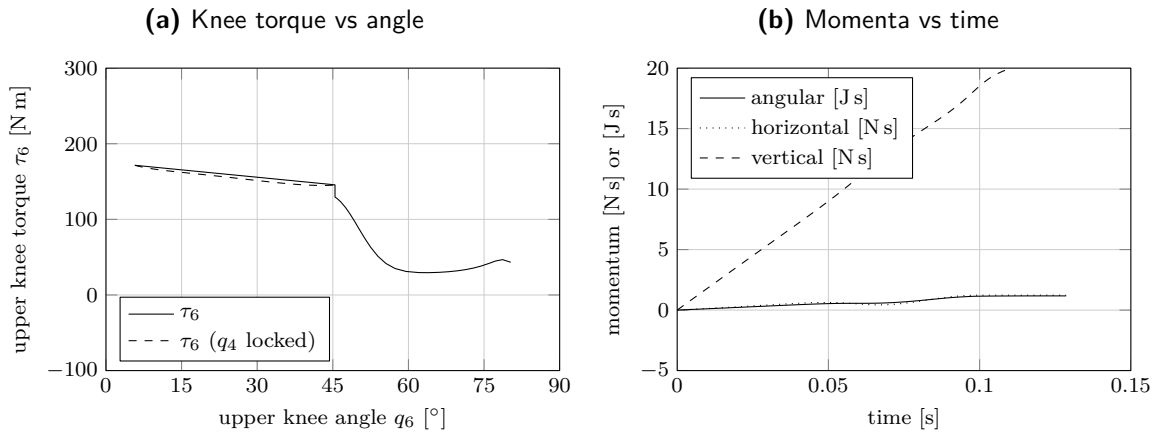


Figure 2-9: The plots show that a different steering action ($\mu = 0$ instead of $\mu = -0.5$ in figures 2-6 and 2-7) can be used to obtain different lift-off momenta, confirming the controllability of the 4 m hop launch.

2-2-6 Results

The results of Skippy with a spring-loaded ankle (with the ankle spring profile and knee torque profile defined in sections 2-2-4 and 2-2-5 respectively) are presented below.

The effect of the spring-loaded ankle is best observed in the momenta plot in figure 2-6b. The results for the thrust phase are nearly the same as those of the system with the locked ankle, except for a very small wobble caused by the ankle spring. The ankle has obvious force oscillations as can be seen in figure 2-7a, but the position oscillations are very small due to the high stiffness when fully flexed such that the dynamics are barely affected.

When switching to the steering phase, the load on the ankle drops significantly which causes it to unload. For $\mu = 0.5$, we are able to bring back both the angular and horizontal momentum, as can be seen in figure 2-6b. A different steering action, such as $\mu = 0$ (implying only a

vertical reaction force) leads to different lift-off momenta, as can be seen in figure 2-9b. This implies improved controllability over lift-off momenta, which was one of the intentions of the ankle spring. The steering furthermore still contributes to thrust, as the released energy of the ankle spring is mainly transferred to vertical translational energy. This can be seen in figure 2-6b, where vertical momentum is still built up, albeit slightly less than during the thrust phase.

We have furthermore experimented with altered ankle spring profile parameters, but their effect on the global system dynamics seemed negligible. For instance, changing the stiffness changes the oscillation frequency but the time of unloading is barely affected.

The GRFs and their direction are plotted in figure 2-7b, from which can be seen that Skippy does not slip ($|F_x/F_y| < 1$) and it can be confirmed that $F_x/F_y = -0.5$ during steering.

The main problem encountered is the discontinuous switch between the thrust torque and steering torque, as can be seen in figures 2-7c, 2-7d and 2-9a. This discontinuity is not reproducible in reality due to the compliant driveline.

Concluding remarks

This chapter has worked towards a simplified planar model of Skippy based on requirements of the physical model that are defined in the beginning of the chapter, with the main purpose of allowing Skippy to hop up to a height of 4 m controllably. The introduction of an ankle spring and launch strategy, where the launch phase is decomposed into a thrust and steering phase, allow us to obtain desired lift-off momenta: Skippy is able to reach a height of 4 m and we have some control over horizontal and angular momentum at lift-off, which has been confirmed by the dynamics study in section 2-2. The latter suggests that we have control over the 4 m hop motion.

The behaviour of the ankle with its current dimensions and profile is satisfactory, and since the leg of Skippy is going to be detachable and replaceable, we are not going to put main focus on optimizing its parameters. However, the transition between the thrust phase and steering phase is discontinuous and the actuated joint is modelled as a perfect torque source. Towards the design of a realistic launch strategy, more viable torque profiles are to be developed and, subsequently, more realistic actuator and driveline models have to be implemented, which is tackled in chapter 3

Chapter 3

Inverse dynamics approach

This chapter builds on the study using the first model of Skippy conducted in section 2-2. The goal of this chapter is to find a pre-defined knee torque profile that can be reproduced by the motor through the series elastic driveline. Section 3-1 aims to find a more realistic and broader set of knee thrust torque profiles than the one found in section 2-2. In section 3-2, we implement the linear and series elastic driveline in Skippy which put limitations on the capabilities of the input, which up till now have assumed to be a perfect torque source. The section then investigates the reproducibility of the new knee torque profiles defined in section 3-1 on Skippy with the newly defined series elastic driveline.

3-1 Knee torque profiles

Recall that it is necessary for Skippy to have a spring in series with the motor for Skippy to reach a height of 4 m. This spring has to be located in the driveline—i.e. between the knee joint and motor—because of the limited velocity of the nut of the ball screw (see section 2-1-2). The thrust profile obtained in section 2-2-6 is the desired thrust profile at the knee joint. However, this profile is not reproducible by a motor connected in series with a spring, as the instantaneous force jump requires an instantaneous position jump of the spring, which is physically impossible. The instantaneous force jump implies that the order of continuity is C^{-1} . The series elastic driveline requires the order of continuity to be at least C^1 . The following steps are taken to find a feasible continuous torque profile:

- Section 3-1-1 implements a different thrust profile to reduce the discontinuous step between the thrust and steering phase.
- Section 3-1-2 implements a fuzzy crossover between the thrust and steering phase to remove the remaining discontinuity.

Results are discussed in section 3-1-3.

3-1-1 Parabolic thrust profile

This section aims to introduce a more realistic thrust profile than the one based on a constant force profile for F_y defined in section 2-2, by replacing that constant profile for a parabolic thrust profile.

The constant thrust force of 200 N leads to an undesired high knee torque at the end of the thrust phase. Moreover, the time derivative of this thrust force is always zero, whereas it ought to be positive initially. The initial configuration of the thrust phase corresponds to the lowest point Skippy reaches, where the velocities of all bodies in Skippy are approximately zero. Initially, the main spring is already partly charged with energy from the previous hop, but will not discharge until shortly after the beginning of the thrust phase. This is related to a non-symmetry of the thrust and landing phase, where we want the nut to do positive work throughout both the thrust phase and most of the landing phase, including the transition from landing to thrust. This implies that the main spring's initial velocity is non-zero, and that it is being charged. The rate of the thrust force is then also positive because the spring's load-displacement profile is monotonically increasing. As a result, we are looking for a new thrust force profile $f_y(t)$ that satisfies

$$f_y(t_f) < f_{y,c} \quad (3-1)$$

$$\dot{f}_y(0) > 0 \quad (3-2)$$

$$\int_0^{t_f} f_y(t) dt = \int_0^{t_f} f_{y,c} dt \quad (3-3)$$

with $f_{y,c} = 200$ N the old thrust torque. Equation (3-3) implies constant vertical momentum at lift-off—and thus equal jumping height. The new transition time t_f will not be exactly the same as the old one, but for a first approximation it is reasonable to assume that it is. Another option is to make the thrust profile displacement-based rather than time-based: $f_{yy}(y)$, where y is the vertical position of Skippy's COM. The integral in equation (3-3) would imply constant kinetic energy assuming that the transition position y_f is the same for both profiles. Both t_f and y_f are expected to change slightly, so neither profile is fully accurate. However, one of the difficulties related to a displacement-based profile, is that $\dot{F}_{y,0} > 0$ and $\dot{y}_0 = 0$ imply that $\partial F_{y,0}/\partial y \rightarrow \infty$. Note that if F_y is a function of only y , and y a function of only t , the only independent variable is t , and we can write:

$$\dot{F}_{y,0}(y(t)) = \frac{dF_{y,0}}{dt} = \frac{\partial F_{y,0}}{\partial y} \underbrace{\frac{dy_0}{dt}}_{\dot{y}_0=0} > 0 \Rightarrow \frac{\partial F_{y,0}}{\partial y} \rightarrow \infty$$

The following parabolic thrust profile satisfies the conditions in equations (3-1) to (3-3):

$$f_y(t) : \tau_2 = f_{y,c} - \frac{d}{c^2}t^2 + \frac{2d}{c}t \quad (3-4)$$

which is conveniently chosen such that it also satisfies $f_y(0) = f_{y,c}$. The profile is depicted in figure 3-1. c is a third of the estimated duration of the thrust phase, i.e. $c = 0.1/3$ s. d is a value to be experimented with and is a measure for the appearance of the parabola. For $d = 0$, $f_y(t) = f_{y,c}$; increasing d allows us to reduce the discontinuity between the thrust and steering phase. However, we have that $f_y(t_f) = f_{y,c} - 3d$, implying that for $d > f_{y,c}/3 \approx 67$

N , $f_y(t_f) < 0$, which means that Skippy would lift-off prematurely. Moreover, high values of d might lead to premature relaxation of the ankle spring, which could lead to undesired system behaviour.

A thrust torque profile for the knee is then devised in the same way equation (2-4) was devised. However, this time we derive two profiles: one as function of the upper knee angle $f_{\tau 6t}(q_6)$ and the other as function of time $f_{\tau 6t}(t)$. We prefer the thrust torque to be defined as function of angle for control purposes, similar to equation (2-4). However, we have that $\partial\tau_{6,0}/\partial q_6 \rightarrow \infty$ because $\dot{y}_0 = 0 \leftrightarrow \dot{q}_{6,0} = 0$ and $\dot{F}_{y,0} > 0 \leftrightarrow \dot{\tau}_{6,0} > 0$. This is troublesome for numerical integration, especially if it will be required to calculate derivatives. In these cases, $f_{\tau 6t}(t)$ (as function of time) is called instead of $f_{\tau 6t}(q_6)$ (as function of angle).

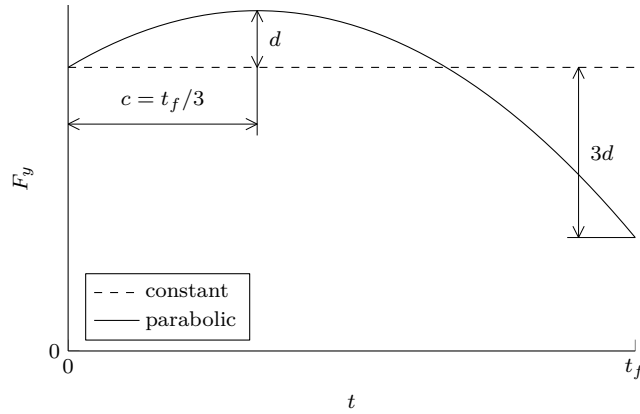


Figure 3-1: Parabolic thrust profile for F_y : the surface area below that of the parabola and constant are equal. This new profile aims to improve the old (constant) profile by having an initially positive rate and lower final value.

We follow the same approach as in section 2-2. Firstly, we lock the ankle from which we can retrieve knee thrust torque profiles based on inverse dynamics. This will be done for multiple values of d to find out how d affects the torque profiles. Next, we will apply the various torque profiles—including the unaffected steering phase—to the model with the ankle spring.

Parabolic thrust profiles $f_y(t)$ for values of d up to $d = 80$ N were tested on Skippy with the locked ankle. The vertical velocity of the COM \dot{y} as function of time t and d is plotted in figure 3-2a. The target velocity of $\dot{y} = 9 \text{ m s}^{-1}$ is reached slightly before $t = 0.1$ s, corresponding to what was found in section 2-2-5. Strongly parabolic thrust profiles reach the target velocity slightly earlier because the initial work delivered is higher, which also affects $q_{6,f}$.

For $d > 67$ N, the system starts to decelerate after it has reached its target velocity due to the negative (i.e. pulling) thrust force F_y , but this data is not of interest and thus not considered in future analyses. The ankle torque as a function of t and d is plotted in figure 3-2b, including a plane of the crossover torque $\tau_{co} = 12$ N m. For higher values of d ($d \gtrsim 30$ N), the ankle torque goes below the crossover torque during the thrust phase. This could lead to premature unloading of the ankle, but is not necessarily the case if the ankle is free to move.

To find a fit for the thrust torque as function of upper knee angle $f_{\tau 6t}(q_6)$, we have to consider the special boundary condition $df_{\tau 6t}(0)/dq_6 \rightarrow \infty$. The fit uses a square root to meet this boundary condition. After an initial increase, τ_6 reaches its maximum and goes down almost

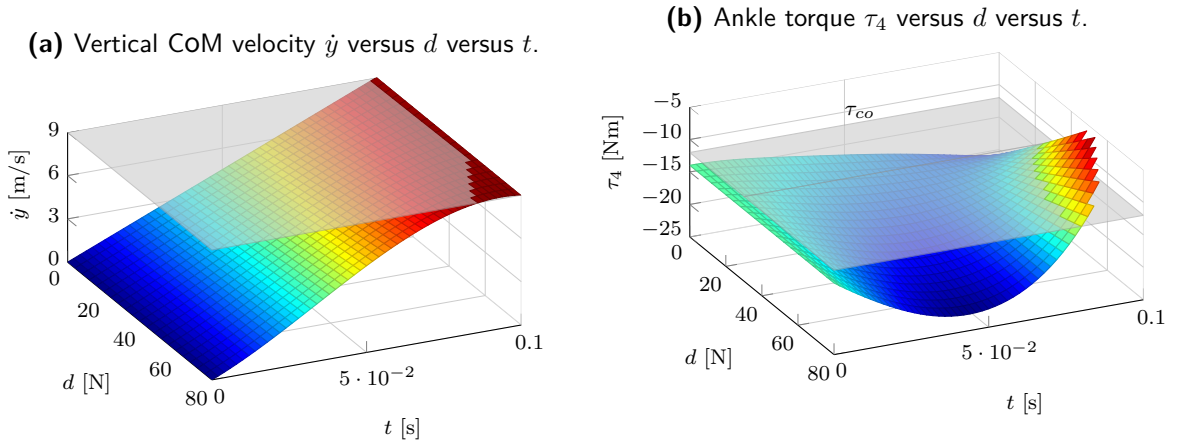


Figure 3-2: Effect of d on velocity and ankle torque. For all values of d , Skippy reaches the end of the thrust phase ($\dot{y} = 9 \text{ m s}^{-1}$) approximately at $t_f : t = 0.1 \text{ s}$. For values $d \gtrsim 30 \text{ N}$, the ankle suggests that it would prematurely unload during the thrust phase, as its calculated torque crosses the cross-over torque τ_{co} .

linearly. A secondary, shifted square root was used and subtracted from the first square root to obtain a nearly linear decent. The function is furthermore defined to start at $f_{\tau 6t}(0) = \tau_{6,0}$:

$$f_{\tau 6t}(q_6) : \tau_6 = u_{d,1} \sqrt{\frac{180}{\pi}} (q_6 - q_{6,0}) - u_{d,2} \left(\sqrt{\frac{180}{\pi}} (q_6 - q_{6,0} + q_{shift}) - \sqrt{\frac{180}{\pi}} q_{shift} \right)$$

The secondary square root is shifted with $\tau_{shift} = \frac{\pi}{6}$ rad, approximately equal to the thrust range of q_6 . $u_{d,1}$ and $u_{d,2}$ are two d -dependent constants which were determined using a least square fit with τ_6 that corresponds to a specific value of d .^[i] Results of the fit function $f_{\tau 6t}(q_6)$ in relation to τ_6 can be seen in figure 3-3a for $d = 50 \text{ N}$.

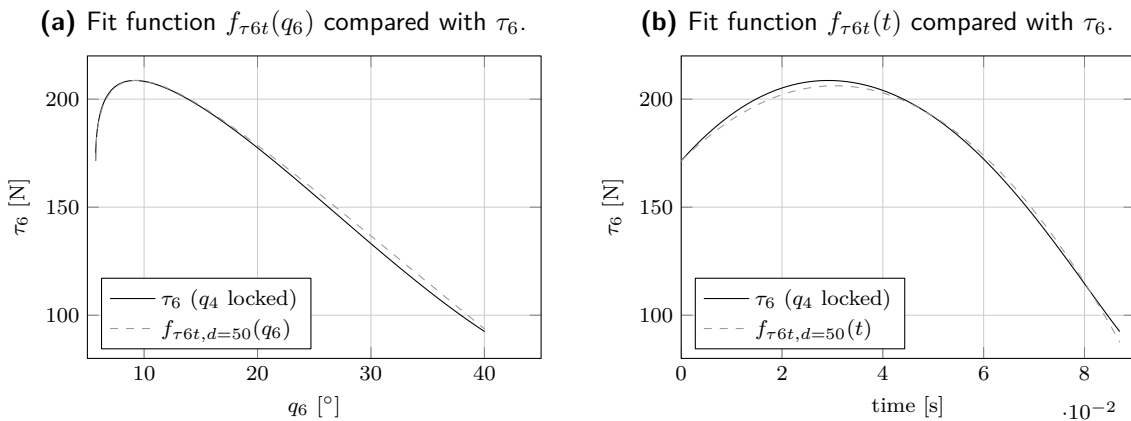


Figure 3-3: Two different fit functions for τ_6 as function of q_6 and t to compare with the desired τ_6 , for $d = 50 \text{ N}$. The fits fit well and meet the boundary conditions.

^[i]We found $u_{d,1} \approx 0.83d - 0.96$ and $u_{d,2} \approx 2.3d + 8.1$.

Similar to $f_y(t)$, τ_6 as function of time took the shape of a 2nd order polynomial. We defined the following fit for $f_{\tau 6t}(t)$, albeit less accurate than $f_{\tau 6t}(q_6)$:

$$f_{\tau 6t}(t) : \tau_6 = \tau_{6,0} + u_1 t + \frac{2d u_2}{c} t - \frac{d u_2}{c^2} t^2 \quad (3-5)$$

in which $c = \frac{0.1}{3}$ s is the same time constant as in equation (3-4). $u_1 \approx -208$ is the slope of the profile for $d = 0$ based on a least square fit with τ_6 . The maximum thrust torque $\tau_{6,\max}$ is nearly linear to d . This observed linearity is embedded in equation (3-5) and the slope of the linear relation is determined by $u_2 \approx 0.83$. Results of the fit function $f_{\tau 6t}(q_6)$ in relation to τ_6 can be seen in figure 3-3b for $d = 50$ N. Both fits are satisfying for the full range of d . However, the fits are unable to completely remove the observed discontinuity between the thrust and steering phase. This problem is tackled in section 3-1-2 by introducing a fuzzy transition between the thrust and steering torque.

3-1-2 Continuous fuzzy transition

This section aims to improve the knee torque profile defined in section 3-1-1 by introducing a fuzzy transition between the thrust and steering torque. This fuzzy transition aims to smooth the torque profile without worsening Skippy's thrust and steering performance.

The parabolic thrust profile discussed in section 3-1-1 cannot fully remove the discontinuity between the thrust and steering phase. The order of continuity would never exceed C^0 *even if* we would be able to exactly match the torque at the end of the thrust and beginning of the steering phase. To smooth the final discontinuity between the thrust and steering phase, a fuzzy transition between both torque profiles is used. The resulting torque profile $f_{\tau 6}(\mathbf{q}, \dot{\mathbf{q}})$ is defined as follows

$$f_{\tau 6}(\mathbf{q}, \dot{\mathbf{q}}) : \tau_6 = f_{\tau 6t}(q_6) v_t(q_6) + f_{\tau 6s}(\mathbf{q}, \dot{\mathbf{q}}) v_s(q_6) \quad (3-6)$$

Or, if $f_{\tau 6t}$ is defined as function of time, we have $f_{\tau 6}(\mathbf{q}, \dot{\mathbf{q}}, t)$:

$$f_{\tau 6}(\mathbf{q}, \dot{\mathbf{q}}, t) : \tau_6 = f_{\tau 6t}(t) v_t(q_6) + f_{\tau 6s}(\mathbf{q}, \dot{\mathbf{q}}) v_s(q_6) \quad (3-7)$$

v_t and v_s are value functions as function of q_6 that define the significance of $f_{\tau 6t}$ and $f_{\tau 6s}$ for the fuzzy transition at $q_{6,f}$. The value functions v_t and v_s are described by continuous sigmoid functions. They are defined as follows, as function of arbitrary variable χ :

$$v_t(\chi) := \frac{1 - \tanh(2s(\chi - \chi_f))}{2} \quad (3-8)$$

$$v_s(\chi) := \frac{1 + \tanh(2s(\chi - \chi_f))}{2} \quad (3-9)$$

The sigmoid function used in equations (3-8) and (3-9) is the hyperbolic tangent (\tanh). The value functions have order of continuity C^∞ and satisfy the following conditions:

$$\begin{aligned}
v_t(\chi) + v_s(\chi) &= 1 \\
0 < \{v_t(\chi), v_s(\chi)\} &< 1 \\
v_t(\chi_f) = v_s(\chi_f) &= 0.5 \\
\lim_{\chi \rightarrow \infty} v_t(\chi) = \lim_{\chi \rightarrow \infty} v_s(-\chi) &= 0 \\
\lim_{\chi \rightarrow \infty} v_t(-\chi) = \lim_{\chi \rightarrow \infty} v_s(\chi) &= 1
\end{aligned}$$

s is the slope of the value function at χ_f and determines the fuzziness of the transition. $s \rightarrow \infty$ corresponds to a non-fuzzy transition and $s = 0$ to a 50-50 mix of both the thrust and steering torque over the full range of χ .

Fuzzier transitions generally lead to profiles with lower derivatives that are more tractable. However, fuzzy transitions could lead to undesired system behaviour, as it could impair Skippy's ability to hop high, to steer (control lift-off momenta) and to maintain grip. This is investigated in section 3-1-3.

3-1-3 Results

We have investigated Skippy's ability to hop high, steer and maintain grip for various values of the parabolic offset force $0 < d < 80$ N, fuzziness slope s and transition angle $q_{6,f}$. Skippy is able to do hops for all values of d , without worsening the performance. It was expected that the ankle would prematurely deflect for values of $d > 30$ N, as the ankle torque crosses its spring's crossover torque τ_{co} during the simulations where the ankle was locked for $d > 30$ N. However, this does not occur, as can be seen in figure 3-4a for $d = 50$ N. Skippy is able to do steering for a large variety of s and $q_{6,f}$ as well. Decreasing s slightly decreases the performance, but Skippy is still able to hop higher than 4 m. A value of $s = 5$ is used in figure 3-4. $q_{6,f}$ is set to $q_{6,f} = 0.8$ rad, but could be lowered to increase steering performance and worsen hopping performance. The resulting torque profile with $d = 50$ N m, $s = 5$ and $q_{6,f} = 0.8$ rad is plotted in figure 3-4d. Momenta are plotted in figure 3-4b, from which can be seen that we are still able to reach our target height and to do steering. Lastly, from figure 3-4c can be seen that the fuzzy transition does not induce any slipping.

With an increasing set of valid torque profiles, we have to find new constraints to figure which of these profiles are really feasible. The series-elastic drivetrain of Skippy is implemented in section 3-2 which will constrain the reproducibility of a desired torque profile.

3-2 Driveline modelling

In section 3-1 we found a feasible set of actuator torque profiles τ_6 with an admissible average power input that allow Skippy to obtain desired lift-off momenta. The set is defined by the ability to vary the parabolic offset force (d), the transition angle between thrust and steering ($q_{6,f}$) and the corresponding fuzziness of this transition (s). Little can yet be said about the capability of the motor to reproduce any of the torque profiles in the set, because an important part of Skippy is still missing in the model. This part is the compliant and highly

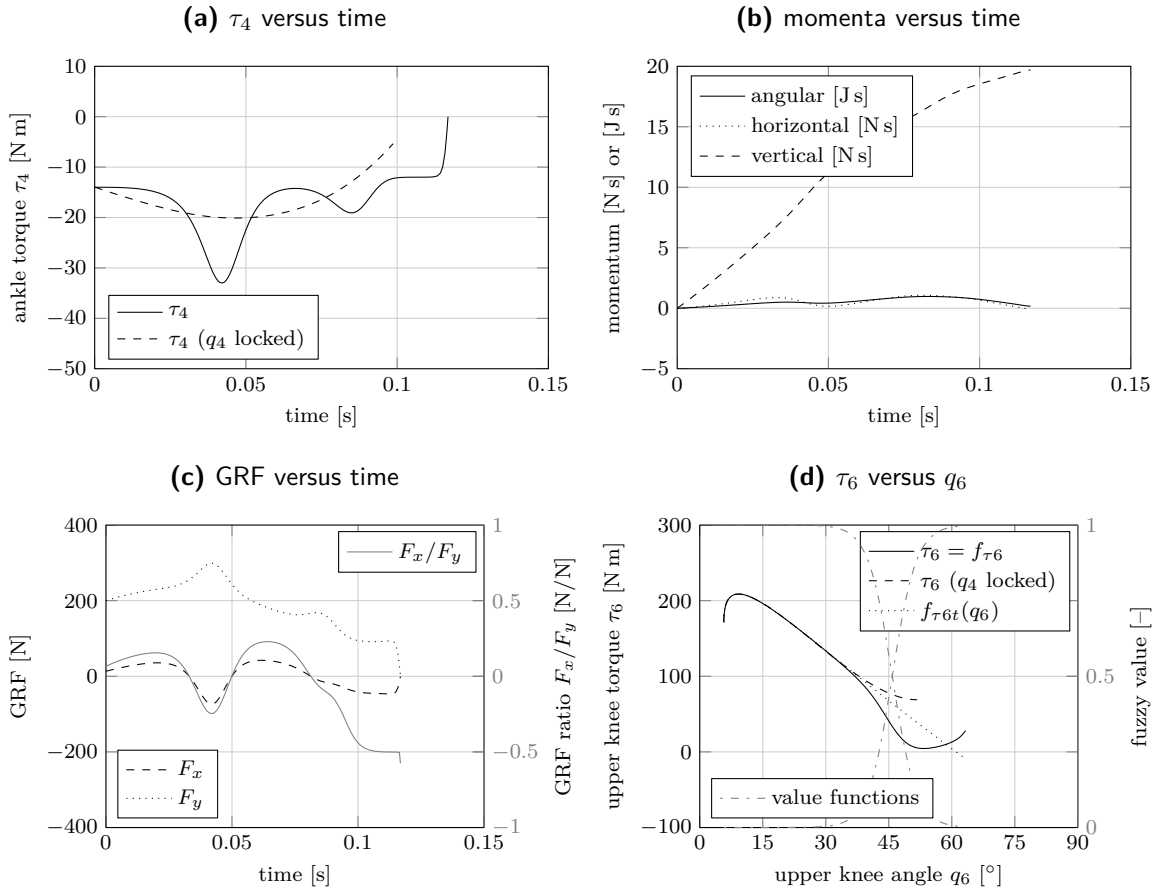


Figure 3-4: Results of Skippy with parabolic and fuzzy thrust profile f_{τ_6} with $d = 50$ N m, $s = 5$ and $q_{6,f} = 0.8$ rad. The more feasible knee torque profile depicted in figure 3-4d still allows Skippy to obtain desired lift-off momenta (figure 3-4b) without slipping (figure 3-4c). Notably, the ankle spring does not prematurely unload (figure 3-4a), in contrast to predictions by figure 3-2b

geared transmission between motor rotor and knee joint, which introduces two additional states and many new system parameters that complicate the model. This section expands the model with linear actuation governed by closed-loop kinematics in section 3-2-1 and the main (energy storage) spring and driveline inertia in section 3-2-2. A PD controller is developed that takes care of the inverse dynamics of the driveline in section 3-2-3. Results are discussed in section 3-2-4.

3-2-1 Linear actuation kinematics

To implement the driveline, two new bodies are introduced, together with fourteen new system parameters. The bodies do not add new degrees of freedom to the system but instead introduce new closed loop kinematics. This section describes the introduction of these bodies in the expanded model, their system parameters and the updated loop closure function to close the kinematic loop.

Model expansion

The model is expanded with two additional joints and bodies. The seventh and eighth body are called motor housing and quadriceps respectively. The seventh joint is a revolute joint connecting the motor housing to the torso. The motor housing holds the motor and spindle of the ball screw. The eighth joint is a prismatic joint connecting the quadriceps to the motor housing. The quadriceps is a body that slides collinear to the motor housing axle and should be connected to the top end of the extended lever. This is done by closing the kinematic loop by adjusting the loop closure function $\gamma(\mathbf{y})$ (equation (2-2)). See figure 3-5 for an overview of the new system.

In the real mechanism, the quadriceps is connected via a spring-loaded prismatic joint to the ball screw's nut, which in turn is connected via a helical joint to the ball screw's spindle, which in turn is connected rigidly to the motor's rotor, which in turn is connected via a revolute joint to the motor's stator and housing. However, it was decided not to model the nut or the spindle and rotor explicitly, and instead simply connect the quadriceps directly to the motor housing via an actuated prismatic joint. This simplification can be justified as follows:

- The nut has negligible inertia, and only a small velocity relative to the torso.
- The spindle and rotor spin very fast, and have a substantial effect on the development of the thrust force. However, this effect can be taken into account in the calculation of the thrust force. For the purpose of calculating the rigid-body dynamics, their inertias can be added to the dynamic model as reflective terms. This is further elucidated in section 3-2-2.^[ii]

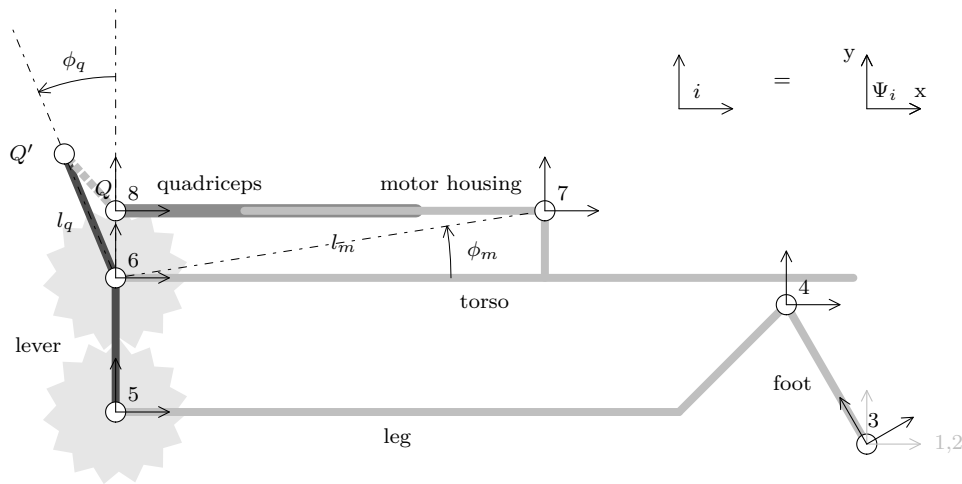


Figure 3-5: Zero configuration ($\mathbf{q} = \mathbf{0}$) of Skippy including linear driveline. Bodies are identified by name, and joints by number. This configuration does not satisfy the loop-closure constraint because the points Q and Q' do not coincide.

^[ii]The rotating spindle and rotor furthermore introduce gyroscopic and inertial accelerations. These effects have to be taken into account in a 3D version of Skippy. This thesis however considers a 2D version of Skippy where out-of-plane forces are assumed to be opposed by constraint forces to keep Skippy in its 2D vertical plane. The swivel bar is used to control Skippy in 3D where we do not have these constraint forces available.

System parameters

The new bodies introduce the following fourteen new system parameters (frame coordinates \mathbf{r} and COM coordinates \mathbf{c} consist of two parameters each):

$$\begin{aligned} \mathbf{r}_7 = \mathbf{M} &= l_m \begin{bmatrix} c \phi_m & s \phi_m \end{bmatrix} && \text{motor housing axle coordinates in } \Psi_6 \\ \mathbf{r}'_8 = \mathbf{Q}' &= l_q \begin{bmatrix} -s \phi_q & c \phi_q \end{bmatrix} && \text{quadriceps axle coordinates in } \Psi'_5 \\ & & \mathbf{r}_8 && \text{floating quadriceps axle coordinates in } \Psi_7 \\ & & m_7, \mathbf{c}_7, I_7 && \text{mass, CoM coordinates and rotational inertia of the motor housing} \\ & & m_8, \mathbf{c}_8, I_8 && \text{mass, CoM coordinates and rotational inertia of the quadriceps} \end{aligned}$$

Ψ'_5 is a coordinate frame embedded in the lever (identical orientation as Ψ_5) but located at the upper knee joint (at Ψ_6) rather than the lower knee joint (at Ψ_5). Q is a point at the origin of Ψ_8 . Q' coincides with the tip of the extended lever defined by \mathbf{r}'_8 . The real system has a revolute joint between the quadriceps and lever that is located at Q . The loop closure constraint is satisfied when Q coincides with Q' .

All inertias of the motor housing and quadriceps (four parameters) are set to zero and the inertias of already existing bodies are unchanged^[iii]. The coordinates \mathbf{c}_7 and \mathbf{c}_8 of the two COMs (four parameters) are thus meaningless as well. The elimination of these eight parameters simplifies our system. It also implies that the dynamic capabilities of this system should be exactly the same as that of the previous one, which allows us to make more valid comparisons between results of new and previous simulations.

Both the motor housing and quadriceps barely rotate with respect to the torso (i.e. $\ddot{q}_7 \approx 0$), so their rotational inertia has little contribution. The quadriceps and spring are furthermore very light, for which we also neglect their mass. The motor housing however is not light as it houses a heavy motor, ball screw spindle, coupling and bearings. Nevertheless, its movement with respect to the torso is negligible, so we have left its mass included in the torso's mass. The torso's orientation is defined in such a way that the torso—modelled as a line—runs through its own COM. The position of the motor housing axle is defined to have a small positive offset $M_y = 5$ cm from this line, because the mass of the swivel motor and bar—which will be mounted below the main motor—will lower the COM of the torso.

The horizontal motor housing axle coordinate is set at $M_x = 32$ cm. This implies that the axle is located towards the end of Skippy's torso, approximately 32 cm away from the knee upper joint. The axle is placed towards the end of the torso for two reasons. Firstly, we require space for the spindle and main spring between the motor housing and lever. Secondly, more mass at the end of the torso is desirable to make high jumps. However, we cannot place the motor housing at the very end of the torso because the motor and swivel bar also require space (estimated 20 to 25 cm) behind the motor housing. The exact positioning of the motor housing axle is not important because it will barely affect the kinematics as long as it is not too close to the knee joint. As such, we will not be further concerned with tuning these parameters in this chapter.

The value of \mathbf{r}_8 is trivial. The quadriceps is modelled as a horizontal prismatic joint collinear to the motor housing axle, implying that $r_{8,y} = 0$ and that we have a DOF in the direction of

^[iii]Recall that the inertia of the torso is supposed to include the inertias of the motor and drive components

$r_{8,x}$. For aesthetic reasons, a horizontal offset $r_{8,x} = -M_x = -r_{7,x}$ is defined to bring its zero position closer to the top end of the lever to which it should be connected by the kinematic loop closure function $\gamma(\mathbf{y})$. The distance between the motor housing axle and quadriceps axle is the driveline length $x_{in} = M_x - q_8$.

This leaves only two parameters, l_q and ϕ_q (or $r'_{8,x}$ and $r'_{8,y}$), which are considered most important as they significantly affect the transmission ratio between the linear driveline and the knee. The lever length l_q is almost directly proportional to the transmission ratio, and is initially set $l_q = 10$ cm based on the primary calculations in section 2-1-3. For an effective lever length of $l_{q,e} = 7$ cm, a knee torque of $\tau_6 = 200$ N m would require a quadriceps force (or nut force) of approximately $\tau_8 = 2.9$ kN, close to the maximum permissible absolute force on the ball screw. The effective lever length is defined as the perpendicular (i.e. shortest) distance between the screw rod axis and the knee upper joint (\mathbf{X}_6)^[iv]. $l_{q,e}$ is a function of the knee rotation and is maximally equal to l_q when the lever is perpendicular to the screw rod axis. ϕ_q influences the effective lever length as it is able to shift this function. Initially, we choose $\phi_q = \frac{\pi}{8}$ rad, which implies that the effective lever length is maximum approximately halfway through the thrust phase, where it is able to transmit the highest torque to the knee joint. The effective lever length becomes lower towards the end of the thrust phase and during the steering phase, which allows for more speed (but less torque).

Loop closure

Joint positions q_5 , q_6 , q_7 and q_8 are all dependent on each other. Since we want to use the quadriceps force τ_8 as input force from now on, we redefine our independent position variables \mathbf{y} to exclude q_6 and include q_8 :

$$\mathbf{y} = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_8]^\top$$

The loop closure function $\mathbf{q} = \gamma(\mathbf{y})$ is rewritten accordingly. The closed-loop kinematics are still implemented explicitly and obtained through analytical calculations, which has the advantage of quicker calculations (smaller system matrices to be integrated and inverted), better accuracy and more robustness than implementing closed-loop kinematics implicitly, as long as we do not cross any singularity. The mechanism is designed not to cross any singularity in its useful ROM ($0^\circ \leq \phi_{knee} \leq 180^\circ$), because this would have devastating consequences for the torque transferred through the knee joint. We have simplified the derivation of the kinematic equations by making use of predefined loop closure functions that calculate the kinematics of triangular mechanisms with sides a, b and c and corresponding opposite angles α, β, γ , provided by R. Featherstone:

$$\{\alpha, \beta, \gamma, \mathbf{G}, \mathbf{g}\} = \Gamma_\Delta(a, b, c, \dot{a}) \quad (3-10)$$

$$\{a, \beta, \gamma, \mathbf{G}, \mathbf{g}\} = \Gamma_\Delta(\alpha, b, c, \dot{\alpha}) \quad (3-11)$$

The function either takes a variable side length a or variable angle α as independent position variable, its time derivative and the lengths of the two *static* sides (b and c) as input. \mathbf{G} and \mathbf{g} are a $[3 \times 1]$ Jacobian matrix and $[3 \times 1]$ convective acceleration vector, respectively, used

^[iv]An effective lever length of $l_{q,e} = 7$ cm for a lever with $l_q = 10$ cm corresponds to an angle between the lever and spindle of $\arcsin(7/10) \approx \frac{\pi}{4}$ rad

to calculate the output derivatives.^[v] These functions do all the trigonometric calculations, which explains why one cannot find any (arc)sine or (arc)cosine in this section.

To derive the kinematics of this version of Skippy we require only one triangle loop closure function. For the derivation of the kinematics of Skippy with a real four-bar linkage (rather than a 1:1 gear ratio between q_5 and q_6), we require the usage of three triangle functions. The first triangle (denoted Δ) is identical for the version of Skippy with and without the full four-bar linkage. The triangle is defined by the variable length $x_{in} = M_x - q_8$ with $\dot{x}_{in} = -\dot{q}_8$ and static lengths l_q and l_m . The latter is the distance between the motor housing axle and upper knee joint. We compute

$$\{\phi_{\Delta in}, \phi_{\Delta q}, \phi_{\Delta m}, \mathbf{G}_{\Delta}, \mathbf{g}_{\Delta}\} = \Gamma_{\Delta}(x_{in}, l_q, l_m, \dot{x}_{in})$$

From which we can find the dependent positions variables

$$q_7 = \phi_m - \phi_{\Delta q} \quad (3-12)$$

$$q_6 = \pi/2 - \phi_{\Delta in} - \phi_m + \phi_q \quad (3-13)$$

Moreover, we still have $q_5 = q_6$ because of the 1:1 gear ratio. Their derivatives—together with \mathbf{G} (size $[8 \times 5]$) and \mathbf{g} (size $[8 \times 1]$)—can be obtained by making use of \mathbf{G}_{Δ} , \mathbf{g}_{Δ} and equations (B-7) to (B-10).

Now that we have $\dot{\mathbf{q}} = \mathbf{G}\dot{\mathbf{y}}$, we can find the transmission ratio between q_6 and q_8 . Since q_6 is only dependent on q_8 , we only have one term in \mathbf{G} that defines this transmission ratio:

$$\dot{q}_6 = G_{\Delta 1} \dot{q}_8$$

where $G_{\Delta 1}$ (first element of \mathbf{G}_{Δ}) is equal to the inverse of the transmission ratio R_2 , which we define as follows:

$$R_2 := \frac{\partial q_8}{\partial q_6} \quad (3-14)$$

R_2 is plotted in figure 3-6. If we assume a frictionless system with zero power loss in the joints, we have that $\dot{q}_6 \tau_6 = \dot{q}_8 \tau_8$ and thus

$$\tau_8 = G_{\Delta 1} \tau_6 = \frac{\tau_6}{R_2} \quad (3-15)$$

Equation (3-15) allows us to directly compute the desired quadriceps force profile τ_8 from the thrust torque profiles defined for τ_6 in section 3-1:

$$f_{\tau_8}(\mathbf{q}, \dot{\mathbf{q}}(t)) : \tau_8 = G_{\Delta 1}(q_8) f_{\tau_6}(\mathbf{q}, \dot{\mathbf{q}}(t)) \quad (3-16)$$

This new force profile should produce *exactly* the same results as those presented in section 3-1-3, because the inertial properties have not been changed since the introduction of the two new massless bodies (the motor housing and quadriceps). This fact has been used as verification. The newly obtained nut thrust profile is plotted in figure 3-7, which indeed leads to exactly the same dynamic behaviour as previously observed.

^[v]See equations (B-7) to (B-10), where $\mathbf{y} = a$ and $\mathbf{q} = [\alpha \beta \gamma]^T$ for equation (3-10) and $\mathbf{y} = \alpha$ and $\mathbf{q} = [a \beta \gamma]^T$ for equation (3-11)

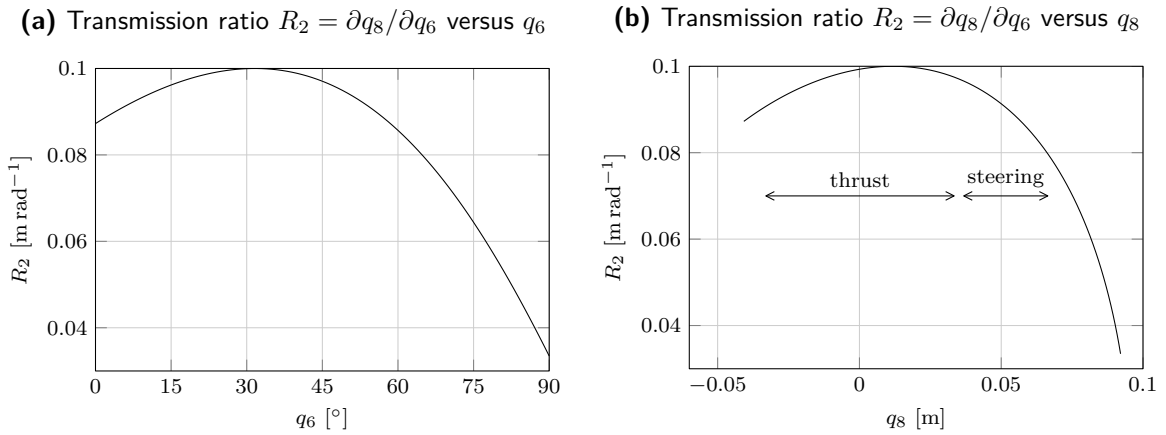


Figure 3-6: Lever transmission ratio R_2 as function of q_6 and q_8 . The transmission ratio is approximately constant during thrust and decreases during steering.

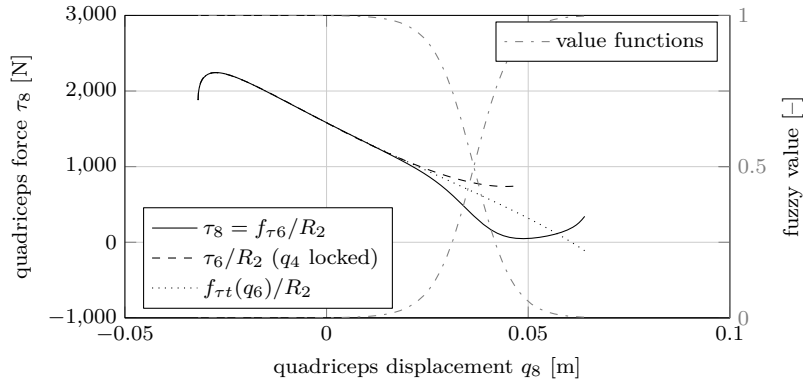


Figure 3-7: τ_8 for identical conditions as in figure 3-4d with identical dynamic system behaviour. τ_8 is the result from a kinematic mapping and is proportional to the torque the motor has to deliver.

3-2-2 Spring and driveline inertia

Having defined the linear driveline in section 3-2-1, we have yet to add a spring and ball screw model including inertia, which together with the electric motor form the series elastic actuation (SEA). The SEA introduces a new body, which is the nut of the ball screw. This section describes the implementation of the SEA and explains how the initial conditions of the nut are determined.

The nut

Implementation of the SEA requires the introduction of an additional body. This body is the ball screw's nut. As mentioned in section 3-2-1, the nut is not modelled as an additional rigid body because it is relatively light and slow. The shaft—consisting mainly of the motor rotor, coupling and spindle—is directly geared to the nut and is also not modelled as an additional body as its velocities with respect to the torso are negligible. It furthermore requires Skippy

to be modelled as a 3D system, which blows up the model complexity. Instead, we model the reflected mass of the shaft at a massless nut, connected to the massless main spring, connected to the quadriceps axle.

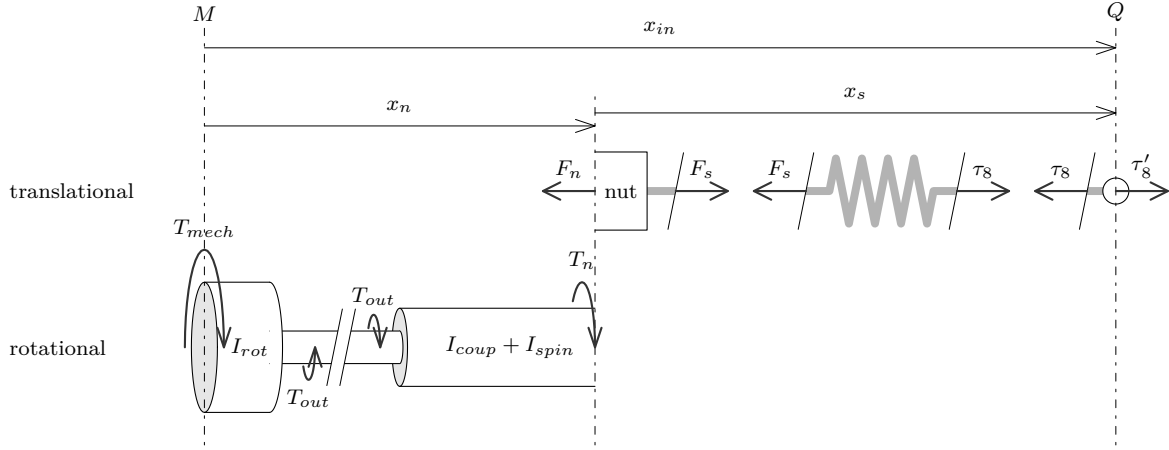


Figure 3-8: 1D driveline model of Skippy: from motor rotor to quadriceps axle. Forces, torques and displacements are drawn positive.

Driveline inertia

Consider the 1D driveline between the motor rotor (close to the motor housing axle) and quadriceps axle in figure 3-8. The frictionless component of the generated electromotive torque T_{mech} —also referred to as mechanical motor torque—acts via the motor rotor with rotational inertia of I_{rot} on the motor shaft with torque T_{out} . T_{mech} differs from the electromotive force (emf) T_{emf} in that the terminal friction torque is already subtracted. The shaft is connected via the coupling with inertia I_{coup} to the spindle with inertia I_{spin} . The spindle connects to the nut where rotational motion is transferred to linear motion and thus torque T_n to nut force F_n . We have

$$\left. \begin{aligned} I_{rot} \ddot{\phi}_h &= T_{mech} - T_{out} \\ (I_{coup} + I_{spin}) \ddot{\phi}_h &= T_{out} + T_n \end{aligned} \right\} I_h \ddot{\phi}_h = T_{mech} + T_n \quad (3-17)$$

I_h is the total shaft inertia $I_h = I_{coup} + I_{spin} + I_{rot}$ and $\ddot{\phi}_h$ the entire shaft's angular acceleration. Note that in equation (3-17), T_n is defined to have the same sign as T_{out} on the spindle, which corresponds to the positive directions drawn in figure 3-8.^[vi]

For a non-accelerating system, all torques in the driveline are equal to each other. However, they could start to deviate significantly when the system accelerates. T_n is proportional to τ_8 and T_{mech} is the actual input torque directly related to the mechanical power that the motor delivers: $P_{mech} = T_{mech} \dot{\phi}_h$. The motor shaft torque T_{out} is furthermore taken apart, as its maximum absolute is interesting for the mechanical design of Skippy, e.g. for coupling selection. T_{mech} is *not* necessarily similar to the torque transmitted through the shaft and is

^[vi]Note that this is not conventional, but it was decided to do so for various practical reasons that are related to the fact that the direction of positive x_{in} and q_8 are also opposite.

thence not interesting for the mechanical design. Since I_{rot} is by far the largest of all three shaft inertias, T_{out} is going to be more similar to T_n than T_{mech} .

The transmission ratio R_3 between the shaft's angular velocity $\dot{\phi}_h$ and nut's linear velocity \dot{x}_n is

$$R_3 = \frac{\dot{\phi}_h}{\dot{x}_n} = -\frac{2\pi}{n} \quad (3-18)$$

where $n = 2$ mm is the lead of the ball screw. We have $\dot{\phi}_h = R_3\dot{x}_n$ and $F_n = R_3T_n$. We can rewrite equation (3-17) to

$$R_3^2 I_h \ddot{x}_n = F_n + R_3 T_{mech} \quad (3-19)$$

$$= \tilde{m}_h \ddot{x}_n = F_n + \tilde{F}_{mech} = F_d \quad (3-20)$$

where $\tilde{m}_h = R_3^2 I_h$ is the reflected mass of the shaft on the nut. It is not a *real* mass, but it is experienced by the nut as inertia to be accelerated. Similarly, $\tilde{F}_{mech} = R_3 T_{mech}$ is not a real force acting on the shaft. We use a halo ($\tilde{}$) to distinguish reflected terms from real terms. F_d is the sum of $F_n = F_s$ and \tilde{F}_{mech} . The spring force F_s pulls the nut towards the quadriceps axle, and the mechanical motor force \tilde{F}_{mech} pushes the nut towards that axle. As such, their sum F_d is effectively the force that accelerates the nut with mass \tilde{m}_h towards the quadriceps axle.

Inertias are taken from real components:

| | |
|---------------------------------|---|
| $I_{coup} = 5.8 \text{ g cm}^2$ | coupling CPCX19-6-6 (Misumi, 2015) |
| $I_{spin} = 7.8 \text{ g cm}^2$ | ball screw 2412 M8 n2 (Steinmeyer, 2015) |
| $I_{rot} = 79.2 \text{ g cm}^2$ | rotor of motor RE35 90 W 24 V (Maxon Motor, 2014) |

Their sum is $I_h = 92.8 \text{ g cm}^2$, which corresponds to a reflected mass of $\tilde{m}_h = 91.6 \text{ kg}$. The rotor inertia I_{rot} is most prominent. The coupling is relatively small and it is selected on (minimal) weight. The coupling is rated to transmit a continuous torque of 1.0 N m. This value corresponds to the maximum permissible absolute torque on the nut, calculated from the maximum permissible absolute axial force on the ball screw of $F_{n,max} = 3.1 \text{ kN}$ ($T_n = F_n/R_3 \approx 1.0 \text{ N m}$). Moreover, the stall torque of the motor at the nominal voltage is $1.2 \text{ N m} > 1.0 \text{ N m}$, which is a torque the motor shaft is *at least* able to transmit. The maximum transmissible torque is likely even higher, although not specified by Maxon. Stepping up to a higher coupling can be done if this appears to be necessary from dynamic simulations. This would slightly increase the weight and rotational inertia of the driveline.

Main spring

The nut connects to the main spring with force F_s and the spring to the quadriceps axle:

$$F_n = F_s = \tau_8 \quad (3-21)$$

The spring force F_s is only a function of its elongation x_s , if we assume that it is purely elastic. The elongation of the spring is

$$x_s = x_{in} - x_n = M_x - q_8 - x_n \quad (3-22)$$

For the spring profile $f_s(x_s)$ we initially assume a linear profile $f_{s,lin}(x_s)$ with rest length l_z and stiffness k :

$$f_{s,lin}(x_s) : F_s = k(x_s - l_z) \quad (3-23)$$

We also define a simple regressive spring profile, since we eventually have a desire for one (section 2-1-3), where the spring stiffness is low during the thrust phase and high during balancing and the steering phase. The simplest regressive profile is arguably a bilinear spring profile:

$$f_{s,bilin}(x_s) : F_s = \begin{cases} k_1(x_s - l_z) & x_s \leq l_z + l_{bi} \\ k_2(x_s - l_{bi} - l_z) + k_1 l_{bi} & x_s > l_z + l_{bi} \end{cases} \quad (3-24)$$

in which k_1 is the first spring constant and k_2 the second spring constant, with a transition between the two at $x_s = l_z + l_{bi}$. For a regressive spring profile, we have that $k_2 < k_1$. Note furthermore that equation (3-24) has continuity C^0 , so we can only differentiate it once. Combining equations (3-20) to (3-22) and Skippy itself, we obtain the diagram depicted in figure 3-9.

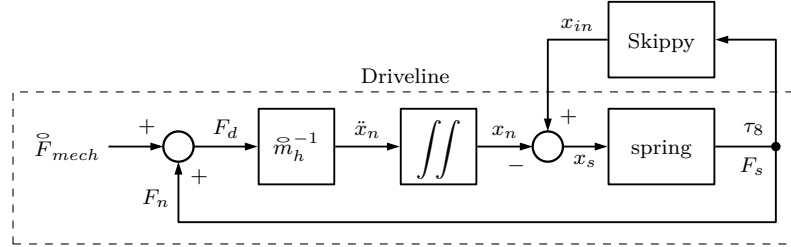


Figure 3-9: Block diagram of driveline in Skippy.

Initial conditions

The two new states require two new initial conditions: the initial nut position $x_{n,0}$ and velocity $\dot{x}_{n,0}$. The initial conditions are chosen such that the initial desired thrust force $f_{\tau 8t,0}$ and its time derivative $\dot{f}_{\tau 8t,0}$ correspond to $\tau_{8,0}$ and $\dot{\tau}_{8,0}$. For this, we have to compute the initial spring elongation $x_{s,0}$ and initial driveline length $x_{in,0}$

$$\left. \begin{aligned} x_{s,0} &= f_s^{-1} = f_s^{-1}(f_{\tau 8t,0}) \\ x_{in,0} &= M_x - q_{8,0} \\ x_{n,0} &= x_{in,0} - x_{s,0} \end{aligned} \right\} x_{n,0} = M_x - q_{8,0} - f_s^{-1}(f_{\tau 8t,0}) \quad (3-25)$$

with $x_s = f_s^{-1}(F_s)$ the inverse spring function. For the nut velocity we calculate:

$$\left. \begin{aligned} \dot{x}_{s,0} &= \frac{\partial x_s}{\partial f_{s,0}} \dot{f}_{s,0} = \frac{\partial x_s}{\partial f_s(x_{s,0})} \dot{f}_{\tau 8t,0} \\ \dot{x}_{in,0} &= -\dot{q}_{8,0} \\ \dot{x}_{n,0} &= \dot{x}_{in,0} - \dot{x}_{s,0} \end{aligned} \right\} \dot{x}_{n,0} = -\dot{q}_{8,0} - \frac{\partial x_s}{\partial f_s(x_{s,0})} \dot{f}_{\tau 8t,0} \quad (3-26)$$

The initial nut velocity calculated in equation (3-26) assumes zero initial shaft acceleration $\ddot{\phi}_h = 0$, which is not required, but a valid and reasonable assumption.

With the initial nut position defined by equation (3-25), the spring's rest length l_z in equations (3-23) and (3-24) has no physical importance (it would only shift $x_{in,0}$ and can hence be used for aesthetic reasons). The stiffness k of a linear spring is predetermined if the initial spring force and initial potential energy storage are known. The required stiffness for a linear spring is then $k_{lin} : k = F_{s,0}^2 / (2E_{s,0}) = 32.15 \text{ kN m}^{-1}$, given an initial amount of available spring energy $E_{s,0} = 55 \text{ J}$ and force $F_{s,0} = \tau_{8,0} \approx 1.9 \text{ kN}$. The spring is going to have a weakening profile eventually. As such, for the steering profile we will have a spring stiffness $k > k_{lin}$ and for the thrust profile $k < k_{lin}$. Correspondingly, for the bilinear spring profile we have $k_1 > k_{lin}$ and $k_2 < k_{lin}$, but their values are not determined.

$F_{s,0}$ is a function of the desired average and initial vertical thrust force $F_{y,0} = 200 \text{ N}$ and selected system parameters, e.g. lever length. If one of these variables is changed, the maximum spring stiffness of the thrust phase k_{lin} changes as well. For instance, a shorter lever length requires a higher spring stiffness.

$E_{s,0}$ is based on an estimate of the amount of energy that we have stored in the spring at $t = 0$. This is all the energy pumped in the spring during the landing phase. This amount of energy is roughly equal to

$$E_{s,0} \approx E_{pre} + W_{mech,lp} + \Delta E_{p,g} - E_{p,a,0} \quad (3-27)$$

ignoring heat losses and rotational kinetic energy. $E_{pre} = 40 \text{ J}$ is the amount of available energy from the previous hop, see sections 2-1-2 and 2-1-5. $W_{mech,lp}$ is the amount of work the motor is expected to do during the landing phase. This is estimated at $W_{mech,lp} = 15 \text{ J}$, which is approximately a third of the total work the motor is required to do during the stance phase, and thus half the work the motor is required to do during the thrust phase. $\Delta E_{p,g} = mg\Delta y = 2 \times 9.81 \times 0.6 = 11.8 \text{ J}$ is the potential energy built during the stroke, where Skippy's CoM travels approximately $\Delta y = 0.6 \text{ m}$ during the launch phase. $E_{p,a,0} = 10.6 \text{ J}$ is the energy initially stored in the ankle spring, which is compressed, obtained by integrating equation (2-3) over q_4 .

3-2-3 Inverse dynamics control

With the series elastic driveline defined in section 3-2-2, the quadriceps force τ_8 is no longer the input force. Instead, the mechanical motor torque T_{mech} (or F_{mech}) is. This section describes how the motor torque is calculated to approximate a desired quadriceps force τ_8 , such as the quadriceps force depicted in figure 3-7.

Having defined the new compliant driveline and its initial conditions, one could firstly try to set the new input to $T_{mech} = -f_{\tau_8}$ defined in equation (3-16). Initially this works out well due to the initial nut conditions defined in section 3-2-2. However, τ_8 starts to diverge significantly from f_{τ_8} because of the newly introduced inertia and compliance in the driveline. The overall performance of Skippy deteriorates. Inverse dynamics calculations are required, which calculate the required input torque T_{mech} to make τ_8 track the desired f_{τ_8} . Directly computing T_{mech} from f_{τ_8} through the 1D driveline would require double differentiation, which could make T_{mech} unnecessarily aggressive and noisy. Instead, a simple controller is developed that makes τ_8 track f_{τ_8} approximately.

We know from basic control theory that a simple mass-spring system depicted in figure 3-9 can be controlled with a simple PD controller (Franklin et al., 2010), assuming that the

external input $x_{in} = M_x - q_8$ fed back from Skippy's main system is unable to destabilize the system. The integration of the PD controller in the system is depicted in figure 3-10.

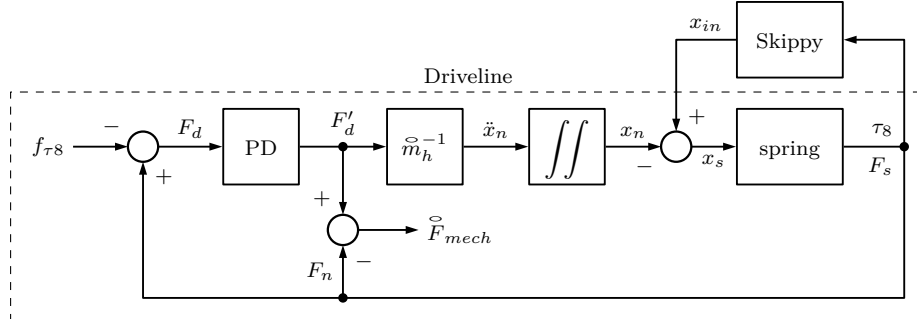


Figure 3-10: Block diagram of driveline with simple PD control in Skippy.

The derivative of the input for the PD is derived analytically to minimize the sensitivity for noise. We have $\dot{F}'_d = \dot{F}'_n - \dot{f}_{\tau 8}$. The time derivative of the nut force $\dot{F}'_n = \dot{F}'_s$ is obtained as follows:

$$\dot{f}'_s(x_s, \dot{x}_s) = \frac{\partial f_s(x_s)}{\partial x_s} \dot{x}_s$$

Calculating the time derivative of the desired nut force (or quadriceps force) profile $f_{\tau 8}$ in equation (3-16) analytically is a tedious task because of the complicated steering strategy calculations incorporated in equations (3-6) and (3-7). Irrespective of what exactly we want to do during steering, τ_8 has to drop significantly and will be around zero. For this reason, we simplify our input profile by temporarily setting the steering profile to either zero or up to a 2nd degree polynomial around zero, as function of q_6 : $f_{\tau 6s}(q_6)$. We furthermore use the desired thrust torque profile as function of time $f_{\tau 6t}(t)$ for its differentiability. Equation (3-16) becomes:

$$\begin{aligned} f_{\tau 8}(q_6, q_8, t) : \tau_8 &= G_{\Delta 1}(q_8) f_{\tau 6}(q_6, t) \\ &= G_{\Delta 1}(q_8) (f_{\tau 6t}(t) v_t(q_6) + f_{\tau 6s}(q_6) v_s(q_6)) \end{aligned} \quad (3-28)$$

Differentiating equation (3-28) to obtain $\dot{f}_{\tau 8}$ requires application of the product and chain rule for differentiation and leads to a series of terms that we all have available.

The controller is tuned while being offline. This means that its output τ_8 is not fed back to Skippy and that its external input q_8 is only a function of time, taken from the previous simulation where $\tau_8 = f_{\tau 8}$. The controller gains are tuned for multiple spring stiffnesses and made stiffness-dependent with the aim of making it work for non-linear springs. The controller is brought online after it is tuned. The corresponding motor torque $T_{mech} = \hat{F}_{mech}$ was then calculated backwards from the output of the PD-controller F'_d : $\hat{F}_{mech} = F'_n - F'_d$, which is also depicted in figure 3-10.

3-2-4 Results

The simulation including closed-loop driveline dynamics and PD controller is run for various profile parameters d , s and $q_{6,f}$ and spring stiffnesses. The controller is able to generate a

force τ_8 that follows the desired f_{τ_8} well for the full set of profiles and a large range of spring stiffnesses, as can be seen in figure 3-11, which shows the results for $d = 50 \text{ N}$ and $s = 5$, $q_{6,f} = 0.8 \text{ rad}$ and a linear spring with $k_{lin} : k = 32.15 \text{ N m}^{-1}$. However, we also observe that T_{mech} is going in a frenzy of sways in order to generate the desired torque profile. A change of the thrust profile and spring parameters have a major effect on T_{mech} , from which we can conclude that the input is sensitive to the desired output. Some sets of parameters could lead to reasonable results up to the steering phase, such as those depicted in figure 3-11. The thrust profile chosen in figure 3-11 is not aggressive: d is low, the steering is simplified to a constant (zero torque) and the transition between thrust and steering is very fuzzy. Yet, the system is exceeding most of its capabilities. These include the following observations, among other exceedances:

- The nut exceeds velocities that are more than an order of magnitude higher than the maximum permissible absolute $\dot{x}_{n,max} = 0.2 \text{ m s}^{-1}$ and the no-load speed of the motor.
- The axial force in the spindle exceeds the maximum permissible absolute $F_{n,max} = 3.1 \text{ kN}$.
- The motor is required to deliver more mechanical power $P_{mech} = \bar{F}_{mech} \dot{x}_n$ than the approximately 320 W it is capable of delivering at optimal rotational velocity, which is at approximately $\dot{x}_n = 14.3 \text{ cm s}^{-1}$.

The PD-controller was partially introduced to have control over the aggressiveness on the computation of T_{mech} . As such, several attempts were done to lower the controller gains to make T_{mech} less aggressive, but this also made f_{τ_8} insufficiently accurately tracked.

In summary, if the motor is supposed to strictly reproduce a master curve (f_{τ_8}), then its calculated torque will be too sensitive to a change of that master curve. If the aggressiveness of the controller is reduced so as to purge the extreme sensitivity, the reproducibility of the master curve is so defective that there appears no point in defining a master curve in the first place.

The reason for high sensitivity of T_{mech} to changes of f_{τ_8} is primarily attributed to the low spring stiffness and secondarily to high reflected mass $\bar{m}_h = 92 \text{ kg}$, which owes its value mainly to the RE35 90 W 24 V Maxon motor mentioned in section 2-1-3. Changing the inertia of the motor to that of a lighter Maxon motor (RE30 60 W 24 V)—reducing the inertia nearly by a factor of 2—did improve the results, but not sufficiently so. The low spring stiffness is especially a problem during the steering phase, during which it should be higher. Increasing the spring stiffness during the steering reduces sensitivity, but it brings the nut velocity closer to the high joint velocity \dot{q}_8 (also plotted in figure 3-11), which nevertheless forces the nut to travel at velocities that exceed its maximum velocity by far. It is cumbersome to find a set of parameters for the launch profile to be viable due to all system limitations and the huge sensitivity of the motor torque command on changes of the desired knee torque profile.

Concluding remarks

This chapter works to a more realistic model of Skippy that includes its series elastic drive-line, which imposes limitations on the knee torque command profile. Towards a more realistic design, pre-defined knee torque profiles are devised that are able to comply with a series

elastic driveline and permit the actuator to do positive work throughout most of the stance phase, without impairing Skippy's ability to jump and steer. These torque profiles take the shape of parabolae and have a continuous fuzzy transition between thrust and steering phase. The implementation of the series elastic driveline, however, has shown that the reproducibility of these torques is nonetheless disappointing. The calculated motor torques and shaft velocities—which were calculated correctly by a successful and simple PD controller responsible for inverse dynamics—turned out to be too sensitive (due to the relatively low spring stiffness) and violated many of the physical limitations (e.g. limited nut velocity). It is concluded that optimizing for the sensitive motor torque command T_{mech} is an inefficient approach.

A different strategy is sought in chapter 4, where we initially impose a feedforward torque on T_{mech} and then proceed to apply constraints to this torque to cope with all the constraints of Skippy.

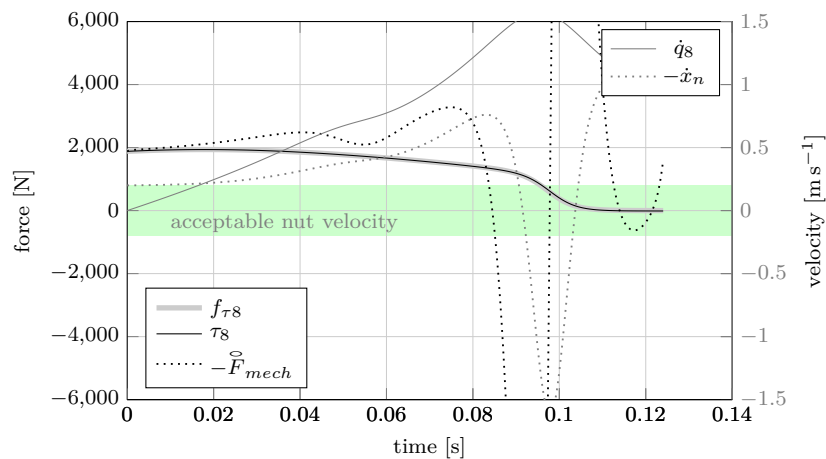


Figure 3-11: Results of Skippy including series-elastic driveline. The controller works well, as it reproduces the torque command $f_{\tau 8}$ at τ_8 . However, the required input torque T_{mech} is too sensitive to the required profile and not reproducible in reality. The nut velocity is also exceeding its maximum permissible velocity by far.

Feedforward approach

In chapter 3 it was found that working towards an optimization from a set of usable thrust profiles is inefficacious. This chapter uses a feedforward approach to find a viable thrust strategy, using the model of Skippy defined in section 3-2 including driveline dynamics depicted in figure 3-9. In section 4-1 we start by feeding a simple force profile directly to \hat{F}_{mech} . In sections 4-2 and 4-3 we cope with encountered problems by applying (electrical and velocity) saturations that take physical limitations of the system into account. In section 4-4 the model is expanded with ball screw friction, and an analysis on energy flow is introduced. The saturations and energy analysis help us in understanding the limitations and possibilities of Skippy, which are subsequently used in section 4-5 to analyse Skippy's ability to do steering. Each of the saturations and the requirement to do steering result in a decrease of Skippy's jumping performance. Finally, section 4-6 focusses on (energy-based) system improvements that result in an improvement of Skippy's jumping and steering performance.

4-1 Applying rotor torques

This section describes the exercise of feeding a simple mechanical motor torque command directly to the motor rotor, and analyses the thrust performance and limitations of doing so.

A constant motor torque is applied throughout the entire thrust phase with the purpose of observing Skippy's behaviour. The primary focus is the thrust phase. The input force command $f_{mech,c} : \hat{F}_{mech}$ lasts till $t = 0.1$ s after which it drops to zero, which is when we would approximately enter the steering phase. Essentially, we are redoing what we have initially tried in section 3-2-3 but with a simpler profile than $f_{\tau 8}$ defined in section 3-1 and more importantly: without the aim of reproducing that profile at τ_8 .

The initial conditions \mathbf{q}_0 and $\dot{\mathbf{q}}_0$ are still identical to the ones used in section 2-2 and the nut position is still calculated according to equation (3-25), corresponding to an initial thrust of $F_y = 200$ N. However, the nut velocity is initially set to its maximum, $x_n = -0.2$ m s⁻¹, and not calculated according to equation (3-26), where it was calculated to obtain a desired time derivative of the initial desired thrust force. The nut velocity calculated in equation (3-26)

is extremely sensitive for the initial desired thrust force rate due to the relatively low spring stiffness, whereas it is unknown what the ideal initial thrust force rate is (as we found many working force rates in section 3-1-3).

With a primary focus on the thrust phase, the initial simulations use a linear spring. The spring is considered floating, i.e. we are not concerned with the initial amount of energy $E_{s,0}$ stored in the spring. This allows us to experiment with different spring stiffnesses $k \leq k_{lin} \simeq 32.15 \text{ kN m}^{-1}$, see section 3-2-2, with an initial estimate $k = k_{lin}$.

Figure 4-1 shows the results corresponding to this analysis for a spring stiffness of $k_{lin} : k = F_{s,0}^2 / (2E_{s,0}) = 32.15 \text{ kN m}^{-1}$ and input force command of $\hat{F}_{mech} = -\tau_{8,0} = -1.9 \text{ kN} = f_{s,0}$, i.e.

$$f_{c,mech} : \hat{F}_{mech} = \begin{cases} -\tau_{8,0} & \text{if } t \leq 0.1 \text{ s} \\ 0 & \text{if } t > 0.1 \text{ s} \end{cases} \quad (4-1)$$

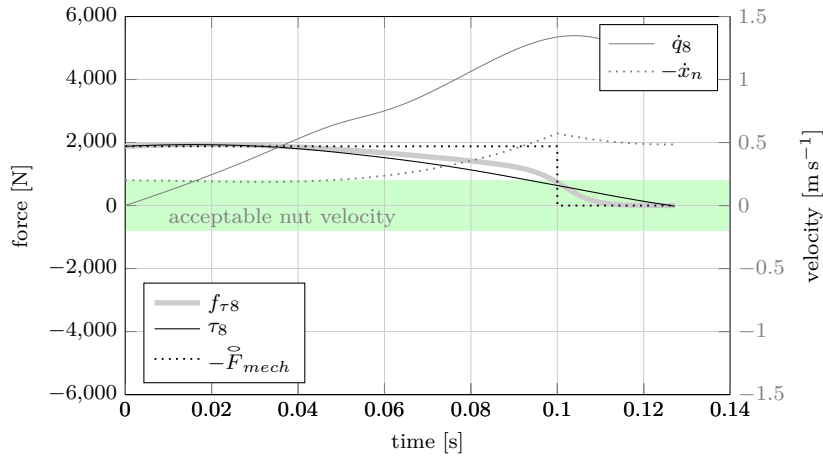


Figure 4-1: Force and velocity results of Skippy for a simple feedforward motor torque command $f_{c,mech} : T_{mech}$ on the motor torque. Results are comparable to those in figure 3-11, but driveline behaviour is less aggressive. Nonetheless, the system is still required to enter physically impossible states, as (for one) the nut is being accelerated beyond its maximum permissible absolute velocity of $\dot{x}_{n,max} = 0.2 \text{ m s}^{-1}$

The jumping height corresponding to this simulation is 4.4 m, without the necessity of lowering the initial spring stiffness. Conveniently, the main spring returns to rest length at lift-off, releasing all of its energy. This is not always the case, as has been confirmed from similar simulations with different system parameters, and as such is considered a lucky coincidence. Figure 4-1 furthermore shows that the nut velocity is approximately constant throughout the first part of the thrust phase, which is a nice result and is not entirely coincidental. The spring is noticeably compliant at $k = 32 \text{ kN m}^{-1}$ and it is approaching its maximum elongation, suggesting that the force derivative is low. If \hat{F}_{mech} is then set to the initial spring force, then the nut is not being accelerated for a substantial time. Furthermore, the transmission ratio during the thrust phase is approximately constant (see figure 3-6), which contributes to the nut velocity being approximately constant. From additional simulations, it was confirmed that a higher input force \hat{F}_{mech} leads to an initial linear increase of the nut velocity, and a lower input force initially decelerates the nut.

However, we also make the following problematic observations:

- The nut starts to accelerate eventually and it exceeds its maximum permissible absolute velocity of $\dot{x}_{n,max} = 0.2 \text{ m s}^{-1}$ by at least a factor of two.
- Due to the increasing nut velocity and constant thrust force, it is evident that more power is drawn at the end of the thrust phase. The amount of power drawn at the end of the thrust phase of $P > 1000 \text{ W}$ is not realisable for the RE35 Maxon motor, which is able to deliver up to approximately 320 W at 27 V and that is *if* the motor is running at its optimal velocity.

Furthermore, the results confirm that momenta are barely changed during the steering phase. This is due to the low spring stiffness during the steering phase and zero motor torque. The steering phase requires the thrust force to be lowered more abruptly, which is not achievable with a low spring stiffness. This supports the idea of using a weakening spring. Decreasing the spring stiffness $k_{lin} : k$ lead to force profiles of \hat{T}_{mech} that are more flattened, and to Skippy jumping higher. This observation makes perfect sense, because we are assuming more initial potential energy stored in Skippy. It is advised not to decrease the spring stiffness more than necessary, because this reduces the control that we have over the nut force profile.

Following the observation that the motor is delivering more mechanical power than it is physically capable of, the next logical step is implementing saturation of motor capabilities.

4-2 Electrical limitations

The goal of this section is to devise a saturation that takes account for the maximum amount of power the motor can deliver. The power that is drawn from the motor in section 4-1 exceeds the maximum amount of power the motor is capable of delivering. The maximum power a motor can deliver for a given maximum voltage is dependent on the velocity of the shaft and the maximum armature voltage or current. Since the nut velocity is not constant, the power saturation is to be implemented as a dynamic saturation. To implement the power saturation correctly, the electric circuit of the motor has to be modelled up to an extent, which is done in section 4-2-1. Voltage and current saturation are then applied in sections 4-2-2 and 4-2-3 respectively, which indirectly induce mechanical power saturation of the motor. The results of the saturations on Skippy's jumping performance are discussed in section 4-2-4.

4-2-1 Electric circuit modelling

The electric circuit of the motor's armature is modelled with terminal resistance R , but without terminal inductance L . The inductance introduces an additional state and complicates the model. With a time constant of a mere $L/R = 0.119 \text{ mH}/0.611 \Omega \approx 0.2 \text{ ms}$ for the 90W Maxon motor, it was decided to neglect its presence. To produce T_{mech} , one requires the following current $J^{[i]}$ assuming that the no-load current J_z creates a constant friction torque, only affected by the sign of the rotor velocity:

$$J = \frac{T_{mech}}{c_T} + \text{sgn}(\dot{\phi}_h) J_z \quad (4-2)$$

^[i] J is used for current rather than I to prevent confusion with rotational inertias

c_T is the torque constant. The electromotive force T_{emf} includes the friction torque and could be calculated as $T_{emf} = Jc_T$. This value is not of interest now but could be useful for analysing heat dissipation later. Multiply the current with the motor's terminal resistance R and one obtains the resistor voltage U_r :

$$U_r = J R \quad (4-3)$$

The back emf voltage U_{emf} can be calculated from the angular shaft velocity $\dot{\phi}_h$:

$$U_{emf} = \frac{1}{c_v} \dot{\phi}_h = c_T \dot{\phi}_h \quad (4-4)$$

$c_v = 1/c_T$ is the speed constant. The armature voltage U is then the sum of the two:

$$U = U_{emf} + U_r \quad (4-5)$$

The required electrical power P_{el} can be computed by taking the product of the voltage and current with a minimum of zero, which is consistent with the assumption that the motor driver electronics is non-regenerative, meaning that if the motor is performing negative work, and therefore operating as a dynamo, then this work is simply lost (as heat) rather than being fed back into the battery.

$$P_{el} = \min(U J, 0) \quad (4-6)$$

P_{el} is the unidirectional power flow from the motor driver circuit to the motor armature circuit. Power losses in the motor driver circuit are not modelled explicitly, but are instead accounted for implicitly by the assumption that the maximum voltage available to the motor is less than the voltage available from the batteries. As such, we consider work done on the system as determined by electricity that flows through the circuit of the motor's armature, where the maximum voltage input is limited to be lower than the theoretical maximum voltage available from the batteries.

4-2-2 Current saturation

The goal of this section is to introduce a simple current saturation based on the model of the electric circuit of the motor's armature described in section 4-2-1. High currents can destroy a motor. The permanent magnet of the DC motor will be demagnetized by the magnetic field caused by the instantaneous current if that current is too high. The limit beyond which this occurs is not specified in data sheets, so we use the stall current as a proxy for the magnetic limit current. Secondly, sustained high currents can also destroy a motor by overheating. Heat generation in the motor is directly related to current through $W_{heat} = J^2 R$. Applying current saturation can thus directly be used to control overheating to an extent. The maximum current is furthermore limited by the motor controller and by the maximum discharge rate of the batteries.

Note that current and voltage are dependent on each other through the shaft velocity. Changing one value requires recalculation of the other. The armature current J is therefore calculated twice, once before voltage saturation in section 4-2-3 and once after voltage saturation. The simplest way of implementing current saturation is by considering an independent maximum current before voltage saturation—i.e. at the current calculated in equation (4-2)—such

that we do not have to recalculate voltage. In theory, this implies that the current can exceed its maximum nonetheless after it is recalculated due to the voltage saturation. However, this can only happen if the motor is doing near maximum *negative* work, which is a scenario to be avoided anyway.

The maximum current used for saturation is the stall current J_s for nominal voltage, for we know that the motor is capable of handling these currents for a short time interval. The stall currents for the Maxon RE35 90 W 24 V and RE30 60 W 24 V motors are 41.1 A and 39.3 A respectively. This limit is also within the maximum current that can be drawn from the batteries, $J_{b,max}$, which is determined by the battery capacity Q_b and discharge rate C_b . For the lithium-polymer batteries that we are considering to use, we have $J_{b,max} = Q_b C_b = 1.3 \text{ A h} \times 35 \text{ h}^{-1} = 45.5 \text{ A}$. The current saturation is not triggered for current simulations; the highest current observed for current simulations is approximately 20 A.

The current is furthermore not limited due to possible overheating, because we assume only short bursts of high power. The stance phase takes approximately $t_{on} = 0.2 \text{ s}$, whereas the thermal time constant of the winding of the RE35 90 W 24 V is 30.2 s. The flight phase lasts approximately $t_{off} = 1.8 \text{ s}$, during which the motor is not intensively loaded. Skippy can reach a height of 4 m in a few hops, in a time that is much shorter than the thermal time constant. Note however that Skippy is not capable of hopping to 4 m indefinitely (that is, until battery depletion) with a duty cycle of approximately $\eta_{on} = 0.2/(0.2 + 1.8) = 10\%$. The maximum duty current I_{on} for an electric motor with a *continuous* duty cycle of η_{on} and nominal current I_N (maximum continuous current) is approximately

$$I_{on} = I_N \sqrt{\eta_{on}^{-1}}$$

according to the Maxon catalogue (Maxon Motor, 2014). For $I_N = 3.62 \text{ A}$ (RE35 90 W 24 V) and $\eta_{on} = 10\%$, we have that $I_{on} = 11.4 \text{ A}$, which is lower than the measured peak of $I = 20 \text{ A}$ in current simulations, and thus suggests that Skippy cannot maintain an indefinite 4 m jumping gait.

4-2-3 Voltage saturation

In addition to current saturation described in section 4-2-2, available voltage is also limited, for which we require another saturation. The static voltage saturation and variable nut velocity provide a dynamic saturation on the mechanical motor power. The armature voltage, U , is subject to the constraint $-U_{max} \leq U \leq U_{max}$, where U_{max} is the maximum voltage at the output of the motor driver electronics. U_{max} is set at 90% of the batteries' total voltage (which is pessimistic). With 2×4 -cell lithium-polymer batteries of 3.7 V per cell (as mentioned in section 2-1-3), this equates to $U_{max} = 26.64 \text{ V} \simeq 27 \text{ V}$. It is expected that we are able to draw slightly more voltage from the batteries in reality.

Given that the rotational velocity $\dot{\phi}_h$ cannot be changed instantaneously, we can compute the new corresponding current and torque using equations (4-2) to (4-5):

$$J = \frac{U - U_{emf}}{R} \quad (4-7)$$

$$T_{mech} = J c_T - \text{sgn}(\dot{\phi}_h) J_z c_T \quad (4-8)$$

It has been explained in section 4-2-2 that voltage saturation is applied after current saturation, for which the current is required to be recalculated after voltage saturation.

4-2-4 Results

Having included the voltage and current saturation, we perform the same simulation as in section 4-1. Figure 4-2 shows that the torque is heavily saturated. The saturation shows that the motor torque is primarily ruled by what the motor is capable of delivering, rather than being determined by the constant torque command. The result is a big improvement, because it shows that the torque command used previously is impractical, and the behaviour now is a lot closer to being feasible. The saturated torque in turn affects the nut velocity and momentum build-up. As a consequence, Skippy jumps lower than previously at 3.89 m.

During the thrust phase, the torque is slightly *not* saturated during $0.018 < t < 0.035$ s, where the voltage is slightly lower than its maximum. In fact, there seems to be no reason *not* to use maximum voltage during the thrust phase, whereas higher voltage corresponds to more power. As such, in future simulations we base our thrust phase on feedforward maximum voltage. This is effectively done by increasing $f_{c,mech}$ for $t \leq 0.1$ s in equation (4-1) up to $\bar{F}_{mech} = -3000$ N, which is also the maximum force that can be transmitted through the ball screw. For this particular simulation, this action led to no significant change of the jumping height.

3.89 m is slightly below the target height of 4 m, but this is not very problematic. A bigger concern is the nut velocity, which—even though reduced—is still being pushed beyond its maximum velocity. The nut velocity exceeds 0.20 m s^{-1} before the end of the thrust phase, reaching 0.24 m s^{-1} at $t = 0.1$ s after which T_{mech} drops to zero, causing the nut to decelerate. The nut velocity can be lowered by lowering the motor torque. A nut velocity saturator is developed in section 4-3-1 based on this principle.

An additional simulation has been done where T_{mech} is not dropped to zero at $t = 0.1$ s, but where it is held constant until lift-off at $t \simeq 0.12$ s. The nut reaches an even higher velocity of 0.28 m s^{-1} at lift-off.

4-3 Velocity limitations

In addition to voltage and current saturation implemented in section 4-2, nut velocity saturation is implemented to prevent the system from obtaining non-feasible velocities. The system knows a few other velocity limitations, e.g. those of the thrust bearings holding the spindle. Since their acceptable velocities are much higher than that of the ball screw, there is no need to take them into account. The implemented nut velocity saturation is best described as a proportional sliding mode controller, where the admissible nut acceleration is cut short the closer the nut velocity approaches its maximum permissible absolute velocity $\dot{x}_{n,max} = 0.2 \text{ m s}^{-1}$, as further explained in section 4-3-1. The results of the hopping performance with the velocity saturation (in cooperation with the electrical saturation) are discussed in section 4-3-2.

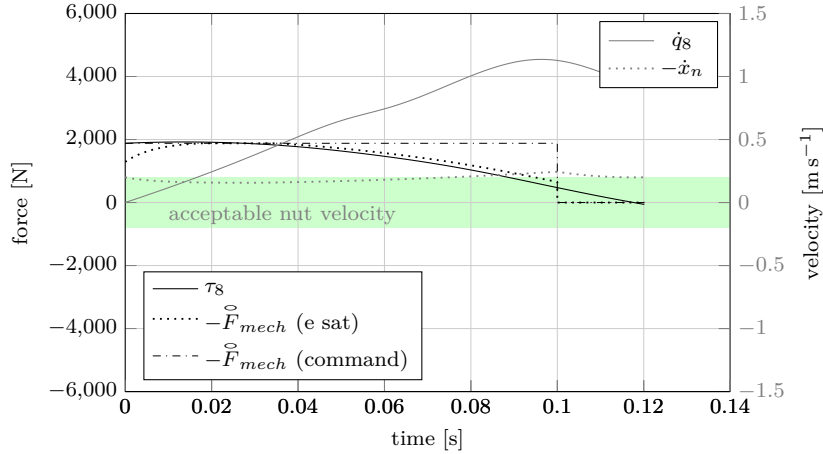


Figure 4-2: Force and velocity results of Skippy with electrical saturation. (command) refers to the non-saturated torque command imposed on the motor torque (equation (4-1)) and (e sat) refers to the electrical (current and voltage) saturation of that torque.

4-3-1 Nut velocity saturation

The goal of this section is to devise and implement saturation that prevents the nut from being accelerated beyond its acceptable velocity. The nut velocity can be controlled by imposing an acceleration on the nut. From equation (3-20) we have that $F_s + \hat{F}_{mech} = F_d$ is proportional to the nut acceleration. Instantaneously, F_s is a given since it is a function of only the spring length x_s . Dynamic saturation is applied on \hat{F}_{mech} , based on an acceptable value for F_d , $F_{d,a}$, to limit the nut acceleration when it approaches its maximum permissible absolute velocity of $\dot{x}_{n,max} = 0.2 \text{ m s}^{-1}$, and to decelerate the nut when it exceeds this maximum velocity. We have that $F_{d,min} < F_{d,a} < F_{d,max}$, where $F_{d,min}$ and $F_{d,max}$ are dynamic. This dynamic saturation is implemented as follows and takes the shape of a sliding P-controller:

$$\hat{F}_{mech} = \text{median} \left(\underbrace{k_n(\dot{x}_n - \dot{x}_{n,max})}_{F_{d,min}} - F_s, \hat{F}_{mech}, \underbrace{k_n(\dot{x}_n - \dot{x}_{n,min})}_{F_{d,max}} - F_s \right) \quad (4-9)$$

in which $\dot{x}_{n,min} = -\dot{x}_{n,max} = -0.2 \text{ m s}^{-1}$. k_n is a gain that determines the aggressiveness of the saturator. If the gain is too low, then too much valuable nut velocity will be sacrificed. If it is too high, then the integration will take too many unnecessary steps due to a micro-wobble around the maximum nut velocity. The gain is set to $k_n = -100000$. This implies, for instance, that at $\dot{x}_n = 0.17 \text{ m s}^{-1}$ the nut can still be pushed up to $F_d = 3000 \text{ N}$ (approximately the maximum force that can be transmitted), linearly decreasing to $F_d = 0 \text{ N}$ at $\dot{x}_n = 0.2 \text{ m s}^{-1}$ (zero acceleration) and $F_d = -3000 \text{ N}$ at $\dot{x}_n = 0.23 \text{ m s}^{-1}$ (forced deceleration).

The nut velocity saturation is implemented before the electrical saturations. The nut velocity saturation feeds the saturated torque to the current saturator and then velocity saturator, after which the current is recalculated.

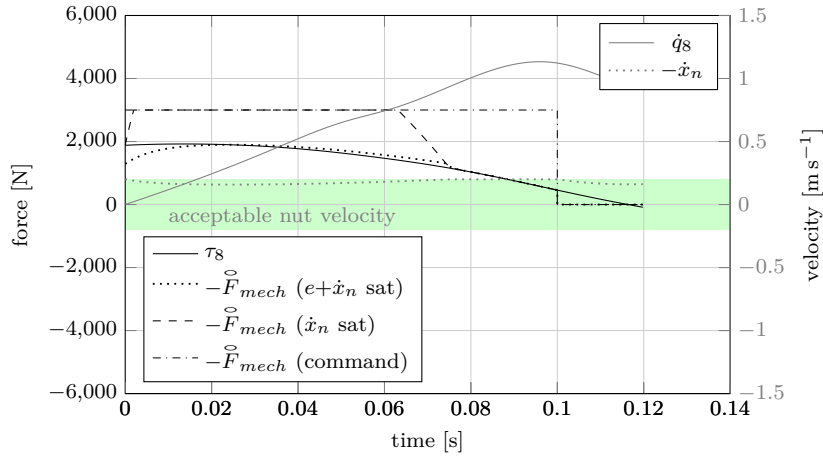


Figure 4-3: Force and velocity results of Skippy with combined nut velocity saturation and electrical saturation on the motor torque. Within limitations of the motor and nut velocity, Skippy is still able to jump up to reasonable heights. (command), $(\dot{x}_n \text{ sat})$ and $(e+\dot{x}_n \text{ sat})$ denote the sequential saturation steps on the motor torque, where (command) is the non-saturated torque command $f_{c,mech}$, and $(\dot{x}_n \text{ sat})$ and $(e+\dot{x}_n \text{ sat})$ denote the first (nut velocity) and second (electrical) step of saturation respectively.

4-3-2 Results

With the nut velocity saturation *and* electric saturations implemented, results of the otherwise identical simulations in section 4-1 are displayed in figure 4-3. Skippy's jumping height has further reduced to 3.87 m, which is still not a major concern. However, heat generation due to friction is yet to be implemented, which likely reduces the jumping height.

Figure 4-3 shows that the nut velocity saturation is satisfactory, as the nut is not accelerated beyond its maximum permissible absolute velocity $\dot{x}_{n,max} = 0.2 \text{ m s}^{-1}$. Moreover, the voltage and current saturations do not interfere with the nut velocity saturation and stack well.

The forces in the driveline are also still within bounds. The nut force F_n does not approach the maximum permissible absolute axial ball screw force of $F_{n,max} = 3.1 \text{ kN}$, as can be seen in figure 4-3, which shows a maximum absolute of $|F_n| = |\tau_8| \approx 2.0 \text{ kN}$. Moreover, the motor shaft torque T_{out} (see equation (3-17)) does not exceed the desired maximum of 1.0 N m, as can be seen in figure 4-4, where the maximum absolute is $T_{out} \approx 0.6 \text{ N m}$.

It is concluded that the nut velocity and electrical saturators work remarkably well. They allow us to experiment with Skippy more efficiently, as the two most important of Skippy's many physical limitations are taken into account. It is noteworthy that the torque profile is primarily dominated by the saturations. It emphasizes that fitting existing sets of profiles as attempted in chapter 3 is an inefficacious and likely unproductive approach.

Achieving decent jumping with viable voltages, currents, nut velocities and forces in the driveline provides a warrant for proceeding to analyse other limitations and requirements of Skippy. In subsequent sections, we include mechanical heat generation due to friction in section 4-4-1 and we analyse steering performance in section 4-5. The jumping height, which has thus far been reduced to 3.87 m, may be further reduced due to mechanical friction.

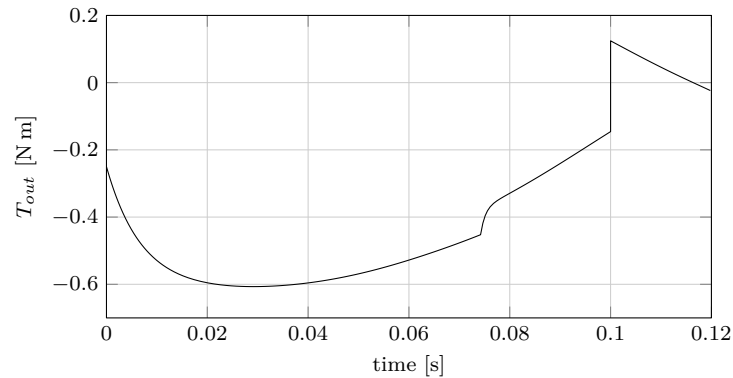


Figure 4-4: Motor shaft torque T_{out} , experienced by the coupling. The torque is of interest for coupling selection for the mechanical design. Preferably, $|T_{out}| < 1 \text{ N m}$

4-4 Energy analysis

This section looks into system energy of Skippy. Ball screw friction is modelled in section 4-4-1, which converts useful kinetic energy to useless heat, and thus likely further reduces Skippy's jump height. In addition, an energy flow analysis is introduced in section 4-4-2.

Mechanical friction is to be included to obtain more realistic and accurate simulations. The friction torque of the motor is already taken into account by computations in this chapter (section 4-2-1). Ball screw friction is one of the other mechanical sources of heat loss, treated in section 4-4-1. Additional sources of mechanical friction include joints and passive elements. All joints are borne by ball and needle bearings that cause little friction, so this source of friction is neglected in the model. Depending on the type of spring elements used, the significance of the friction and its friction model are different. It is yet unknown which type of spring elements are going to be used. Moreover, the fact that real springs cannot be purely elastic has been taken into account in the design constraint that only 50% of the necessary energy to perform a 4m hop can be retrieved from the previous hop (see section 2-1-5). As such, we will focus only on ball screw friction for the moment.

Subsequently to the introduction of ball screw friction, an energy flow analysis is introduced in section 4-4-2. The analysis is a response to the increased complexity of Skippy's model. It serves as a validity check for our simulations and it aims to improve our understanding of system behaviour. The effect of the ball screw friction on the jumping height is also shortly discussed in this section.

4-4-1 Ball screw friction

This section describes the friction model of the ball screw that is used in Skippy. In order to devise a valid friction model, it is essential to have a basic understanding of the contact behaviour of balls between the nut and spindle of the ball screw.

As mentioned in section 2-1-3, the ball screw used for Skippy is a 4-point contact ball screw. Furthermore, the channels where the balls roll are helical. From these two facts, it follows necessarily that the four contact points are not exactly co-planar. As such, the balls roll

with little sliding, from which follows that there must be sliding friction. In addition, balls roll about their contact normals. The balls press firmly into the channel because of the high spring force that is acting on the nut. In reality, this will cause the balls to make small area contacts (rather than perfect point contacts), and since balls rotate about their contact normals, it follows that torsional slipping occurs about the contact normals of the balls.

When ball screws are operating at substantial power transmission levels, losses are measured to be close to a constant percentage of the power throughput (NSK, 2008; SBC, 2015). This justifies the practice of quoting the efficiency of a ball screw as a single percentage. The efficiency is dependent on the direction of power throughput. Forward efficiency η applies when the power is transferred from the spindle (rotation) to the nut (translation) and backward efficiency $\eta' < \eta$ applies if the power flow is the other way round.

A simple friction model is to be implemented that takes into account frictional effects that (1) are caused by dry friction in reality and (2) sap a percentage of the power throughput that is only a function of the direction of the power flow. A model that fits these criteria is a Coulomb friction model in which the ‘normal’ force is the thrust force magnitude. The model is described below.

Friction model

In equation (3-21), the spring force F_s is defined as being equal to F_n . Consider now the following relation:

$$F_n = F_s - F_{fric} \quad (4-10)$$

where F_n is the force resulting from the motor and driveline inertia acting on one side of the nut and F_s is the spring force acting on the other side of the nut. F_{fric} is the friction force in the nut, modelled as Coulomb friction resulting from the high normal force pressing on the balls. The coulomb friction results from the imperfect rolling motion (i.e. sliding) of the balls.

To solve the EOM, we have to be able to calculate the acceleration from equation (3-20):

$$\ddot{x}_n = \frac{F_d}{\bar{m}_h} = \frac{\bar{F}_{mech} + F_n}{\bar{m}_h}$$

However, F_n is unknown. F_{fric} has to be known to obtain F_n . As such, an expression for F_{fric} has to be found.

In case of motion ($\dot{x}_n \neq 0$), the sign of F_{fric} is equal to that of the nut velocity and F_{fric} is defined to be equal to the stiction force F_{stic} , which is the static force required to induce motion. This follows from the assumption that the static and kinetic coefficient of friction are equal. We have:

$$\left. \begin{aligned} F_{fric} &:= F_{stic} \\ \text{sgn}(F_{stic}) &= \text{sgn}(\dot{x}_n) \end{aligned} \right\} \text{if } \dot{x}_n \neq 0 \quad (4-11)$$

The magnitude of F_{stic} is yet to be defined, as equation (4-11) only defines its sign. Its magnitude can be obtained through the known forward efficiency η (as it can be found in catalogues of ball screws). The forward efficiency η is defined as:

$$\eta := \frac{P_s}{P_n} = \frac{\dot{x}_n F_s}{\dot{x}_n F_n} = \frac{F_s}{F_n} \quad \text{with } 0 < \eta < 1 \quad (\text{forward efficiency}) \quad (4-12)$$

in which P is power. Recall that this is the efficiency of the ball screw if the rotational motion (input F_n) drives the translational motion (output F_s). From equations (4-10) to (4-12) we have

$$F_{stic} = \text{sgn}(\dot{x}_n) \left(\frac{|F_s|}{\eta} - |F_s| \right) = \frac{1-\eta}{\eta} |F_s| \quad (4-13)$$

Equation (4-13) provides the *backward* efficiency η' :

$$\eta' = \frac{F_n}{F_s} = \frac{|F_s| - |F_{stic}|}{|F_s|} = 1 - \frac{1-\eta}{\eta} = \frac{2\eta-1}{\eta} \quad (\text{backward efficiency}) \quad (4-14)$$

This is the efficiency if translational motion drives the rotational motion. The exact same relation is found in graphs of catalogues of ball screw manufacturers (NSK, 2008; SBC, 2015). For an efficiency of $\eta = 0.9$, the backward efficiency is $\eta' = 0.8889$. If $\eta \leq 0.5$, then $\eta' \leq 0$, implying that the ball screw is not back-drivable.

The EOM can now be solved. A distinction has to be made between scenarios where stiction is experienced and where it is not. During stiction, the nut velocity and acceleration are zero: $\dot{x}_n = \ddot{x}_n = 0$. This implies that $\widehat{F}_{mech} = -F_n \rightarrow F_d = 0$. Stiction is maintained as long as $|F_s - F_n| < |F_{stic}| \rightarrow |F_s + \widehat{F}_{mech}| < |F_{stic}|$. In a single line:

$$F_d = 0 \leftarrow F_{fric} = F_s + \widehat{F}_{mech} \text{ if } \dot{x}_n = 0 \text{ and } |F_s + \widehat{F}_{mech}| < |F_{stic}| \quad (4-15)$$

If the velocity of the nut is positive or negative, or if the velocity is zero and about to become positive or negative, the friction force is $F_{fric} = F_{stic}$:

$$F_d = \widehat{F}_{mech} + F_s - F_{stic} \leftarrow F_{fric} = F_{stic} \text{ if } \dot{x}_n \neq 0 \text{ or } \left(\dot{x}_n = 0 \text{ and } |F_s + \widehat{F}_{mech}| > |F_{stic}| \right) \quad (4-16)$$

The conditions $\left(\dot{x}_n = 0 \text{ and } |F_s + \widehat{F}_{mech}| > |F_{stic}| \right)$ in equation (4-16) apply to situations where the nut is “breaking out” of its stiction.

Efficiency of Skippy's ball screw

The constant percentage η that describes the efficiency of a ball screw depends on the friction coefficient of the materials (related to the quality of the ball screw) and the lead angle, and is generally very high. The lead angle ϕ_{bs} is calculated as follows:

$$\phi_{bs} = \arctan \left(\frac{n_{bs}}{\pi D_{bs}} \right)$$

in which n_{bs} is the lead and D_{bs} the diameter of the spindle, measured as the centre-to-centre distance of two balls on opposite sides of the spindle. The 2412-M8 ball screw that we are considering has a lead $n_{bs} = 2$ mm and diameter $D_{bs} = 8$ mm, from which follows that $\phi_{bs} = 5.5^\circ$. The data sheet of corresponding ball screw does not provide an efficiency nor friction coefficient, but catalogues of competing manufacturers show that ball screws with identical lead angle $\phi_{bs} = 5.5^\circ$ have forward efficiencies ranging from 89% to 97% (NSK, 2008; SBC, 2015). We consider a forward efficiency of $\eta = 90\%$.

4-4-2 Energy flow analysis and results

This section analyses the energy flows in Skippy. These flows improve our understanding of Skippy's behaviour during the launch phase, and can possibly be used to see how Skippy's performance can further be improved.

As a response to the introduced complexity of the model, energy flows have been analysed. The total sum of all system energies E_{sys} and work done by the system W_{out} minus work done on the system W_{in} should be constant, which has been verified. See appendix C for respective calculations, energy audits that are performed as a check for the correctness of the simulations and a full definition of all the energy terms. Figure 4-5 shows the cumulative energy flows of the system of the analysis in section 4-3-2 including ball screw friction:

- All energy flows above zero represent system energies E_{sys} that can theoretically be converted to other types of system energies. If this flow is decreasing, it means that the system is dissipating more heat than the motor is doing work on the system.
- The energy flows below zero correspond to heat loss, which are unusable and therefore considered as work done by the system W_{out} . W_{out} is monotonically increasing, since heat can never be harvested by the system.
- The electric energy delivered to the system equals the work done on the system W_{in} . W_{in} is monotonically increasing because of the fact that electrical power flow is defined to be always positive (see equation (4-6)), implying that the sum of W_{out} and E_{sys} must be monotonically increasing as well.

The energy graph confirms that both springs are approximately at rest at lift-off, as has been observed in section 4-1. Most of the energy is desirably converted to translational kinetic energy of the system $E_{k,xy}$, which is the kinetic energy associated to Skippy as a point mass located at Skippy's COM. The graph also shows that some of the system's energy is uselessly, yet inevitably converted to rotational kinetic energy $E_{k,\phi}$. $E_{k,\phi}$ is defined as the total kinetic energy of the robot's bodies minus $E_{k,xy}$, so it includes (1) all kinetic energy due to relative (rotational) motions of bodies within the system and (2) kinetic energy that results from the overall rotation of the mechanism about its COM. The former (i.e. relative rotations) are considered an inevitable energy 'sink': even if Skippy lifts off with net zero rotational momentum, the leg and torso are still required to counter-rotate at lift-off, or Skippy would not have an upward velocity. The rotational kinetic energy stored in the driveline $E_{k,h}$ is fortunately negligible. The gravitational energy furthermore increases as Skippy stretches towards lift-off.

The energy graph furthermore shows that most heat losses occur in the motor, W_{JT} . Heat losses from the ball screw W_{bs} are minor. W_{reg} does not appear in figure 4-5 as they are heat losses that occur if the motor does significant negative work, which is not the case in the current simulation.

The motor does approximately $W_{mech} = 27 \text{ J}$ work during the thrust phase. With a ball screw forward efficiency of $\eta = 90\%$ (see section 4-4-1) it is expected that the ball screw generates approximately $W_{bs} = 2.7 \text{ J}$ heat, and consequently causes Skippy to jump approximately $W_{bs}/(mg) = 14 \text{ cm}$ less high. The new simulation shows that this estimation of W_{bs} was accurate. Remarkably, Skippy jumps only 7.1 cm lower at 3.79 m (previous: 3.87 m). It appears that this is related to the fact that the friction causes the nut to be slowed down such

that the velocity saturation—which effectively reduces the voltage and thus power output from the motor—is activated later than before.

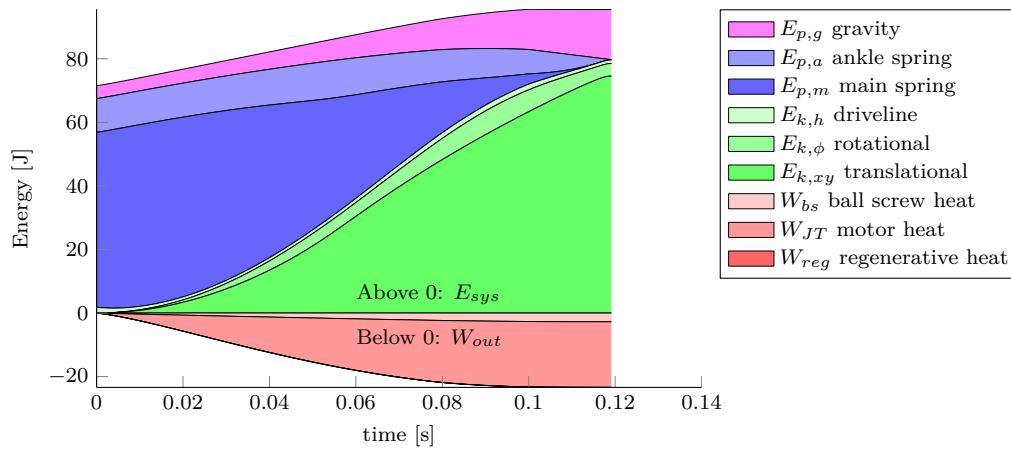


Figure 4-5: A clear overview of cumulative system energies (above zero) and heat dissipation (below zero). Most of Skippy’s energy is efficiently converted to useful translational kinetic energy needed for jumping (green), with the exception of a few inevitable energy sinks. The ball screw produces little heat.

4-5 Steering analysis

Another item to be looked into is the steering phase, which has thus far been neglected in this chapter. The goal of this section is to investigate how the various driveline limitations limit steering abilities. A simplified steering strategy is introduced in section 4-5-1, which has the aim of obtaining an indication of the limits of steering. Corresponding results are discussed in section 4-5-2. As a result of the observed steering capabilities, the linear spring is replaced by a bilinear spring in section 4-5-3. The new jumping performance is analysed in section 4-5-4.

4-5-1 Simplified steering strategy

This section describes an executable feedforward steering strategy, with the purpose of simplifying the feedback steering strategy proposed in section 2-2-5. The limits of what can be achieved by steering can be obtained by putting the GRF on both edges of its friction cone, which is the strategy that has been employed in section 2-2-5. However, it was found in section 3-2-4 that—for current system parameters—it is not feasible to produce a torque command that is required for this steering action due to its aggressive behaviour. The corresponding torque command requires a voltage and current that are not available and it forces the nut to exceed its maximum velocity.

Instead, we try to simplify the steering strategy, based on earlier observations of steering torques in section 2-2. We observe from figures 2-6b and 2-9b (and additional simulations) that the lift-off momenta are generally positively correlated with the knee torque. A positive steering action (higher horizontal and angular lift-off momentum) requires (on average) a

higher torque and a negative steering action (lower horizontal and angular lift-off momentum) requires a lower torque. As a simplification of the steering action, we impose a negative and positive command torque for $t > 0.1$ s. The magnitude of the command torque is chosen sufficiently high to have at least one saturator triggered during the entire steering phase.

Note that this new strategy is not as good as the original steering strategy, and it is not to be used on a real version of Skippy. The goal of employing this simplified strategy is (1) to find out what can approximately be achieved with steering and (2) to find out what is limiting the capability of steering.

4-5-2 Interim steering results

Momentum results for both minimum and maximum steering actions are plotted in figure 4-6a, together with the GRF ratio in figure 4-6b. Force and velocity results are plotted in figure 4-7 for the minimum steering action. The jumping height is 3.82 m and 3.87 m for the minimum and maximum steering action respectively. It is satisfactory to observe from figure 4-6a that Skippy is still able to fully negate the positive momenta build-up during the landing phase. However, the difference between built-up momenta for both actions at the end of the steering phase is disappointing. Figure 4-6b shows that slipping only occurs in the very last milliseconds^[ii], and slightly earlier for negative steering. The fact that slipping only occurs in the very last moment before lift-off supports the validity of the steering test.

By observing figure 4-7, steering capabilities seem mainly to be limited by nut velocity saturation. The steering capabilities are poor mainly because of the low spring stiffness, which does not permit fast changes of force with the limited nut velocity. Other than changing the spring stiffness, steering capabilities could be improved by decreasing the transmission ratio towards the steering phase, effectively increasing the nut velocity. The steering capability is also limited by the driveline inertia, which limits the nut acceleration. Changing the nut from its minimum to maximum velocity (i.e. $|\Delta\dot{x}_n| = 0.4 \text{ m s}^{-1}$) takes approximately 0.01 s, which is approximately half of the steering phase, as can be seen in figure 4-7. This acceleration demands a coupling torque of approximately $T_{out} = 2.5 \text{ N m}$ and terminal motor current of $J = 40 \text{ A}$. As such, preceding nut velocity saturation, current saturation dominates the torque profile shortly after the thrust phase at $0.100 < t < 0.105$ s. Figure 4-7 does not separate current saturation from voltage saturation, but its effect can be seen by the limited thrust force $-\bar{F}_{mech} \approx -3.8 \text{ N}$ at $0.100 < t < 0.105$ s. If we have a desire to reduce the current (due to overheating) or coupling torque (due to mechanical coupling or shaft failure), the acceleration of the nut is further reduced.

Improving steering is primarily done by increasing the spring stiffness in section 4-5-3.

4-5-3 Bilinear spring profile parameters

As a response to the poor steering performance observed in section 4-5-2, this section re-introduces the bilinear spring profile defined in equation (3-24) (section 3-2-2), which is expected to improve the steering performance. The bilinear spring introduces two new parameters which are required to be defined properly to comply with initial force and potential energy storage.

^[ii]Recall that we assume slipping to occur for $|F_x/F_y| > 1$

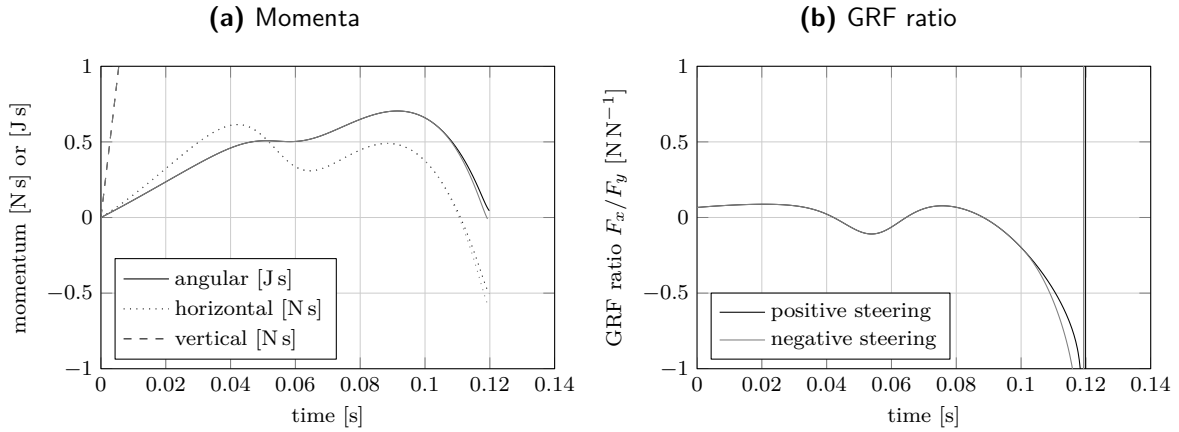


Figure 4-6: Momentum and GRF results for minimum (grey) and maximum (black) steering actions with a linear spring. Steering capabilities are insufficient, as the difference between both steering actions are minimal. The GRF ratio plots tell us that slipping only occurs in the very last milliseconds.

Given an initial amount of available spring energy $E_{s,0} \simeq 55 \text{ J}$ (estimated in section 3-2-2) and an initial spring force $F_{s,0} = \tau_{8,0}$ (computed from the initial GRF $F_y = 200 \text{ N}$), the spring profile of the main spring is partly defined. The spring profile is fully determined if the spring is linear, because the only parameter that affects the spring behaviour is the stiffness: $k_{lin} : k = F_{s,0}/(2E_{s,0})$. The bilinear spring has three parameters that affect its behaviour: k_1 , k_2 and l_{bi} . A dependency can be introduced between these parameters to fulfil the initial force and energy constraints. If k_2 is selected as dependent parameter, we have:

$$k_2 = \frac{\frac{1}{2}F_{s,0}^2 - \frac{1}{2}l_{bi}^2k_1^2}{E_{s,0} - \frac{1}{2}k_1l_{bi}^2} \quad (4-17)$$

This relation implies that k_1 and l_{bi} are still free to choose, which is favourable. For k_2 it is mainly important that it is sufficiently low for the nut to be able to travel with adequate speed, of which we know that it is, because $k_2 < k_{lin}$ and preceding sections have shown that k_{lin} is already capable of making Skippy jump up to nearly 4 m with saturated nut velocity. However, k_1 and l_{bi} will affect balancing and steering capabilities and are yet to be found. We know that $k_1 > k_{lin}$ and it has been shown in section 4-3 that k_{lin} is not sufficiently stiff to permit adequate steering. As an initial guess for adequate steering, we choose $k_1 = 150 \text{ kN m}^{-1}$, almost five times higher than $k_{lin} = 32.15 \text{ kN m}^{-1}$. This stiffness theoretically allows for a change of spring force $\Delta F_s \approx 600 \text{ N}$ in $\Delta t = 0.01 \text{ s}$, which is approximately half the steering phase—calculated as follows:

$$\Delta F_s \approx \frac{F_s}{x_s} |\dot{x}_{n,max} - \dot{x}_{n,min}| \Delta t = 2k(x_s) \dot{x}_{n,max} \Delta t$$

With an approximate gear ratio of $R_2 = 0.1$, $\Delta F_s \approx 600 \text{ N}$ corresponds to a knee torque of $\Delta \tau_6 \approx 60 \text{ Nm}$, approximately equal to the difference in torque observed for the various steering actions in section 2-2-6. l_{bi} is initially approximated as

$$l_{bi} = \sqrt{\frac{F_{s,f}^2}{k_{lin} k_1}} = 6.6 \text{ mm}$$

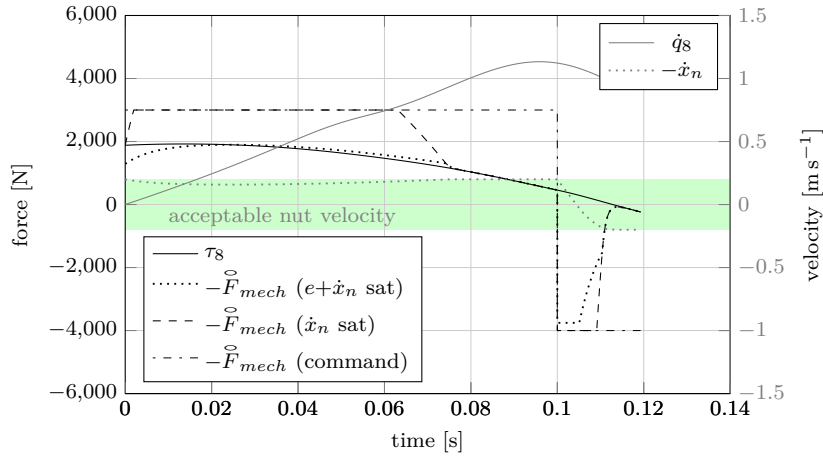


Figure 4-7: Force and velocity results of Skippy for a ‘minimum’ steering action. Forcing the nut to run at either of its velocity extremities ($\dot{x}_{n,min}$ and $\dot{x}_{n,max}$) has little effect on the thrust profile due to the compliant spring. Steering capabilities are primarily limited by the over-compliant spring in combination with nut velocity saturation, but also by the limited ability to accelerate the nut (related to current saturation, voltage saturation and high rotor inertia), as it takes approximately half the steering phase to accelerate the nut over its full velocity range, which can be seen for $0.10 < t < 0.11$ s.

where $F_{s,f}$ is equal to the spring force at the transition between thrust and steering at t_f : $t = 0.1$ s for an identical model that uses the linear spring (i.e. the model analysed in section 4-4-2). The above relation implies that an equal amount of energy is stored in the spring at the transition between the thrust and steering phase for both models.

4-5-4 Results

This section analyses the results of Skippy after the introduction of the bilinear spring as defined in section 4-5-3. From running the simulation with the bilinear spring, we observe a problem. It is observed in figure 4-8 that the spring fully unloaded itself prior to lift-off, after which it starts to be loaded in the other direction. The coincidence that we struck upon in section 4-1, where the spring conveniently reaches zero length by the end of the thrust phase, is lost due to new spring profile. The behaviour follows from the fact that a spring with a weakening profile is able to store the same amount of energy in less spring elongation, whereas the nut travels with the same speed. This causes the the spring to be unloaded prematurely. The fact that the spring starts pushing the joint makes it even worse. The pushing force corresponds to leg retraction and reduces F_y , such that Skippy prematurely leaves the ground. The ankle is unable to finish its steering action and the energy stored in both the ankle and main spring cannot be used to build up additional momentum. The energy loss is best observed in figure 4-9. The jumping height of Skippy has reduced to 3.35 m, which is the result for the positive steering action. The result for the negative steering action is even worse.

The problem could be solved by increasing the initial spring energy $E_{s,0}$ or by decreasing the initial spring force $F_{s,0}$, which effectively decreases k_2 (equation (4-17)) and increases the initial spring elongation. However, changing $E_{s,0} = 55$ J and $F_{s,0}$ is to be avoided as this

corresponds to making the assumptions about the landing phase more optimistic, which is less likely to be realisable. Instead, we have to focus on changing the transmission ratio to alter the travel of x_s , which ought to alter the efficiency of energy conversion. This strategy is addressed in section 4-6-2.

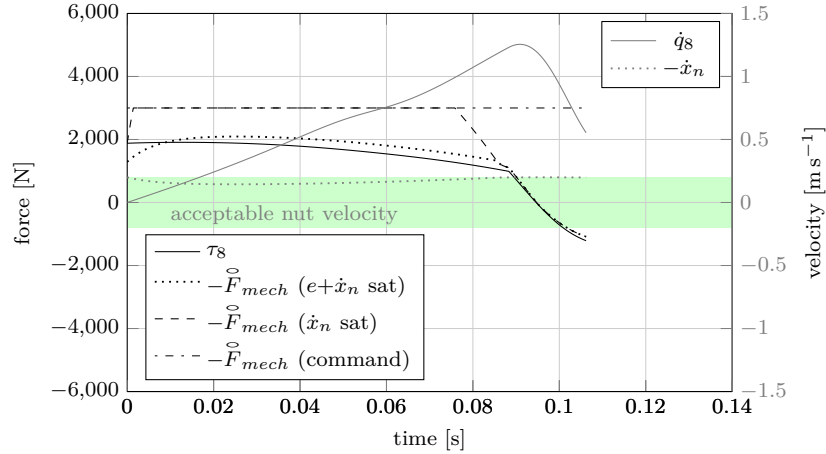


Figure 4-8: Force and velocity results of Skippy with bilinear main spring. Springs are not unloaded at lift-off and lift-off is earlier, causing Skippy to lose jumping height.

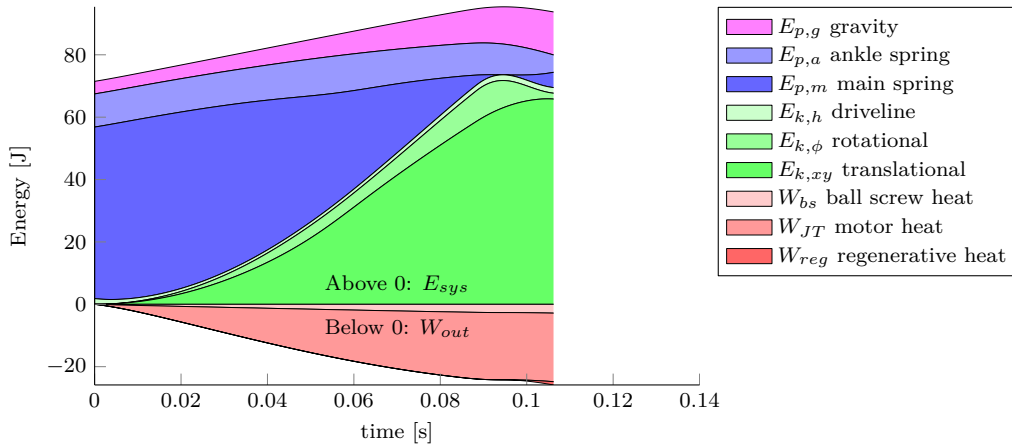


Figure 4-9: Energy flow of Skippy with bilinear main spring. The energy stored in springs at lift-off indicate that energy is sub-optimally transferred to useful kinetic energy.

4-6 Energy-based model improvements

With the introduction of the nut velocity and electrical saturations and ball screw friction, Skippy's jumping height has been reduced to 3.79 m (see section 4-4-2). The inclusion of the bilinear spring, depending on the regressiveness, further reduces the jumping height to 3.35 m (see section 4-5-4). From an energetic point of view, the following options could be exploited to make Skippy jump higher:

- increase initial system energy;

- increase mechanical work done by the motor;
- increase conversion to useful kinetic (translational) energy, i.e. improve energy conversion efficiency.

Reasonable approximations have been made concerning the initial energy and thrust force in section 2-1, and little more can be said about these numbers until we take a closer look at the landing phase, which does not fit in the current thesis project.

Prior to the inclusion of a bilinear spring, it was observed in section 4-4-2 from figure 4-5 that all energy is converted to desired translational kinetic energy and some inevitable types of energy (gravitational potential and rotational kinetic energy and heat), as both springs are approximately at rest at lift-off. Little more can be done to improve energy conversion efficiency. As a result, the focus is primarily on increasing the mechanical work that is done by the motor, treated in section 4-6-1, without the inclusion of a bilinear spring.

With the inclusion of a bilinear spring, it was observed in section 4-5-4 that energy conversion deteriorated. Both springs had energy stored in them at lift-off. The main spring was prematurely unloaded after which it was starting to be negatively loaded towards lift-off. This led to Skippy prematurely lifting off. As a consequence, the ankle spring did not fully unload and the launch phase was shorter. Improvement of energy conversion is treated in section 4-6-2, with the inclusion of a bilinear spring.

4-6-1 Increase mechanical work

The goal of this section is to analyse possibilities to increase the mechanical work done by the motor. Mechanical work by the motor can be increased in the following two ways

- Increase mechanical power output by the motor.
- Elongate the launch phase, of which mainly the thrust phase, to elongate positive mechanical power output over time.

It was found that effectively elongating the launch phase is only possible when increasing the size of Skippy, effectively increasing the stroke Δy . If the launch phase is elongated without increasing Δy , then this corresponds to a lower lift-off velocity and thus a lower jumping height. We are trying to avoid increasing Skippy's size, because this implies that it becomes harder or even impossible to design Skippy within the 2 kg budget whilst keeping Skippy physically robust and compact. We have to focus on increasing mechanical power output by the motor instead.

The maximum power output is dependent on the rotor velocity and voltage. The rotor velocity ω and torque τ for maximum mechanical power output P_{\max} are roughly equal to half the no-load velocity ($\omega_z/2$) and half the stall torque ($\tau_s/2$) respectively, which scale roughly proportional to the armature voltage^[iii], from which follows that maximum mechanical power output scales approximately quadratically with voltage:

$$\left. \begin{aligned} P_{\max} &\simeq \frac{\tau_s \omega_z}{4} \\ U' = SU &\simeq \frac{S\tau_s}{2} \frac{R}{c_T} + c_T \frac{S\omega_z}{2} \end{aligned} \right\} P'_{\max} \simeq S^2 P_{\max} \quad (4-18)$$

^[iii]Scaling is not entirely proportional due to the constant friction torque.

where S is a scaling factor and c_T the torque constant. The optimal rotational velocity for maximum power is approximately equal to

$$\omega_{opt} : \dot{\phi}_h = \frac{\omega_{z,N} U_{max}}{2 U_N}$$

in which N corresponds to nominal values, retrieved from data sheets.

The power output can be maximized by maximizing the voltage and by having the motor rotate at approximately half the no load velocity for that voltage. Most of the thrust phase is already determined by maximum voltage output due to the voltage saturation. However, the maximum voltage can be increased by introducing an additional battery cell. Moreover, we can alter the transmission ratio by adjusting the lever length, which affects the duration of the thrust phase that is governed by voltage saturation, and the nut velocity during that phase. Both options are exploited and further explained in the remainder of this section. Lastly, we analyse the capabilities of a previously mentioned lighter motor—the Maxon RE30 60 W 24 V—as it has a number of advantages over the heavier RE35 motor.

Additional battery cell

Through iterative studies of simulations in this section, it was decided to add an additional battery cell to increase the maximum voltage. We have moved from 2×4 -cell lithium-polymer batteries to 3×3 -cell lithium-polymer batteries. If the assumption is made that 90% of the voltage can be used (see section 2-1-3), we have $0.9 \times 9 \times 3.7 = 0.9 \times 33.3 = 30$ V available. The additional cell corresponds to an increase of maximum power of approximately $(8/9)^2 - 1 = 26\%$. A disadvantage of increasing voltage is increasing heat generation and adding weight, as such, we try to avoid adding more batteries. The decision to add an additional cell was thence not taken lightly. With 9 cells or 30 V, the 24 V motor can be over-driven up to 25%.

With the 9th cell, Skippy's performance instantly improves by jumping up to 3.97 m instead of 3.79 m^[iv].

Changing lever length

In addition to the 9th battery cell, the transmission ratio can be adjusted to maintain a near-optimal nut velocity throughout the thrust phase, maximizing the mechanical power delivered by the motor. The optimal nut velocity is approximately $\dot{x}_n = \omega_{opt}/R_3 = -0.161$, whereas the nut travels at a velocity of $-0.2 < \dot{x}_n < -0.178 \text{ m s}^{-1}$. By reducing the lever length, the nut velocity can be lowered to approach optimal nut velocities. A 20% reduction of the lever length, from $l_q = 10$ cm to $l_q = 8$ cm, is expected to reduce the minimum nut velocity by approximately the same percentage, since the lever length is approximately proportional to the transmission ratio. It is thus expected that this change leads to a nut velocity of $-0.2 < \dot{x}_n < -0.143 \text{ m s}^{-1}$ throughout the thrust phase, for which the nut passes its optimal velocity twice. Simultaneously, the increased lever length is expected to improve steering and

^[iv]This result corresponds to the performance of Skippy analysed in section 4-4-2, including ball screw friction, with a linear spring

balancing, since it effectively allows for faster actuation velocity. This can also be reflected from the increased spring stiffness, which is recalculated accordingly to match the initial increased force (ca. +20%) and energy requirements (identical).

The implementation of the new lever length leads to an increment of the jumping height to 4.10 m, with a minimum absolute nut velocity of 0.145 m s^{-1} , close to the estimated 0.143 m s^{-1} . Results of force and velocity curves are plotted in figure 4-12. The improved jumping height appears mainly to be attributed to the fact that the voltage saturation dominated the thrust phase for longer, before the nut velocity saturation starts dominating the thrust torque, reducing the maximum available voltage. The effect of a changing nut velocity during voltage saturation is minor. The effect can be seen in figure 4-10a, which shows the mechanical power delivered by the RE35 motor for $l_q = 10 \text{ cm}$ and $l_q = 8 \text{ cm}$. It can be seen that the power input is nearly at its maximum of approximately 380 W in the beginning. During this phase, the thrust phase is dominated by voltage saturation, where $U = U_{max} = 30 \text{ V}$. When the nut reaches its maximum permissible absolute velocity of $\dot{x}_{n,max} = -0.2 \text{ m s}^{-1}$, the mechanical power input decreases. The power decrease follows from the fact that the spring is unloading and the nut is travelling with constant speed. An unloading spring corresponds to a dropping spring force. The spring force equals the nut force. The product of this declining force and constant velocity equals power, which must therefore be dropping as well.

The power reduction is reflected by the fact that the nut velocity saturation now dominates the torque command, from which follows that $|U| \leq U_{max}$ instead of $U = U_{max}$. A reduced voltage input reduces the mechanical power generated by the motor. The maximum power output scales approximately quadratically with voltage, as evaluated in equation (4-18). In contrast, maximum power as function of nut velocity is a parabola, meaning that small changes from the optimal velocity cause very little change to the maximum output power. Figure 4-11a further elucidates the principle. The figure shows constant power lines as function of nut velocity and voltage, for the upper half of the voltage and velocity range on square axes. The horizontally stacked curves show the major dependency of power on voltage. As such, it can be concluded that it is most important to focus on maintaining maximum voltage rather than maintaining an optimum nut velocity, as long as the nut velocity does not deviate too much from its optimal value.

We have furthermore reduced the initial nut velocity to $\dot{x}_{n,0} = \dot{\phi}_{h,0}/R_3 = \omega_{opt}/R_3$ which was considered to be a more realistic assumption than $\dot{x}_{n,0} = -0.2 \text{ m s}^{-1}$ given that we do not aim for Skippy to be dominated by nut velocity saturation by the end of the landing phase. Because of this minor change, nut velocity saturation kicks in slightly earlier which slightly reduces the jumping height, from 4.10 m to 4.08 m.

It could be considered to reduce the lever length even further, to $l_q = 7 \text{ cm}$, extending the duration of maximum voltage domination even more. However, there are various reasons not to go down this route. With the current system parameters, Skippy's performance regarding jumping height is satisfactory. Lowering the lever length also has disadvantages. Foremost, the lever length is approximately inversely proportional to the axial force on the spindle, which is limited to approximately $F_{n,max} = 3.1 \text{ kN}$. It can be observed from figure 4-12 that the forces are higher. Moreover, decreasing the lever length increases the torque:velocity ratio of the motor, which decreases the motor efficiency and subsequently leads to more heat generation. Lastly, nut velocities that are more suboptimal from the optimal velocity reduce the maximum power output more significantly. In fact, a re-run of the simulation for $l_q = 7 \text{ cm}$

has shown an improvement of mere 3 cm jumping height.

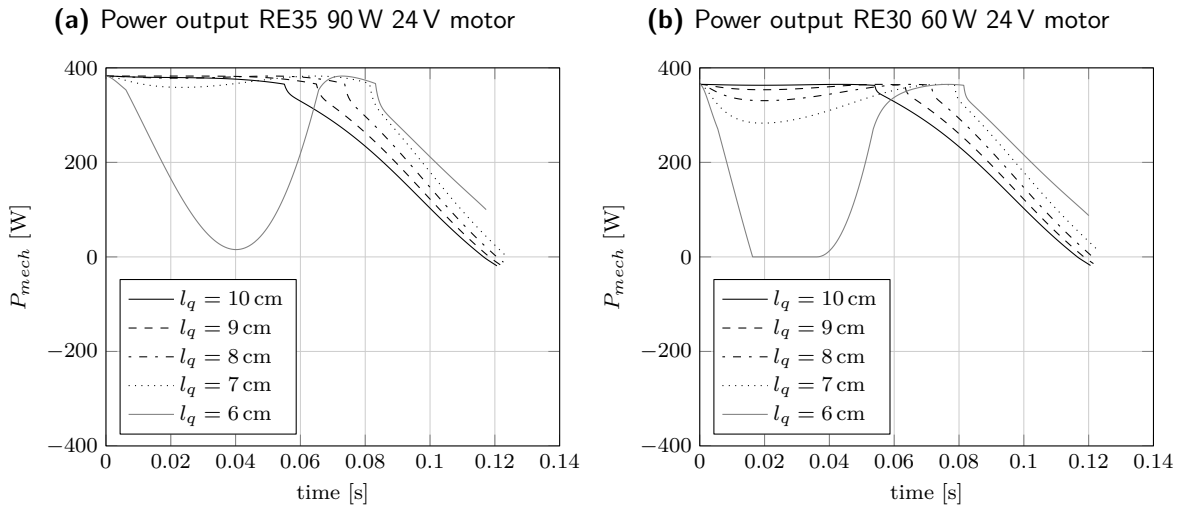


Figure 4-10: Mechanical power output of both motors, compared for various lever lengths l_q ($\dot{x}_{n,0} = \omega_{opt}/R_3$, $U = 30$ V). Shorter lever lengths permit the battery to deliver maximum voltage for a longer time. In general, this increases work done by the motor as long as lever lengths are not too short ($l_q \gtrsim 7$ cm), for which the required motor velocity becomes significantly suboptimal.

RE30 motor

It may be worth considering to step down to the lighter RE30 60 W 24 V motor instead of the RE35 90 W 24 V motor, for it has some interesting advantages and it is able to deliver an almost identical amount of mechanical power as the RE35 model, at 365 W versus 380 W. Its rotor inertia is almost twice as low as the RE35 model, which is convenient for balancing and steering. This was already noted in section 3-2-3 and section 4-3. Moreover, at 260 g it is 80 g lighter than the RE35. A disadvantage is that its thermal time constant of the winding at 16.3 s is almost twice as low, which means that it overheats quicker. Although, since hopping up to 4 m only takes a couple of seconds, this should not be a major problem. Secondly, the absolute nut velocity for maximum power output is 0.183 m s^{-1} at 30 V, which is approximately 0.02 m s^{-1} faster than the 0.161 m s^{-1} optimum speed of the RE35 motor. This seems like an advantage at first, because this value is closer to maximum nut velocity, allowing us to travel more efficiently at maximum nut velocity. However, we have also seen that maximum power is mainly determined by maintaining maximum voltage, for which we desire a relative short lever length. A lever length shorter than $l_q \approx 10$ cm will cause the motor to run primarily below its optimum speed. The effect on mechanical power output is not a problem for small velocity deviations, but is already obvious for a lever length of $l_q = 8$ cm, as can be seen in figure 4-10b. Moreover, it is generally preferred to run the motor at speeds slightly above ω_{opt} than speeds below ω_{opt} , because the motor's efficiency deteriorates for higher torque/velocity ratios, causing even more heat generation. Nevertheless, Skippy is still able to jump up to a height of 4.01 m and 3.94 m for $l_q = 8$ cm and $l_q = 10$ cm respectively, making this motor a viable candidate as alternative for the RE35, not to be excluded from

(a) Power output is most sensitive to U than \dot{x}_n (b) Current does not exceed maximum (ca. 40 A)

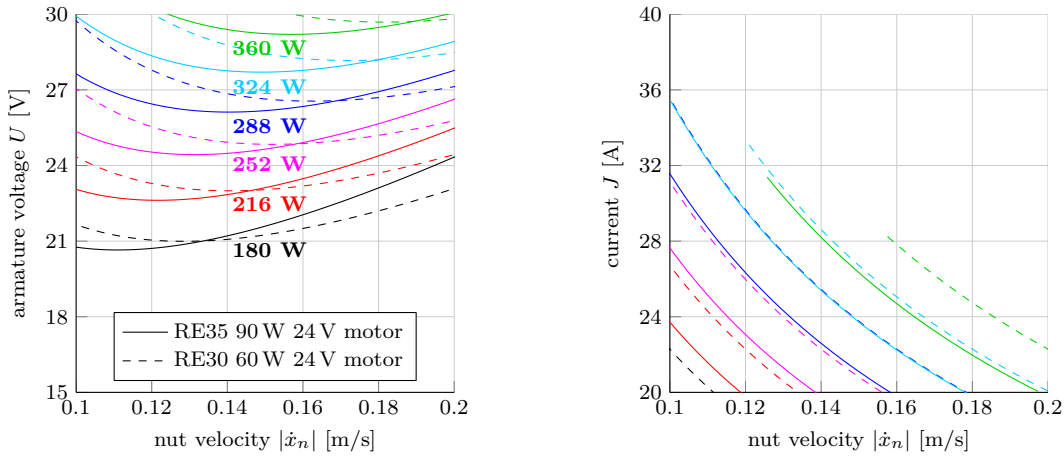


Figure 4-11: Armature voltage and current as function of nut velocity for constant mechanical power outputs. Axes have equal length and plot upper half ranges of velocity, voltage and current to compare their effect on the power output. The nearly horizontal curves in figure 4-11a indicate that—for our range of nut velocities—power output is mainly dominated by voltage rather than velocity. Currents do not exceed the maximum of 40 A.

considerations.

4-6-2 Improving energy conversion

This section focusses on improving energy conversion to useful kinetic energy during the thrust phase, which should increase the jumping height. When rerunning the simulations with the new system with the bilinear spring, we observe similar behaviour as in section 4-5-4: suboptimal unloading of both springs. The reduced lever length has somewhat improved the results, but not sufficiently so. A shorter lever results in shorter nut travel, which is favourable for reduced spring length. However, the fact that a shorter lever also redefines the spring to be stiffer and thus even shorter, negates part of this effect. We require a change in transmission ratio that does not or barely affect the initial transmission ratio, but increases the transmission towards and during the steering phase. This results in reduced spring travel without having to increase the nut velocity and without affecting the spring profile. Moreover, the increased transmission ratio allows for quicker knee joint motions during the steering phase and thence improves steering capabilities.

The variable transmission ratio can be tuned properly in a future version of the dynamical model, where the complete four-bar linkage is modelled. However, we can already obtain an incremental transmission ratio by reducing the lever angle ϕ_q . This angle is currently $\phi_q = \frac{\pi}{8}$ rad = 22.5° , which we can reduce to approximately $\phi_q = 10^\circ$. For lower lever angles, upright balancing would no longer be possible because the linkage would cross a singularity. In addition to a change of lever angle, it was decided to reduce the lever length to $l_q = 7$ cm, as this further improved the steering capabilities and energy transfer efficiencies. The transmission ratio R_2 for $\phi_q = 10^\circ$ and $l_q = 7$ cm is displayed in figure 4-13, compared to old transmission ratios. Note that the new R_2 nearly crosses zero at $\phi_{knee} = 2q_6 = 180^\circ$. Steering

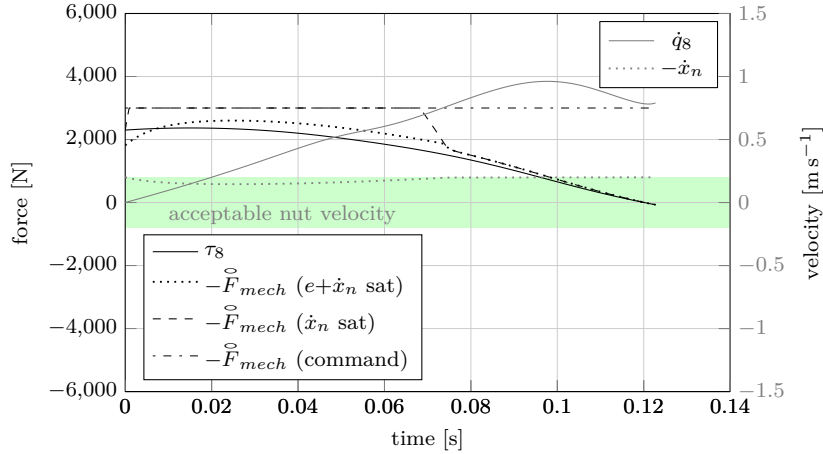


Figure 4-12: Force and velocity profiles with new lever length $l_q = 8$ cm. The thrust phase is longer dominated by voltage saturation; the nut does not reach its maximum permissible absolute velocity of $\dot{x}_{n,max} = 0.2 \text{ m s}^{-1}$ until $t \approx 0.7$ s. This improves jumping performance.

occurs for $45 \lesssim q_6 \lesssim 60^\circ$, for which the transmission ratio is reduced by approximately 20% to 40%.

4-6-3 Results

The spring profile is defined by $k_1 = 150 \text{ kN m}^{-1}$, $l_{bi} = 5.0 \text{ mm}$ and $k_2 = 52.22 \text{ kN m}^{-1}$ ^[v] according to equation (4-17). The results of the simulations are plotted in figures 4-13 to 4-16 for the RE35 90 W 24 V Maxon motor, $\phi_q = 10^\circ$ and $l_q = 7$ cm.

We have performed identical simplified steering tests as in section 4-5-1, where the nut is sent to its minimum and maximum velocity for $t > 0.1$ s. From figure 4-14a can be observed that steering has much improved with respect to figure 4-6a. Similarly to what was observed in figure 4-6b, slipping only occurs in the very last milliseconds before lift-off. Moreover, from figures 4-15 and 4-16 is observed that both springs are approximately unloaded at lift-off, indicating that energy is transferred more efficiently to kinetic energy. For this particular motion, where Skippy is performing positive steering, Skippy is able to jump up to a height of 4.16 m. For negative steering, energy conversion is less efficient and Skippy jumps up to 3.96 m. We still observe that Skippy lifts off earlier for negative steering, which reduces steering capabilities. Other steering strategies can possibly improve upon this undesired behaviour. Lower hops likely also allow for better steering actions, since more time will be available and $|\dot{q}_8|$ will likely not be much higher (if even higher) than $|\dot{x}_n|$. Introducing the full four-bar linkage (to replace the knee joints that are geared 1:1) increases possibilities to change the transmission ratio, which could be optimized to further improve steering capabilities. However, it is expected that introducing the four-bar linkage does not further increase the jumping height, because energy transfer during the launch phase is already nearly optimal, as can be observed in figure 4-16. The implementation of the four-bar linkage is furthermore

^[v]Note that k_2 is higher than previously due to the reduced transmission ratio, even higher than the previous value of the linear spring stiffness 32.15 kN m^{-1}

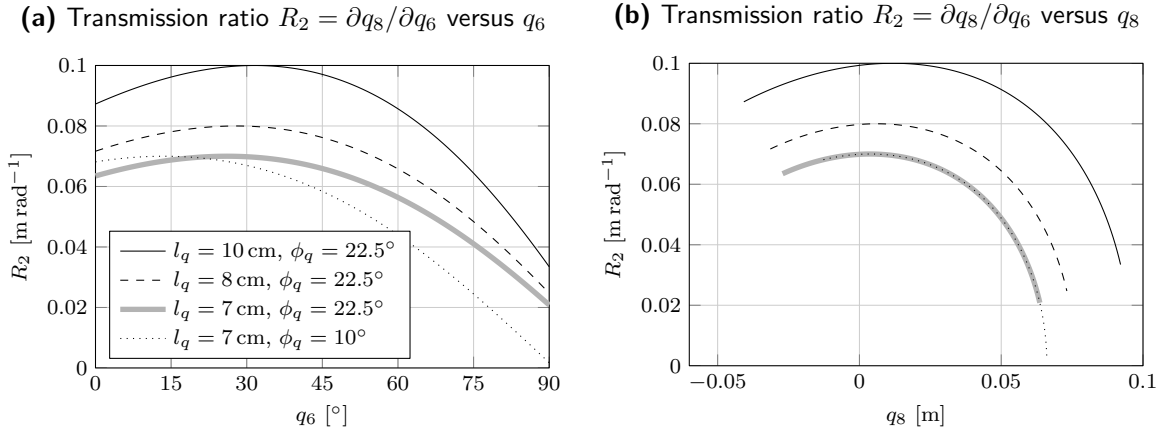


Figure 4-13: Lever transmission ratio R_2 as function of q_6 and q_8 for various lever parameters. The new (dotted) transmission ratio is more regressive towards and during the steering phase, which improves steering capabilities for which we require fast but low-torque motions.

not included in this work, because its many parameters require an optimization that does not fit within the scope of the thesis.

Because of the reduced lever length, the thrust phase is now dominated by maximum voltage for more than 80%. During part of the thrust phase, the shaft velocity is lower than the optimal velocity ω_{opt} , but still sufficiently close to the optimal velocity to do more than 96% of maximum mechanical power that the motor can deliver. Shortly after $t = 0.8$ s, the thrust phase is dominated by nut velocity saturation, causing a voltage drop and thus a significant drop of mechanical power delivered to the system (approximately 60% of maximum power at $t = 0.1$ s).

The disadvantage of running at velocities lower than the optimal velocity is more heat generation, which can be observed in figure 4-16. For the RE30 60 W 24 V motor, heat generation is even worse due to higher no-load speeds. For the RE30 motor, it is recommended not to step much lower than $l_q = 8$ cm, although this significantly reduces the jumping height below 4 m. The RE30 however is slightly better at steering due to its lower rotor inertia. Nevertheless, since steering capabilities are mainly limited by limited nut velocities, rotor inertia should not be a main design criterion.

The reduced lever length has further increased the axial force on the nut, up to $\tau_8 \approx 2.5$ kN. This is still below the maximum permissible absolute of $F_{n,max} = 3.1$ kN. For positive steering, the torque on the coupling is still $|T_{out}| < 1$ N m.

Simulations have also been done for different spring parameters k_1 and l_{bi} . In general, we found that more weakening spring profiles (i.e. higher k_1 , higher l_{bi}) slightly deteriorate energy conversion (i.e. springs are not fully unloaded at lift-off). Nevertheless, jumping up to 4 m and decent steering can be achieved for a variety of spring profiles.

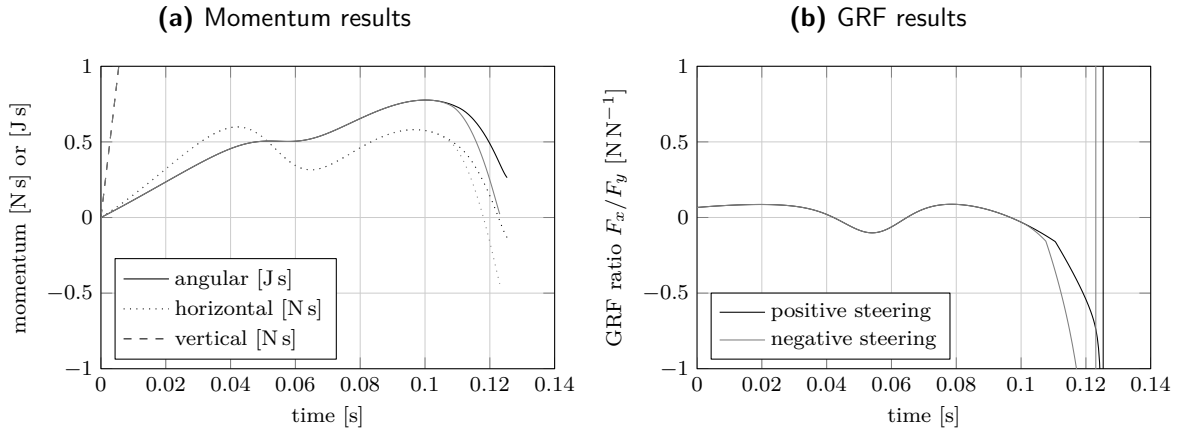


Figure 4-14: Momentum and GRF results for minimum (grey) and maximum (black) steering actions with a linear spring. Steering abilities have improved with respect to figure 4-6a, as the differences between horizontal and angular momenta for both steering actions differ more at lift-off. The robot does tend to slip just before lift-off, as the GRF ratio falls below -1 .

Concluding remarks

In this chapter we have initially fed a simple feedforward torque to the motor, after which we have dealt with various physical limitations of the systems by dynamically saturating that torque command. It was found that the action strategy for thrusting and steering is mainly—if not completely—dominated by these physical limitations. It confirms that attempting to reproduce an imposed master curve on the quadriceps force τ_8 such as attempted in chapter 3 is an inefficacious approach. Instead, the new action strategy allows us to be on the edge of what the system is physically capable of delivering. Moreover, the strategy improves our understanding of the system behaviour and limitations. The obtained knowledge is used to further improve physical parameters of the system.

We found that the thrust phase is primarily dominated by available voltage, and secondarily by limited nut velocity. Maximum mechanical power can be pumped into the system if we can send maximum voltage through the motor for as long as possible while the motor is rotating at *roughly* the optimal velocity $\omega_{opt} \sim \omega_z/2$. Focussing on maximum voltage is thus key. Elongating maximum voltage input can be realized by preventing the nut velocity saturation to be activated, which can be done by preventing the nut to approach a velocity of $|\dot{x}_n| = 0.2 \text{ m s}^{-1}$. This is realized by reducing the lever length l_q , effectively reducing the transmission ratio. By doing so, one has to be cautious for additional heat generation and increased axial forces on the driveline.

The steering phase is primarily dominated by limited nut velocity. Aggressive steering actions are furthermore dominated by current. Steering capabilities have severely deteriorated since the introduction of the essential series elastic driveline. However, lowering the transmission ratio towards and during steering and increasing the spring stiffness have shown improvements to steering. The full four-bar linkage can likely be used to further improve steering capabilities, but not the jumping height.

Having achieved sufficient jumping heights and decent steering actions for a variety of spring

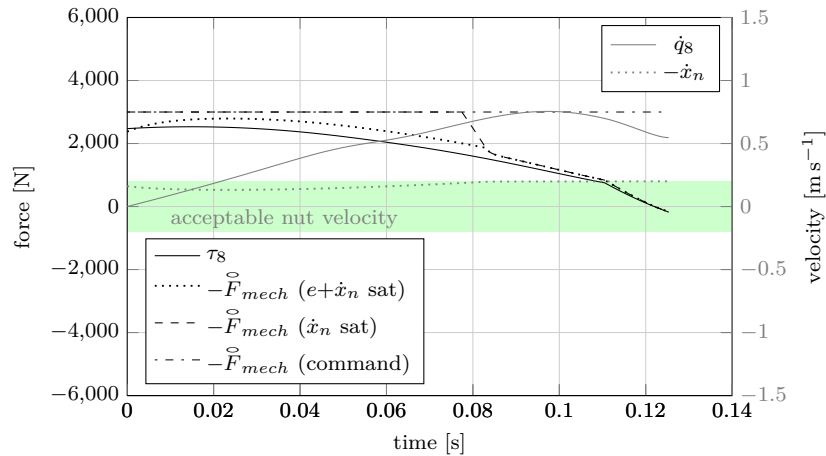


Figure 4-15: Forces and velocities for system with bilinear spring, $l_q = 7$ cm and $\phi_q = 10^\circ$. The performance has improved with respect to figure 4-8 as contact with the ground is prolonged and springs are nearly unloaded at lift-off.

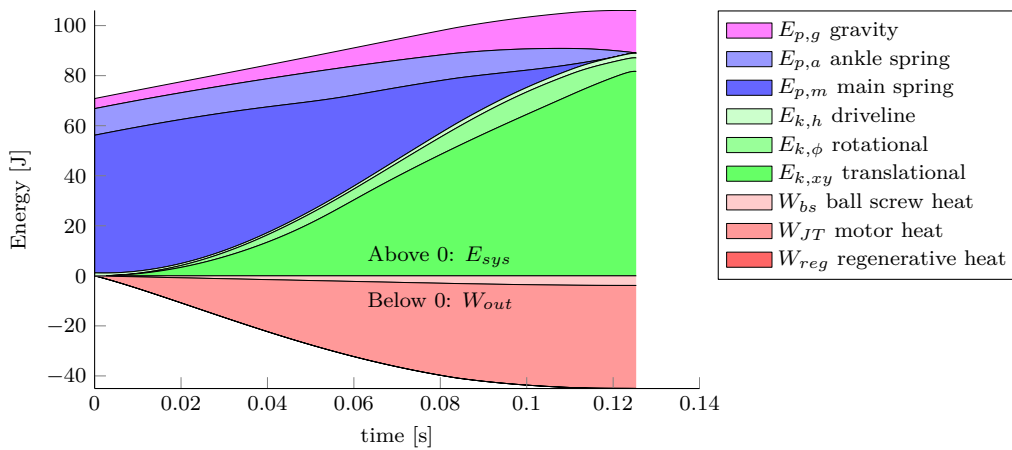


Figure 4-16: Energy flows for system with bilinear spring, $l_q = 7$ cm and $\phi_q = 10^\circ$. Energy is very efficiently converted to translational kinetic energy required for jumping.

profiles, system parameters and a viable action strategy, it seems appropriate to analyse some of Skippy's required additional abilities towards achieving 4 m high hops. These abilities could further reduce the scope of our solutions. Chapter 5 focusses on Skippy's physical ability to balance, which is one of these abilities.

Chapter 5

Balancing

In chapter 4 it was shown that it is possible to achieve 4 m high hops with a reasonably realistic planar model of Skippy. However, only the thrust phase of the final hop was considered. In order to get to the beginning of that thrust phase, Skippy has to be able to do the following, ordered in reverse chronology:

- The landing phase of the final stance phase.
- In-flight control
- Hopping higher from low hops
- Low hopping from standstill (a balanced configuration)
- Balancing
- Standing up from a crash

In the scope of the thesis project it is not feasible to focus on all these phases. The action that is most different from making high jumps is balancing. Without the ability to balance, there is no ability to hop. As such, Skippy's ability to balance is investigated in this chapter.

The focus of this project is not on control, but on design. We are interested in the effect of various system parameters on Skippy's physical balancing performance. These parameters include inertial and dimensional properties, but also spring stiffness. For a future optimization of Skippy's parameters, it is important to know how these properties affect Skippy's balancing performance. Section 5-1 investigates Skippy's physical ability to balance through a measure called the velocity. Section 5-2 uses a simple controller to verify predictions about the balancing performance by the velocity gain. As a result, a closer analysis of Skippy's mass distribution is done in section 5-3 with a successful attempt to improve Skippy's balancing performance.

5-1 Physical ability to balance

Previous studies have shown that a robot with a morphology similar to that of Skippy is able to balance in 3D with 2 actuators (Azad, 2014). The study also shows that the 2D

balancing problem can be decoupled from the 3D system. The 2D morphology of Skippy is very similar to that of acrobot (Berkemeier and Fearing, 1998), which is a system that has been studied thoroughly in the field of control engineering. The studies however do not consider our unique finite stiffness driveline with limited nut velocity, nor do they use the inertial and dimensional parameters that we use for Skippy. In fact, none of the control studies investigate the physical ability of acrobot-like robots to balance. A recent study has addressed exactly this topic, and introduced a measure called the velocity gain (Featherstone, 2012a). This section investigates Skippy's ability to balance with that measure, which has not yet been used by anyone for actual robot design. Section 5-1-1 defines the velocity gain and section 5-1-2 describes simplified robot models for which the velocity gain is calculated and analysed in section 5-1-3.

5-1-1 Velocity gain

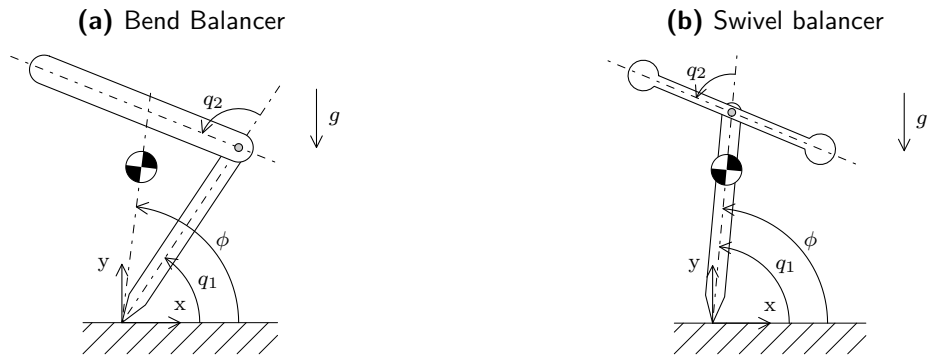


Figure 5-1: Planar models for bend and swivel balancing, consisting of only two bodies each. The swivel balancer uses a frontal view of 3D Skippy, where the leg and torso are considered as a single body, effectively locking the knee.

This section describes the velocity gain, a measure to analyse the physical ability of a robot to balance. Referring to figure 5-1, the velocity gain G_ω is a ratio of the movement of the robot's COM to the movement of the actuated joint that causes the movement (Featherstone, 2012a). More specifically, the velocity gain G_ω is defined as the ratio between the instantaneous change of $\dot{\phi}$ and an instantaneous change of the actuated joint \dot{q}_a , induced by an impulsive torque applied at q_a :

$$G_\omega(\mathbf{q}) := \lim_{\Delta q_a \rightarrow 0} \frac{\Delta \phi(\mathbf{q})}{\Delta q_a} = \frac{\Delta \dot{\phi}(\mathbf{q})}{\Delta \dot{q}_a} \quad (5-1)$$

Identical to the definition of ϕ in section 2-2-2, ϕ is the angle of the vector passing through the COM and the ground contact point (toe) with respect to the ground, where the ground is defined as being perpendicular to gravity. ϕ is the angle that is to be controlled to maintain balance. The robot has a balanced configuration if $\phi = \frac{\pi}{2}$ rad.

G_ω is a function of the robot's configuration described by joint coordinates \mathbf{q} . It is important to note that the velocity gain is a dimensionless quantity, which means that it is invariant to an equal scaling of all system dimensions and mass. A robot is a better balancer if $|G_\omega|$ is higher for all sets of joint coordinates for which the robot is supposed to be able to balance. If $G_\omega(\mathbf{q}_z) = 0$, the robot is physically unable to balance at the configuration described by $\mathbf{q} = \mathbf{q}_z$.

5-1-2 Simplified balancing model

Skippy's velocity gain can be readily calculated using equation (5-1). Higher values of the velocity gain suggest better balancing abilities. However, to obtain a better indication of how good of a balancer Skippy is, we compare Skippy's velocity gain with the velocity gains of two other (acrobot-like) double-pendulum models. This section provides a description of these models and a simplified model of Skippy.

Two simple double-pendulum robot models are introduced. The first model consists of two identical slender beams, which is a well-known model in the field of control theory and dynamics. The second model is optimized to have a high velocity gain, and is referred to as 'good balancer' (Featherstone, 2012a). To ease comparison of the balancing performance, we simplify our model of planar Skippy to also be a double-pendulum robot, depicted in figure 5-1a. The new leg parameters are those of the old leg and foot combined. The lever with two 1:1 geared joints are replaced by a single revolute joint, which connects the torso to the leg. This joint is the knee joint with joint position q_2 . The new joint position is the knee angle $q_2 = \phi_{knee}$ and thus corresponds to the sum of the upper and lower knee joint positions q_5 and q_6 as defined in section 2-2-1. All models now have a morphology identical to that of acrobot, albeit with different ratios of inertial and dimensional properties. Recall that the velocity gain is a dimensionless quantity, which means that it is invariant to a uniform scaling of all systems dimensions and masses. Table 5-1 lists a set of inertial and dimensional parameters of the robots that are scaled so as to match the leg length and mass of the simplified Skippy model. This scaling is applied as it allows for better comparisons between the various robot parameters for the reader.

| | i | joint | body | m_i [kg] | I_i [kg m ²] | $c_{i,x}$ [m] | $l_i = r_{i+1,x}$ [m] |
|--------------------------|-----|-------|-------|------------|----------------------------|---------------|-----------------------|
| simplified planar Skippy | 1 | toe | leg | 0.2 | $m_1 0.19^2$ | 0.24 | 0.55 |
| | 2 | knee | torso | 1.8 | $m_2 0.17^2$ | 0.38 | – |
| good balancer | 1 | toe | leg | 1.63 | $m_1 0.12^2$ | 0.24 | 0.55 |
| | 2 | knee | torso | 0.37 | $m_2 0.27^2$ | 0.60 | – |
| slender beams | 1 | toe | leg | 1 | $m_1 0.16^2$ | 0.275 | 0.55 |
| | 2 | knee | torso | 1 | $m_2 0.16^2$ | 0.275 | – |

Table 5-1: Parameters of a simplified two-link model of Skippy, compared with parameters of two other acrobot-like balancers with identical morphology. Note that l_2 has no effect on the dynamics, and is therefore not listed.

The models each have two joint position variables $\mathbf{q} = [q_1 \ q_2]^T$, where only q_2 is actuated, i.e. $q_a = q_2$. q_1 is the angle of the lower link (leg) with respect to the ground and q_2 is the angle of the upper link (torso) with respect to the lower link.

We will not elaborate on the derivation of the velocity gain for acrobot-like robots. It can be found in (Featherstone, 2012a). However, it is to be noted that for the particular case of inverted double pendula, G_ω is only a function of q_2 , i.e. $G_\omega(\mathbf{q}) \rightarrow G_\omega(q_2)$.

5-1-3 Results

Figure 5-2 plots G_ω as function of q_2 for the simplified version of Skippy, the two slender beams and the good balancer. $q_2 = 0^\circ$ and $q_2 = 180^\circ$ correspond to a fully stretched and

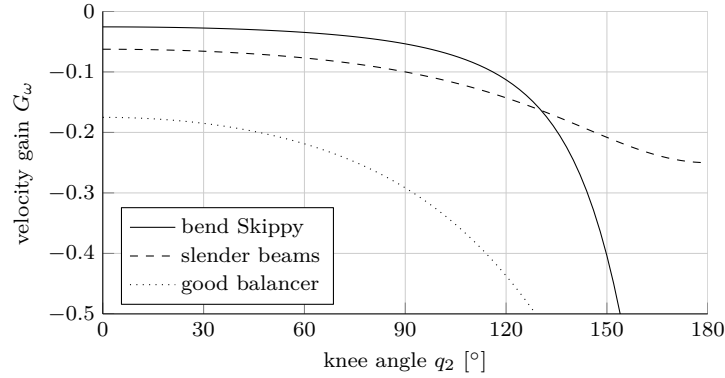


Figure 5-2: Skippy is a poor balancer: its velocity gain is inferior compared to that of other double-pendulum robots.

bent configuration respectively. We observe that Skippy is a very poor balancer for stretched configurations, as its velocity gain at $|G_\omega(0^\circ)| = 0.0256 \approx 1/40$ is extremely low. The slender beams and good balancer models have a velocity gain that is much better, at $|G_\omega(0^\circ)| = \frac{1}{16}$ (exact) and $|G_\omega(0^\circ)| \approx 1/5.71$ respectively.

If we could linearise the system at the given configuration, Skippy's velocity gain implies that a CoM offset of $\Delta\phi = 0.1^\circ$ CoM requires *at least* a $\Delta q_2 = 0.1 \times 40^\circ = 4^\circ$ correction by the actuated joint. In reality, the total angular travel of the actuated joint $\int |\dot{q}_2| dt$ exceeds 4° because the robot cannot move infinitely quickly. During the corrective motion, gravity acts on the robot which worsens the balance error. Simulation studies of other robots have shown that the overshoot is two to three times the minimum travel predicted by the velocity gain (Featherstone, 2015b), which indicates that a correction of approximately $8^\circ < \Delta q_2 < 12^\circ$ is required.

The approximation of above linearisation becomes worse for larger CoM offsets, for which we have to consider the maximum balancing range of the knee joint. The balancing range of the knee is smaller than the total knee range of 180° . The swivel bar would hit the ground if the knee is fully bent, and balancing in an upright configuration is extremely hard as the swivel axis (even though slightly tilted with respect to the torso) would be nearly perpendicular to the ground. We assume a balancing range of $15^\circ \lesssim \phi_{knee} \lesssim 165^\circ$, approximately 150° in total.

Eventually, we are interested in the ability of Skippy to correct for a $\Delta\phi = 5^\circ$ CoM offset, as mentioned in section 2-1-5. The linearisation suggests that this offset requires the actuated joint to move by at least 200° , with an expected rotation that is approximately two to three times this amount: $400^\circ < \Delta q_2 < 600^\circ$. The resulting numbers exceed the knee's total range of maximum 150° by far. However, this approximation assumes the velocity gain to be equal for all its states, which is not the case. In fact, the velocity gain improves for flexed (crouched) configurations, as can be seen in figure 5-2. This suggests that the expected required rotation reduces *if* Skippy is doing its corrective motion by crouching, which is the case if $\phi > \frac{\pi}{2}$ (Skippy leans forwards).

5-2 Verification

It is not our goal to devise a stable balancing controller in this thesis. Nonetheless, it is desired to verify the abstract predictions by the velocity gain in section 5-1-3 about Skippy's poor balancing performance. As such, we apply a readily available balancing controller for verification. The results are discussed in section 5-2-3, using the controller selected in section 5-2-1 including electrical saturation as described in section 5-2-2.

5-2-1 Controller selection

The goal of this section is to find a readily available controller that can be used to verify the predictions of the velocity gain through dynamic simulations. Many balancing controllers have been developed for acrobot-like robots. A selection of prominent controllers have been compared by Azad (Azad, 2014), these include Spong's LQR controller (Spong, 1995), Berkemeier and Fearing's linear controller (Berkemeier and Fearing, 1999), Grizzle's non-linear controller (Grizzle et al., 2005) and Azad's non-linear angular momentum-based controller. The comparison concluded that the latter offers the overall best performance, regarding overshoot and settling time. A disadvantage of the controller is that it relies on symbolic derivations of the system's dynamics. Featherstone has expanded on the momentum-based control principle that Azad uses, and has devised a controller that does not require symbolic derivations (Featherstone, 2015a), for which it is more broadly applicable to robotic systems with larger kinematic trees.

We use Featherstone's controller to analyse Skippy's balancing behaviour, for the following reasons:

- **availability** The controller has specifically been tested for acrobot-like models and has proven its successfulness for (at least) this model, and for the type of reference signals that we are interested in (step functions). Correspondingly, previously derived equations can directly be applied to our models.
- **performance** Featherstone's controller is extremely similar to Azad's controller, which has been shown to outperform many other controllers that were developed for or tested on acrobot.
- **generality** Despite the fact that the control system is readily available, the controller does not require symbolic derivation and is generalizable for other planar balancing robots. The controller consists of a simple linear control law that acts on a plant described by state variables $[q_2 \ L \ \dot{L} \ \ddot{L}]^T$ and two (variable) parameters $\mathbf{Y} = [Y_1 \ Y_2]^T$. L is the angular momentum of the robot about its support. For a constant reference signal $q_r : q_2 = \text{constant}$, the control law equals

$$\ddot{L} = k_1 \ddot{L} + k_2 \dot{L} + k_3 L + k_4 (q_2 - q_{2,r}) \quad (5-2)$$

in which k_1 to k_4 are gains. The controller as a whole is non-linear, because L (and its derivatives) and \mathbf{Y} are non-linearly related to the system's state variables $[q_1 \ q_2 \ \dot{q}_1 \ \dot{q}_2]^T$. However, general expressions for L and \mathbf{Y} can be derived that take elements of the readily available joint-space inertia matrix $\mathbf{H}^{[i]}$ as input, which also holds for systems

^[i]For a definition of the joint-space inertia matrix \mathbf{H} , see appendix B (Featherstone, 2008c)

with larger kinematic trees. This has two advantages. Firstly, since \mathbf{H} and the system's state variables are available numerically, we do not require any symbolic derivation to obtain the controller equations. Secondly, the controller does not lose its generality, for it is able to be applied to different balancing systems with different inertia matrices \mathbf{H} . In fact, it is expandable to models of Skippy with more complicated kinematic trees, which is desirable for future research.

- **simplicity** The controller is simple for two reasons. One reason is the above discussed generality: the plant parameters and state variables are functions of elements of \mathbf{H} , and thus automatically adapt to a system with different inertial and dimensional parameters. Moreover, the linear control law allows for easy placement of the poles from the characteristic equation. The four poles are all placed at a single point $-p$ on the negative real axis^[iii], which leaves the user only with one variable (p) to affect the behaviour of the controller. Choosing the variable is a trade-off between settling time and overshoot and is typically done through trial-and-error. A higher value of p corresponds to a faster control action, but leads to more overshoot. The fact that only one variable can be changed by the user simplifies tuning significantly.

5-2-2 Electrical Limitations

In addition to analysing Skippy's balancing performance with the controller selected in section 5-2-1, we implement electrical saturation as described in section 4-2, which can also be used to approximate nut velocity saturation if the motor torque is low. The inclusion of this saturation is used to analyse how driveline limitations further limit Skippy's balancing performance in section 5-2-3.

The controller can be supplemented by electrical saturation without considering the series elastic and inertial driveline. This implies that the motor shaft velocity $\dot{\phi}_h$ is proportional to the knee joint velocity \dot{q}_2 . We expect that the inclusion of driveline inertia has little effect on balancing performance, as it was observed in chapter 4 that the nut can be accelerated from zero velocity to its maximum 0.2 m s^{-1} in a mere 5 ms, whereas the action required for balancing is much slower than thrusting and it is governed by lower forces. Instead, the expected delimiter of balancing performance is the limited nut velocity.

To implement electrical saturation, it is required to consider the transmission ratio R_{123} between $\dot{\phi}_h$ and \dot{q}_2 . The corresponding control diagram is depicted in figure 5-3. The controller takes the system state variables $(\mathbf{q}, \dot{\mathbf{q}})$ and a reference angle $q_r : q_2$ as input. The controller outputs the torque command for the knee $\tau_{2,c}$, which is fed to the electrical saturation after it has been converted to the motor torque command $T_{mech,c}$. The dynamic saturation is a function of the shaft velocity $\dot{\phi}_h$ and outputs the motor torque T_{mech} .

R_{123} is an estimation of the real transmission ratio between the knee joint and shaft. It consists of three components:

$$R_{123} = R_1 R_2 R_3$$

R_1 , R_2 and R_3 have been defined earlier in the thesis. R_1 is the transmission ratio between the upper knee and knee and R_3 is the gear ratio between the motor shaft and nut. They are

^[iii]Placing all poles of a characteristic equation in a single point is a reasonable approach, since mainly the pole closest to zero affects the speed of the controller.

given by equations (2-1) and (3-18) respectively and are both constant. R_2 is the gear ratio between the nut and upper knee joint, given by equation (3-14), which is not a constant. To simplify the control problem, we approximate R_2 with a constant as well, such that R_{123} is constant. The range of the upper knee joint is approximately $\frac{\pi}{2}$ rad, which corresponds to a linear quadriceps travel of $\sqrt{2}l_q$. We use their ratio to redefine R_2 as a constant:

$$R_2 = \frac{\sqrt{2}l_q}{\pi/2} = \frac{2\sqrt{2}l_q}{\pi}$$

For $l_q = 7$ cm, $R_2 = 0.063$ m rad⁻¹, which corresponds approximately to the average of R_2 in figure 4-13 (for $l_q = 7$ cm). R_{123} becomes

$$R_{123} = -\frac{2\sqrt{2}l_q}{n} \approx -99$$

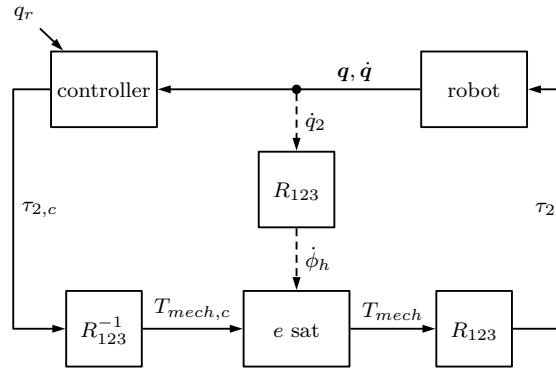


Figure 5-3: Block diagram of a robot (e.g. Skippy) and balancing controller with electrical saturation. Dashed lines correspond to inputs that affect the dynamic saturation limits.

5-2-3 Results

This section presents and discusses simulation results of the balancing performance of the double-pendulum robots. The corresponding simulations are done using the controller selected in section 5-2-1 including electrical saturation described in section 5-2-2.

The task of the controller is to recover from a 5° balancing offset. We compare results of Skippy with that of a good balancer and slender beams. The initial knee angle is chosen at $q_{2,0} = 15^\circ$. This is close to the stretched configuration $q_2 = 0^\circ$. The initial leg angle $q_{1,0}$ is chosen such that the COM angle ϕ initially equals $\phi_0 = 180^\circ + 5^\circ$. The initial velocity is set to $[\dot{q}_{1,0} \ \dot{q}_{2,0}]^T = [0 \ 0]^T$. The reference signal q_r , which is the desired knee angle that is fed to the controller, is set to a constant $q_r : q_2 = 90^\circ$. This balancing position is approximately in the middle of the knee's ROM.

The simulation experiments are performed with a maximum voltage of $U_{max} = 30$ V on Skippy and the two other double-pendulum robot models (see figure 5-1a and table 5-1), by using the block diagram in figure 5-3. In addition, we perform an experiment with the model of Skippy and a maximum voltage of

$$U_{max} = 18 \text{ V} \approx \frac{U_N}{\omega_{z,N}} \dot{x}_{n,max} R_3$$

where $U_N = 24 \text{ V}$ is the nominal voltage and $\omega_{z,N}$ the no-load rotational velocity corresponding to the nominal voltage. Assuming that the actuation torque is relatively low, setting the maximum voltage is a simple yet reasonable method to indirectly implement nut velocity saturation.

The pole p has been selected through trial-and-error to minimize the overshoot. Typically, a lower pole decreases the overshoot, but it slows down the balancing. However, if the system is too slow, secondary effects such as gravity dominate system behaviour. This could increase the overshoot or causes the system to go unstable.

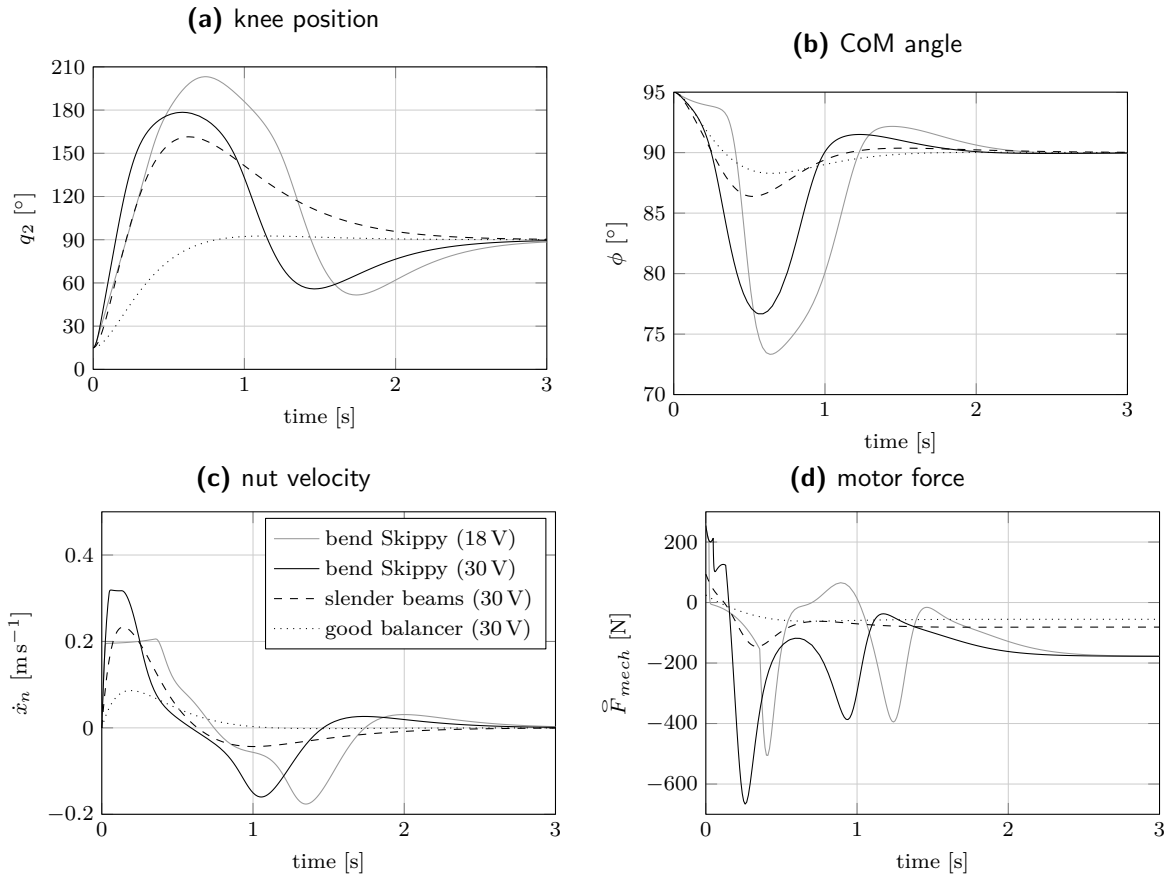


Figure 5-4: Results of balancing simulations of various robotic models. Skippy is a poor balancer; its settling time and overshoot are higher.

All results are plotted in figure 5-4. From figures 5-4a and 5-4b, we observe that Skippy is indeed a poor balancer, but not as bad as predicted in section 5-1-3. Skippy requires its full range to recover from its initial offset (i.e. $q_2 \rightarrow 180^\circ$), for which the swivel bar would hit the ground in reality, but it *does* recover. We observe that velocity saturation is only active in the very beginning, but it does not cause the controller to go unstable. The effect of voltage saturation on the nut velocity can be seen in figure 5-4c, where the nut velocity $\dot{x}_n = \dot{q}_2 R_1 R_2 = \dot{\phi}_h / R_3$. It can be observed that the reduced voltage performs adequately as a velocity saturator, as the nut velocity is capped at approximately $\dot{x}_n = 0.2 \text{ ms}^{-1}$. This is likely due to the low torques observed in figure 5-4d (where $\hat{F}_{mech} = \tau_2 / R_1 / R_2 = T_{mech} R_3$),

of which the peak value is approximately 10 times lower than the maximum that can be transmitted. Note that the reduced voltage further decreases the performance, so that q_2 reaches $200^\circ > 180^\circ$ (figure 5-4a). This would require Skippy's torso to bend through its leg.

The performance of the slender beams is better, as predicted by the velocity gain, and would be satisfactory for Skippy. The performance of the good balancer is excellent, as we barely observe any overshoot. The result is likely never attainable with Skippy, because it requires a mass distribution that is terrible for jumping.

To validate the neglect of the main spring, we can check what the nut velocity would have to be if it would be driving the joint via a spring with stiffness k . This velocity equals $\dot{x}'_n = \dot{x}_n + (1/k)(\dot{\tau}_2/R_1/R_2)$ (this equation is similar to equation (3-26)). $(1/k)(\dot{\tau}_2/R_1/R_2)$ is a term that adds (or subtracts) a velocity to obtain the required nut velocity to drive the joint, which is found to be negligible for most of the simulation time due to the relatively high stiffness $k = 150 \text{ kN m}^{-1}$ (the stiffness used in section 4-6-3). This supports the validity of neglecting the spring in the driveline.

The main conclusion drawn from the current analysis is that Skippy's balancing performance is insufficient. The balancing performance is improved by changing inertial and dimensional system parameters of Skippy in section 5-3.

5-3 Improving balancing performance

To improve Skippy's balancing performance, a redistribution of its mass is required. Most of Skippy's mass is located towards the swivel bar, which is done for the purpose of jumping as described in section 2-1-4. The good balancer model suggests that much of Skippy's mass should be moved to the leg, but this is not acceptable for jumping. We also cannot accept a big difference between leg and torso length, as explained in section 2-1-4. However, we can attempt to move some of the torso mass towards the knee to improve the balancing performance, and congruently move another part of the torso mass further towards the swivel bar as to prevent deterioration of thrust performance. This effectively increases the radius of gyration (ROG) of the torso—and thus of the whole robot—which should also improve balancing performance.^[iii] This section sketches a more in depth and realistic scenario of mass distribution of Skippy's torso (section 5-3-1) including the swivel bar (section 5-3-2) than has previously been done. Results of the new mass distribution are discussed in section 5-3-3.

5-3-1 Torso Mass distribution

A closer study of the masses in Skippy's torso (including the swivel bar, i.e. 3D Skippy) allows us to analyse the possibilities of changing the mass distribution. Roughly, the mass

^[iii]A robot is in fact unable to balance if it consists of only a single point mass. A physical understanding can be enhanced by considering two extremities of impossible balancers. If a balancer would have all of its mass concentrated at one point (such as at the far end) in the torso, a knee rotation would only cause the mass to move radially with respect to the contact point (and thus vertically if the robot is in a balanced position). If all mass is concentrated at the knee or anywhere in the leg instead, then a rotation of the (inertia-less) torso would not accelerate any mass at all. In both cases, no mass is accelerated horizontally, whereas this is required for balancing.

of the torso is determined by two motors, a ball screw, a swivel bar, batteries, a structural frame and a protective cover. Furthermore, the four-bar linkage has a significant mass due to the various links and axles that have to transfer high forces. The mass of the four-bar linkage is also modelled as part of the torso, because the linkage itself is not modelled. By modelling the main parts of Skippy in SolidWorks, reasonable approximations can be obtained about their mass and their location. The SolidWorks modelling, which has been an ongoing project during the simulation studies, is not further treated in this thesis. Below follow some considerations regarding mass distribution that came forth from this project, starting with the motor housing mass (including motors) and constructional mass (including four-bar linkage, frame and cover) in this section, and swivel bar mass in section 5-3-2.

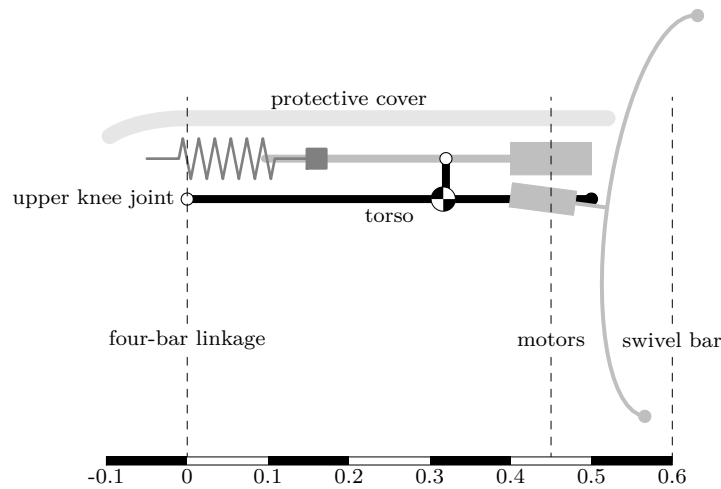


Figure 5-5: Sketch of Skippy's torso with an indication of part placement for determining mass distribution.

Motor housing

The main motor, with a weight of 320 g for the RE35 or 240 g for the RE30, is attached to the back of the motor housing. The axle of the motor housing is located at $\mathbf{M} = [0.32 \ 0.05]$ in Ψ_6 , see figure 3-5, approximately 0.32 m from the upper knee joint (the joint to which the torso is connected). The horizontal distance $x = 0.32$ m in Ψ_6 is also written as $0.32 u_{6,x}$, where $u_{6,x}$ is a vector of unit length along the x axis in the frame Ψ_6 . This distance allows for placement of the spring, lever and the longest spindle from Steinmeyer series 2412-M8 that we could possibly need. The CoM of the main motor and its housing is estimated at $0.4 u_{6,x}$. Moving the motor housing closer to the knee is possible with a shorter ball screw, but the mass cannot be moved closer towards the knee than approximately halfway the torso, which is undesired for increasing the torso's ROG. Instead, the mass can be slightly pushed further towards the tip of the torso to improve jumping performance. Note however that various parts—of which mainly the swivel bar^[iv]—are to be placed behind the motor. These parts add to the length of Skippy's torso, see figure 5-5. With the current motor placement,

^[iv]The swivel bar is assumed to have a bent shape such that it can buckle during an unfortunate landing, for which we reserve approximately 10 cm of torso length.

it is expected that both tips of the swivel bar are approximately located at $0.6 u_{6,x}$ ^[v]. In this scenario, the motor of the swivel bar is placed below the main motor, and cannot be moved closer towards the knee. Moving the motor housing axle further towards the swivel bar likely stretches the total length of the torso beyond 0.6 m, which is to be avoided unless absolutely necessary for Skippy to be able to jump up to a height of 4 m. The mass of the two motors including their housing is budgeted at 0.5 kg.

Constructional mass

The mass of the four-bar linkage itself, with all of its components and axles, is budgeted at 0.3 kg, located at the upper knee joint. The joints, drivelines and electronics are held together with (aluminium) structural elements (also referred to as structural frame). Furthermore, the entire torso is to be covered by foam or an other protective cover, with the purpose of protecting the robot from crashes. The mass of structural elements, protective covers, electronics, the main spring and the ball screw are initially modelled as an evenly distributed mass between $-0.1 u_{6,x}$ and $0.5 u_{6,x}$ with a total mass of 0.6 kg. -0.1 m implies that some of the mass overlaps the knee joint, corresponding to the protective cover. The amount of mass that extends over the knee joint is likely overestimated, since it only includes the protective cover (no structural elements). Nonetheless, it is expected that the protective cover is thicker here, because the knee joint is a ‘corner’ in the robot’s morphology that is more vulnerable and likely to receive more concentrated impacts than the ‘straight’ torso. Future engineering studies ought to provide more accurate answers on the possibilities and requirements of mass distribution of the cover and structural elements. The fact that we assume the distributed mass to extend to no further than $0.5 u_{6,x}$ indicates that the mass does not extend all the way to the end of the torso, which is related to the presence of the swivel bar that protrudes from the torso at this point. The mass of the swivel bar is yet to be decided on, and the batteries are yet to be placed. A closer analysis of swivel bar actuation is done in section 5-3-2.

5-3-2 Swivel actuation

The goal of this section is to analyse the effect of mass in the swivel bar, and to find an acceptable mass for the swivel bar and its motor. The analysis is a result from analysing the mass distribution of Skippy’s torso in section 5-3-1, in which the mass of the swivel bar is integrated. The analysis also affects battery placement and swivel motor selection.

Swivel bar mass

The swivel bar has a non-negligible mass itself. To analyse swivel actuation, we consider a simple out of the plane 2D swivel balance robot, consisting of a flywheel on a stick. This is a simplification of Skippy where we view the frontal (coronal) plane rather than the sagittal plane, featuring the flywheel as the swivel bar and the stick as the rest of Skippy (which implies a locked ankle and knee joint). The system is depicted in figure 5-1b. This robot is a special case of the acrobot, where $c_{2,x} = 0$. The system’s dynamic response is only a function of the torque on the swivel bar, which is important for balancing. The induced swivel bar

^[v]Note that a torso length of 0.6 m is already slightly beyond the previous estimate of 0.55 m

acceleration and thus velocity are lower if the swivel bar's rotational inertia is higher. This is confirmed by the velocity gain, which is positively correlated with the rotational inertia of the swivel bar.

The velocity gain is no longer a function of the angle marked q_2 in figure 5-1b, due to the symmetry induced by $c_{2,x} = 0$, and simplifies to the following expression^[vi]:

$$G_\omega = -\frac{I_2}{I_2 + m_2 l_1^2 + I_1 + m_1 c_1^2} = -\frac{I_2}{I_{robot}} \quad (5-3)$$

from which it is evident that a higher I_2 increases G_ω . I_2 and G_ω are even approximately proportional ($I_2 \propto G_\omega$) if the rotational inertia of the swivel bar is relatively low compared to the rotational inertia of the entire robot about its contact point (toe) I_{robot} . The expression furthermore implies that the velocity gain deteriorates if Skippy is stretched (directly related to the increase of c_1 , L_1 and I_1), which is in conformance with observations of the velocity gain in figure 5-2.

From the positive correlation between I_2 and G_ω , it follows that the required power for controlling the swivel bar is lower if the swivel bar has a higher rotational inertia. The swivel motor—whose size is related to the power it can deliver—can thus be smaller if the swivel bar has a higher rotational inertia. As such, the swivel bar is ideally a long beam with two point masses on both ends, maximizing its rotational inertia for its size and mass. Here we find another trade-off. A light swivel bar requires a heavy motor and a light motor requires a heavy swivel bar. Battery placement can possibly offer a solution.

Batteries

The 3×3 -cell battery packs weigh $m_{bp} = 0.10$ kg per pack. The batteries are one of the few parts that permit more freedom regarding their placement. They can be placed close to the knee or close to the swivel bar. We found that a heavier swivel bar allows for a lighter swivel motor. However, it is not desired to add additional 'dead' mass to the robot—mass that serves no purpose other than to alter the mass distribution—since we want the robot to be as light as possible. Instead, we can exploit the possibility of placing battery packs in the swivel bar. The batteries would then be placed at both tips of the swivel bar, to maximize its rotational inertia. If the swivel bar has a length of $l_{sw} = 0.5$ m, the batteries add $\frac{1}{2}m_{bp}l_{sw}^2 \approx 0.05$ kg m² rotational inertia to the swivel bar, allowing for the bar itself to be very light. One concern is the wiring for the electricity that is drawn from the (rotating) batteries, for which we possibly require a slip ring. Another concern is that we want the swivel bar to be able to 'snap off' or at least 'snap through' if a crash would otherwise cause it to break, which possibly causes Skippy's power supply to be dismantled. Controversially, a detachable swivel bar allows for a possibly quick or hot-swappable battery replacement mechanism.

Swivel motor selection

To obtain an indication of the size of actuator we require for swivel balancing, we conduct a small balancing experiment with the earlier described flywheel on a stick model, depicted

^[vi]The expression of the velocity gain for any $c_{2,x}$ is very long and can be found in (Featherstone, 2012a)

in figure 5-1b. Since this system is a special case of the acrobot, we only have to adjust previously defined system parameters. Table 5-2 lists the new parameters. The first body called ‘main’ represents all of Skippy, except for the rotational inertia of the swivel bar; the parameters are an approximation of Skippy with a knee that is bent 90° .

| | i | joint | body | m_i [kg] | I_i [kg m ²] | $c_{i,x}$ [m] | $l_i = r_{i+1,x}$ [m] |
|---------------|-----|--------------|------------|------------|----------------------------|---------------|-----------------------|
| swivel Skippy | 1 | toe | main | 2 | $m_1 0.23^2$ | 0.63 | 0.78 |
| | 2 | swivel joint | swivel bar | 0 | $2 m_{bp} 0.25^2$ | 0 | — |

Table 5-2: Skippy in the swivel plane.

In contrast to the torso of the ‘bend’ balancers, the swivel bar is permitted an indefinite range of rotation. The symmetry of the system furthermore allows it to act more slowly, for which we lower the pole of Featherstone’s controller to $p = 2$ to lower the required power input. It also implies that the velocity gain can be much lower than those of bend Skippy in figure 5-2. Moreover, the swivel bar is not required to reach a final position for q_2 . As such, we set the positional gain k_4 in equation (5-2) to zero, such that the controller’s task is not divided between tracking a configuration and maintaining balance, but only focussed on the latter.

We run the balancing experiment described in section 5-2-3—where Skippy has to recover from a 5° balancing offset—with parameters listed in table 5-2 and $m_{bp} = 0.1$ kg, and without electrical saturation. We observe peak powers of 40 W, a maximum swivel velocity of $\dot{q}_2 \approx 35$ rad s⁻¹ and a maximum swivel torque of $\tau_2 \approx 2.5$ rad s⁻¹. A light 24 V motor that fits this profile is the Maxon DSX22S 15 W 24 V motor, with a weight of a mere 66 g. At 30 V, the motor has a peak power of $P_{mech} = 61$ W, a stall torque $\tau_s = 0.15$ N m and no-load velocity $\omega_z = 1623$ rad s⁻¹. The torque and velocity range require a relatively small transmission ratio, for which we install a miniature harmonic drive with transmission $R_{123} = 30$ and efficiency $\eta = 0.70$. The resulting rotor’s reflected rotational inertia was found to be negligible due to the low rotor inertia and low transmission ratio. The resulting peak power, stall torque and no-load velocity are $P_{mech} \approx 43$ W, $\tau_s \approx 3.2$ N m and $\omega_z = 54$ rad s⁻¹ respectively, which should be sufficient for balancing with the swivel bar.

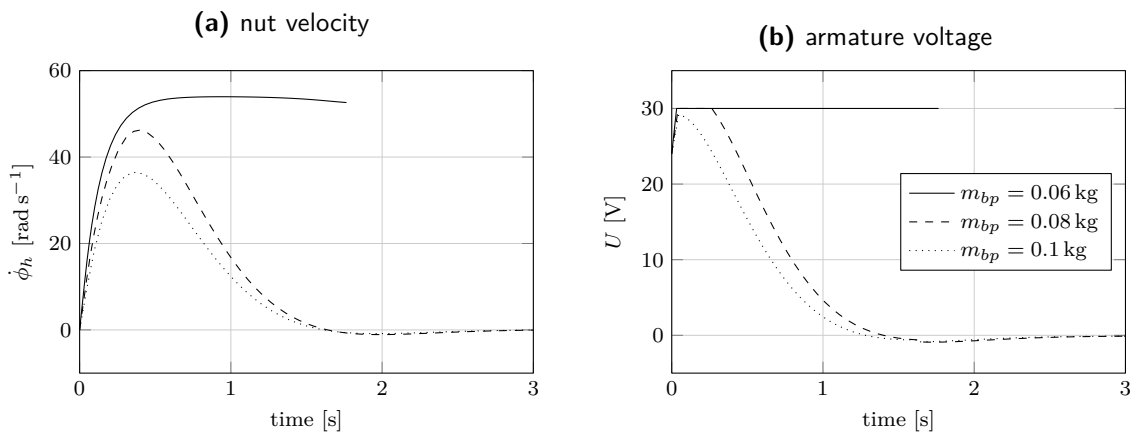


Figure 5-6: Balancing results of balancing in the swivel plane for various battery masses m_{bp} attached to the swivel bar. Balance performance deteriorates for lower values of m_{bp} .

We run the same balancing experiment, including electrical saturation. Results are plotted

in figure 5-6 for three values of $m_{bp} \in \{0.06 \ 0.08 \ 0.1\}$ kg. The corresponding velocity gains are $G_\omega \in \{0.0085 \ 0.0113 \ 0.0141\}$ respectively, which are notably lower than the velocity gains in figure 5-2. We observe that Skippy is able to recover balance without any voltage saturation for $m_{bp} = 0.1$ kg. For $m_{bp} = 0.08$ kg, balancing is, as expected, more difficult and for $m_{bp} = 0.06$ kg, Skippy is unable to recover balance and hits the ground at $t = 1.8$ s.

It is anticipated that we do not have to add additional weight to the swivel bar, as it performs well with the two battery packs and a sufficiently light motor that fits our mass budget.

5-3-3 Results

This section discusses results of a new conceptual mass distribution of Skippy's torso, as defined by sections 5-3-1 and 5-3-2. New torso parameters are presented, including a discussion of their effect on the balancing and jumping performance.

Most of the torso's mass is assigned to particular parts of the torso, including the mass of two battery packs. The study concludes that there is little freedom to choose the mass distribution. After re-analysing Skippy's jumping performance, it was decided to place also the third battery pack towards the swivel bar (close to the motors), the tip of the torso. The estimated CoM of the motors and third battery pack is $0.45 u_{6,x}$. Skippy's jumping performance has slightly deteriorated as a result of the new mass distribution. The mass of the swivel bar including its driveline, protective parts and two battery packs is estimated at a total 0.3 kg at $0.6 u_{6,x}$. The new torso mass and inertia is then determined by the following final mass distribution:

| description | type | mass [kg] | x in Ψ_6 |
|------------------|-------------|-----------|---------------|
| four-bar linkage | point | 0.3 | 0 |
| motors & housing | point | 0.6 | 0.45 |
| swivel bar | point | 0.3 | 0.6 |
| frame & cover | distributed | 0.6 | -0.1 to 0.5 |

This mass distribution is graphically illustrated by figure 5-5. The corresponding RoG and location of the CoM are $r_g = 0.22$ m and $r_{6,x} = 0.32$ m. Compared to the previous values ($r_g = 0.17$ m and $r_{6,x} = 0.38$ m), the increased RoG is expected to improve balancing performance, but the reduced value of $r_{6,x}$ is expected to worsen jumping performance.

The new velocity gain is plotted in figure 5-7. It shows that, for an extended knee joint, Skippy's velocity gain has nearly doubled, which suggests a major improvement to Skippy's balancing performance. In addition, simulations confirm the improved balancing performance. Identical simulations to those conducted in section 5-2-3 are conducted, where we compare both results of Skippy for both 30 V and 18 V. New and old balancing results are plotted in figure 5-8. Skippy no longer hits itself or the ground.

Rerunning the jumping simulations described in section 4-6-2 confirms Skippy's reduced jumping performance, which now jumps to a height of 3.73 m, approximately 10% less than previously. The effectively shorter torso (lower value of $r_{6,x}$) and increased RoG causes less vertical kinetic energy build-up. Firstly, the launch phase is slightly shorter due to the lowered value $r_{6,x}$, which reduces the maximum amount of work that can be delivered by the motor. Secondly, build-up of energy is less efficient since more energy is stored as rotational energy, which is due to the effectively shorter torso and increased RoG. More rotational and

horizontal momentum is built-up, which is in line with the predictions in appendix A, where it was predicted that a longer leg (which is effectively equivalent to a shorter torso) and a higher ROG build up more horizontal and rotational momentum. By changing the initial conditions of joint positions (e.g. setting the initial COM offset to $\phi_0 = \pi/2 + 0.3$ instead of $\phi_0 = \pi/2 + 0.2$ rad), small improvements to the jumping performance can already be made that lead to a jumping height of approximately 3.83 m. Further improvements could be made by increasing the torso length, but this is advised against due to the limited mass budget. Instead, it is advised to look into the landing phase, which is possibly able to improve jumping performance, as further described in chapter 6.

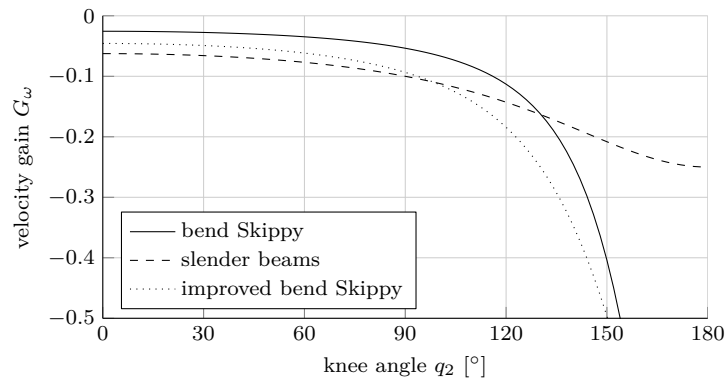


Figure 5-7: Skippy's new mass distribution improves the velocity gain and thus balancing performance.

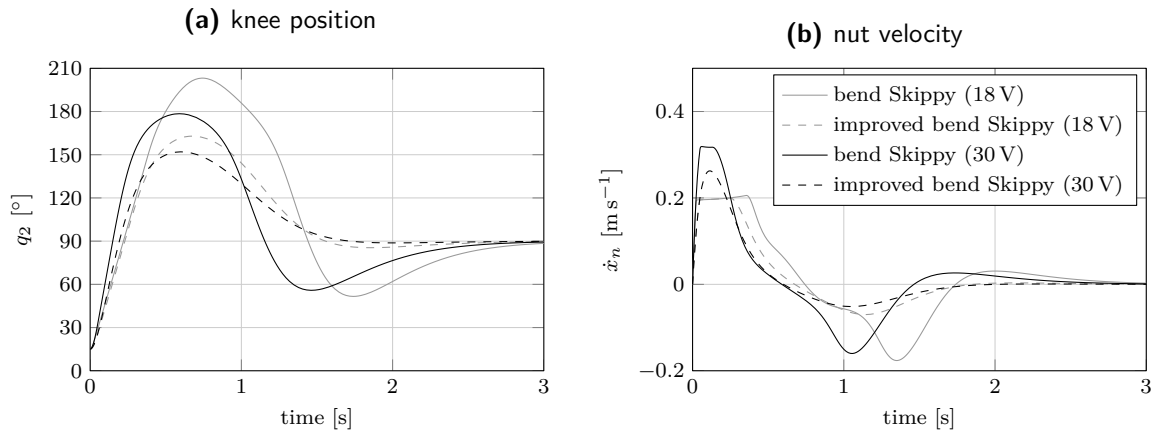


Figure 5-8: Skippy's new mass distribution improves balancing performance.

Concluding remarks

This chapter addresses the required additional ability of balancing. Balancing is an action that differs much from jumping, which is reflected by desired mechanical system requirements that are different from those required for hopping. Even though not focussed on controller design, we have been able to make sound predictions about Skippy's balancing performance

through a measure called the velocity gain, and verifications of those predictions have been done with a simple controller. The balancing performance is hugely affected by dimensional and inertial properties of the various bodies. A closer analysis concerning mass distribution in Skippy is done consequently, which resulted in a mass distribution that improves the balancing performance and—expectedly—slightly deteriorates jumping performance. According to the velocity gain, which has doubled for the worst balancing positions, Skippy’s new balancing performance is much improved. However, Skippy jumps approximately 10% less high. The results stress the general importance of taking required additional abilities at an early stage of the iterative design process into consideration for design criteria, as they can demand mechanical adjustments that worsen the jumping performance.

Chapter 6

Discussion

The goal is to design an action strategy and mechanism for Skippy to hop up to 4 m that is compliant with Skippy's required additional abilities and physical limitations. This thesis presents an iterative study of machine and behaviour co-design towards this objective. The various chapters in this thesis show how step-by-step progression is made with this approach. Every chapter looks into limited aspects of Skippy. The scope of the possibilities is initially reduced by thorough reasoning and careful decisions to obtain a simple robot model (i.e. the 2D robot model defined in chapter 2). The action strategy and mechanism are then expanded on (e.g. by introducing an elastic driveline with a non-linear spring) and new restrictions and requirements are introduced that shrink the set of solutions (e.g. modelling driveline limitations).

The above described iterative approach is proven effective, as it led to the design of a realistic strategy and mechanism able to make powerful hops towards the desired height in a controlled way, for which we exploited the possibilities of combining the design of the mechanical system and the action strategy (e.g. adjusting dimensional and stiffness properties to allow maximum possible power delivery by the motor). The iterative approach helps to keep the model sufficiently simple and thus the scope of possibilities comprehensible, which translates to keeping the number of tunable parameters manageable. An example of model simplification is the initial neglect of the linear and elastic driveline kinetics, or the simplification of considering only 2D systems.

The pitfall of iterative design methods is that its many design steps are governed by early and many decisive stages. Already in the very beginning of the project, many decisions are to be made to obtain a feasible research scope, even for a robot as simple as Skippy. These decisions include those related to the mechanical design and action strategy, such as the morphology and type of actuation, but also the model simplifications, such as the decision of using joints that are geared 1:1 instead of a full four-bar linkage. In some cases, the decisions taken are not right and lead to a dead end, in which case it is necessary to do backtracking and to re-evaluate previous decisions. With the benefit of hindsight, a better decision can be made. For example, we have experienced a dead end in section 3-2-3 where an inverse dynamics approach was pursued to find an action strategy. This approach resulted in an aggressive

action strategy that violated driveline limitations. From hindsight was learnt to base the action strategy on these driveline limitations instead of violating the driveline limitations as a result of the analysis.

In using the iterative approach, we have been able to make substantial progress with Skippy. Future research is to further improve the degree to which the model is realistic, but it is advised to keep the scope of solutions comprehensible. That is, the introduction of objectives, limitations and tunable model properties should be balanced. A selection of topics that are of interest for future research are presented below, as they build directly on assumptions and limitations of the current research.

- **required additional abilities** It is confirmed that balancing and jumping are two conflicting requirements. Other required additional abilities have yet to be analysed, as they could also require changes of mechanical system properties—such as mass distribution and spring stiffness—that can improve or worsen the thrusting or steering performance. These actions include landing, in-flight control, small hops and standing up. The landing phase, which is similar to the launch phase, is possibly able to improve the jumping performance. One of the (pessimistic) assumptions made in the thesis is the initial thrust force of $F_y = 200$ N, which followed from an approximation of the desired average thrust force made in section 2-1-2. This force has a significant influence on the system behaviour, partly because the spring profile depends on it, see equation (4-17). While the assumption is not a bad one, a better estimation can be made by considering also the landing phase.
- **3D Skippy** This thesis has focussed only on 2D models, where out-of-plane forces are assumed to be opposed by constraint forces that keep Skippy in its 2D vertical plane. Whereas this is a valid assumption for initial thrust and balancing analyses, future research will have to extend to 3D modelling. It has been shown that a robot with a similar morphology as Skippy is able to follow controlled trajectories in 3D (Azad, 2014), but no hopping strategy has yet been developed that is able to control both lift-off momenta in the bend and swivel plane. Some additional abilities, e.g. in-flight control, are largely determined by 3D effects such as the non-integrability of 3D rotations.
- **sensors and real-time control** This thesis has focussed little on controllers and sensors and has not considered a real-time system, including a discrete system representation, various types of delays, measurement errors and state estimation errors. The stance phase is very short, which implies that the controller has to act very fast. Nonetheless, the action strategy during most of the stance phase is relatively simple, as it is determined by (feedforward) voltage saturation. Slightly more difficult to implement is nut velocity saturation, as the nut velocity is limited by a simple feedback control law (i.e. a sliding P-controller). It is expected that the beginning of the landing phase and end of the steering phase are most challenging to control, where the controllers are to prevent Skippy from slipping. With the incredibly short time available shortly after landing and shortly before lift-off, it may be required to make the anti-slip controllers as simple as look-up tables that are generated from for instance reinforcement learning.
- **spring design** The spring profiles that are used in the thesis are not based on real components and materials. Designing a main spring that (1) fits the regressive spring profile, (2) is extremely light and (3) has little damping is a challenging task. Two out of many options are suggested. The first is a tension spring made from rubber, as rubber has a high gravimetric energy density and a regressive stress-strain profile.

However, its viscous properties may be troublesome. The second option is a glass fibre buckling spring with linear guide, which is heavier than a rubber spring, but still lighter than known alternatives. A friction model of the spring will have to be included in the simulations as they will negatively influence Skippy's jumping performance. Several other mechanical challenges are to be tackled, which are all related to the requirement of minimizing weight (e.g. the protective cover and the four-bar linkage).

- **multi-objective optimization** With the various conflicting objectives that Skippy has, it is advised to do a multi-objective optimization (also known as Pareto optimization)—which allows optimization of multiple objective functions simultaneously—to make decisions about final design parameters, which include the dimensions of the final four-bar mechanism and the stiffness of realistic spring profiles. The optimization can improve the current jump height, which is slightly below 4 m, without having to further increase Skippy's size. Moreover, the four-bar linkage allows for more transmission ratios due to its many dimensional parameters, which can be optimized to further improve the steering capabilities.

The above list with the many topics for future research—which is not complete—shows that the research that is performed and described in this thesis marks only the starting point of the design project of Skippy. Nevertheless, the results show that there is realistic potential for the realisation of such a hopping machine.

Chapter 7

Conclusion

This thesis makes the first steps towards the realisation of a robot, named Skippy, with unique athletic abilities. The robot is to be able to jump up to heights of 4 m, do acrobatic manoeuvres (including somersaults), balance, stand-up and survive its crashes—yet, it is maximally simple in having only two actuators. The focus is on the most difficult task of Skippy, which addresses the mechanical design and the pertinent action strategy for jumping up to a height of 4 m in a controlled way. The problem demands a complete understanding of the mechanical system behaviour, because the requirements of Skippy aim to achieve performance that is on the edge of what is physically possible with COTS components—and as such cannot be achieved by relying solely on either an action strategy or a mechanical system.

By considering both the action strategy *and* the mechanical system, we open up new dimensions of possibilities. However, the large number of possibilities could also lead to loss of overview and understanding. As such, it is decided to do an iterative design study of machine (mechanical system) and behaviour (action strategy) co-design, where little steps are taken forward, during which we preserve our understanding of the system behaviour, its possibilities and its limitations. Withal, a detailed design approach is made feasible owing to the simplicity of Skippy. Initially, simplifications are made where possible and only those aspects that are most crucial are taken into consideration. For this reason, Skippy's system is simplified to be planar in this thesis, for which Skippy only uses one instead of two actuators.

Initial studies on Skippy are coarse in that they consist of only single-dimensional calculations about energy flow, system dimensions and feasible components. Nonetheless, they lead to very carefully taken design decisions that severely reduce the scope of the project to one that can be built on. These studies deduce a mechanism that realizes passive leg elongation through the introduction of an ankle with spring. The mechanism practically decouples Skippy's launch towards 4 m into a thrust phase and steering phase, which respectively address the power input and controllability. The thrust phase requires a series elastic actuation for energy storage, which is connected to a driveline that consists of a ball screw and a four-bar linkage for desired motion transfer characteristics. The steering phase requires a passive ankle that allows for leg elongation shortly before lift-off, which improves the controllability of lift-off momenta. However, the required thrust profile—even if smoothed—that is to be delivered to

the mechanism by the motor through the series elastic driveline is difficult to find through inverse dynamics.

Instead of finding an action strategy through inverse dynamics, the thesis devises a feedforward action strategy that considers various system limitations that are taken into account by dynamic saturations. These limitations include the limited nut velocity of the ball screw (which is part of the transmission) and maximum armature voltage. Both the thrust and steering strategy consist of a simple torque command that is dominated by these dynamic saturations and therefore allow for achieving that which is on the edge of what is physically possible. Thrust is mainly limited by maximum voltage and steering by nut velocity. The strategy also allows for a better understanding of system behaviour, which has led to adjustments of system parameters that result in maximum power input, efficient energy conversion and acceptable steering capabilities. These parameters include stiffness of non-linear spring profiles and dimensional properties that affect the variable transmission ratio. An interesting lesson is learnt here. Instead of pursuing an approach where Skippy is forced to act in a particular way, we consider what happens if the robot is acting at its physical limits (i.e., in saturation), after which it is tried to move the physical limits by redesigning the physical system, to improve the behaviour. This illustrates the importance of combining the physical design and the system behaviour (i.e. action strategy) if one is aiming for high performance.

It has been shown that required additional abilities should also be taken into consideration for decisions in an early stage of the iterative design project. This has been done by analysing one such ability: Skippy's physical ability to balance. Desired system properties for balancing counter those of thrusting, which complicate design decisions. With the newly defined system characteristics that put more favour to balancing, Skippy's jumping height reduces slightly below 4 m, whereas it jumped over 4 m prior to the balancing analysis.

For the final design of Skippy, all required additional abilities have to be taken into account and performance trade-offs have to be made. Once all capabilities and limitations of Skippy and its parameters are quantified, a multi-objective optimization can be done that aids in trade-offs for final design decisions, which is likely able to improve the current jump height.

Appendix A

Quasi-static momentum analysis

In order for Skippy to jump high, positive vertical momentum has to be built up. However, we also have to control for the horizontal and angular momentum of the robot in order to jump in a desired fashion. It is explained in section 2-1-4 that it is expected that Skippy will build up positive horizontal and angular momentum during the thrust phase. This section contains a quasi-static analysis that aims to provide more evidence and insight towards build-up of angular momentum during the thrust of a simplified model of Skippy.

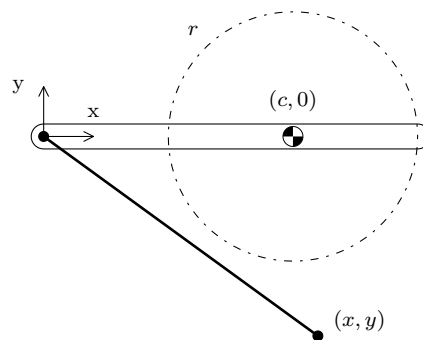


Figure A-1: Sketch of a simple hop robot with the global coordinate frame at the 'bend' joint and a massless leg.

Consider a very simple model of planar Skippy, depicted in figure A-1. The coordinate frame is fixed in the actuated revolute bend joint, oriented such that the x-axis passes through the CoM of the torso. The CoM of the torso is located at a unit distance $c = 1$ from the coordinate frame with radius of gyration (ROG) of r . The leg is massless and connects to the ground via a passive revolute joint which we call the foot, located at (x, y) . Gravity is not modelled, on the grounds that the desired thrust acceleration is much greater than gravitational acceleration. For a non-zero thrust torque τ at the bend joint $(0, 0)$, it can be derived that the angle of the GRF $\alpha_F(x, y)$ at the foot can be calculated as follows, neglecting the convective accelerations:

$$\alpha_F = \arctan \left(r^2 x + y^2, y (x - 1 - r^2) \right) \quad (\text{A-1})$$

Note that the direction of the GRF is invariant with the magnitude of the torque, the mass of the torso and the uniform scaling of all lengths (c , r and the leg length $\sqrt{x^2 + y^2}$). This justifies the assumption $c = 1$. The direction of the GRF according to equation (A-1) is plotted in figure A-2 for $r = 0.5$ and $r = 1$. The set of foot positions in which the thrust torque causes the torso to rotate about the origin is marked by a dashed line. If the foot is on this line then the leg acceleration is zero. The gray area marks the set of foot positions that cause a negative moment about the COM. The edge of this area—which is a half-circle between the bend joint and the COM (not a function of the ROG)—marks the set of foot positions in which the GRF points through the COM. All foot positions outside this area (white) cause the thrust torque to deliver a positive moment about the COM.

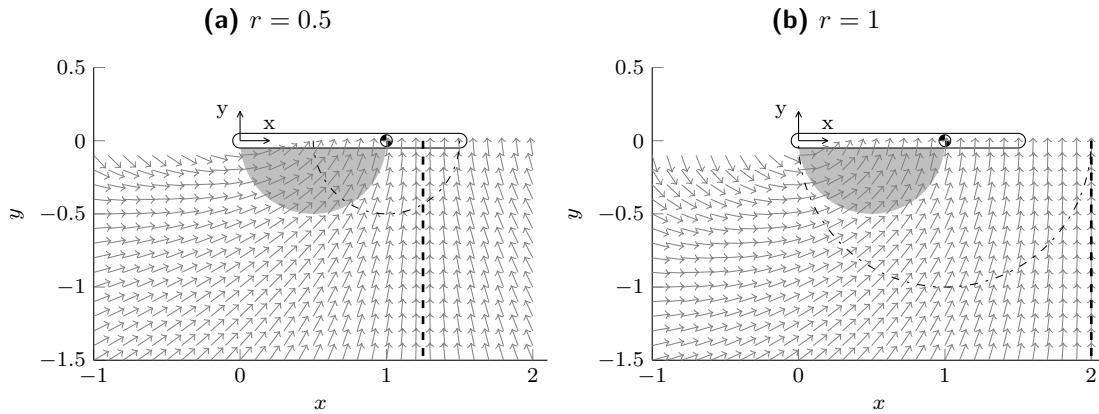


Figure A-2: The direction of the GRF as function of the foot position (x, y) for two different ROGs r , denoted by the dash dotted line. The dashed

From the figures it is evident that almost all foot positions cause a positive moment about the COM. During the launch phase, the foot moves from a flexed to stretched orientation over a circle about the origin; positive angular momentum is built up towards the end of the launch phase as a result of integrating the positive moment over time. This implies that Skippy would be bound to do somersaults in order to be able to land, which is undesired. The rotational energy in Skippy can never be utilized to jump higher.

Furthermore, note how the lines of action of GRF's pass further to the right of the COM in figure A-2b. It can be observed and devised that a greater radius of gyration of the upper body leads to more momentum build-up. Moreover, longer legs generally lead to more momentum build-up.

Several options can be exploited to obtain different lift-off momenta.

- Change the leg length. Shorter legs build up less angular momentum. However, shorter legs impair the ability to jump high, unless we add the length to the torso, which in turn would increase its radius of gyration and thus the positive momentum build up. Exchanging leg length for torso length moreover means that Skippy becomes heavier, because the torso contains more structural elements than the leg. As a further matter,

if the leg is to build negative momentum by partially swinging through the grey area in figure A-2, then it has to be significantly shorter than c . A leg shorter than c does not allow Skippy to balance in very crouched configurations and is thus to be avoided.

- Change the initial conditions of the launch phase. An initial angular offset of the COM with respect to the balanced configuration will introduce a counter-moment due to the gravity component, which has been neglected in this analysis. Secondly, the initial angular momentum can be changed by assuming a non-zero initial angular velocity. This option is certainly to be exploited, but changing initial conditions does not improve the controllability or robustness. If a set of (unfortunate) initial conditions is given, then little can be done to tweak the outcome.
- Alter the thrust torque profile. As the moment arm of the GRF varies with foot position, it is possible to vary the cumulative total angular momentum by varying the thrust torque as a function of leg angle. This would still not allow Skippy to perform a non-rotating hop, because the angular momentum would always be positive; it would at least allow Skippy to perform *only* controlled somersaults. A disadvantage is that changing the thrust torque profile limits the possibilities to build up a lot of vertical momentum.
- Lower the average thrust torque, which extends the duration of the launch phase, to increase the influence of gravity that acts on the model. Note that lowering the thrust torque does not change the direction of the GRF (and thus momentum build-up) according to equation (A-1); however, in reality, gravity acts on the model which also influences momentum build up. An extended launch phase allows for gravity to build up more (or less, depending on the direction in which the robot is falling) angular momentum. Nonetheless, an undesired side-effect of extending the duration of the launch phase (without extending the length of the stroke) is less build-up of *vertical* momentum. Ideally, the launch phase maximizes build-up of vertical momentum.

A more practical solution was found for Skippy, which splits the launch phase into a thrust and steering phase. The thrust phase focusses only on thrusting (building up vertical momentum) and the steering phase only on steering (correcting for build-up of horizontal and angular momentum). A leg was introduced able to change its length through a passive ankle joint. The ankle prolongs ground contact by decompression at the end of the thrust phase. This decompression can be used for steering, where momentum is brought to a desired value.

Appendix B

Hybrid dynamics and loop closure

This attachment describes how the equations of motion (EOM) of Skippy are formulated to solve Skippy's dynamics, which incorporates so-called hybrid dynamics (Featherstone, 2008b) and kinematic loop closure functions (Featherstone, 2008a).

The EOM of an open-loop rigid-body system can be written in the following canonical form:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \quad (\text{B-1})$$

\mathbf{H} is a joint-space inertia matrix and \mathbf{C} a joint-space vector of force terms, including Coriolis and centrifugal forces and gravity. Efficient algorithms are available to calculate \mathbf{H} and \mathbf{C} . Forward dynamics FD or inverse dynamics ID calculations can be done to calculate the joint accelerations $\ddot{\mathbf{q}}$ or forces $\boldsymbol{\tau}$, respectively, given the other variables in equation (B-1) as known:

$$\begin{aligned} \ddot{\mathbf{q}} &= \text{FD}(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}) \\ \boldsymbol{\tau} &= \text{ID}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \end{aligned}$$

Generally, FD is used to calculate unknown joint accelerations. However, Skippy's system has two difficulties:

- \ddot{q}_1 and \ddot{q}_2 are already known (they are zero, fixing the foot to the ground) and their forces (reaction forces $\tau_1 = F_x$ and $\tau_2 = F_y$) are unknown. So we reformulate the EOM in the form of a hybrid dynamics equation that takes τ_1 , τ_2 and \ddot{q}_3 to \ddot{q}_6 as unknowns. Hybrid dynamics combines FD and ID.
- Skippy's joint variables are not all independent, because of the loop-closure constraints. To obtain correct joint acceleration variables, we require a mapping from independent to dependent joint accelerations.

Let us define the following subscripts to distinguish between terms corresponding to known (1) and unknown (2) joint accelerations:

$$\begin{array}{l} \ddot{q}_1 \text{ known, } \tau_1 \text{ unknown:} \\ \ddot{q}_2 \text{ unknown, } \tau_2 \text{ known:} \end{array} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{21} \\ \mathbf{H}_{12} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} \quad (\text{B-2})$$

We furthermore define \mathbf{y} as a vector of independent position variables that define \mathbf{q} uniquely, such that we can write the EOM in equations (B-1) and (B-2) as

$$\boldsymbol{\tau}_y = \mathbf{H}_y(\mathbf{y})\ddot{\mathbf{y}} + \mathbf{C}_y(\mathbf{y}, \dot{\mathbf{y}}) \quad (\text{B-3})$$

\Rightarrow

$$\begin{array}{l} \dot{\mathbf{y}}_1 \text{ known, } \boldsymbol{\tau}_{y,1} \text{ unknown:} \\ \dot{\mathbf{y}}_2 \text{ unknown, } \boldsymbol{\tau}_{y,2} \text{ known:} \end{array} \begin{bmatrix} \boldsymbol{\tau}_{y,1} \\ \boldsymbol{\tau}_{y,2} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{y,11} & \mathbf{H}_{y,21} \\ \mathbf{H}_{y,12} & \mathbf{H}_{y,22} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{y}}_1 \\ \dot{\mathbf{y}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{y,1} \\ \mathbf{C}_{y,2} \end{bmatrix} \quad (\text{B-4})$$

Equation (B-4) can be simplified for the case of Skippy. For Skippy, the only dependency occurs in \mathbf{q}_2 . $\ddot{\mathbf{q}}_1$ is a 2×1 vector of independent (known, zero) joint accelerations and $\boldsymbol{\tau}_1$ corresponds to the (unknown) GRFs. So we have:

$$\begin{aligned} \ddot{\mathbf{y}}_1 &= \begin{bmatrix} \ddot{y}_1 \\ \ddot{y}_2 \end{bmatrix} = \ddot{\mathbf{q}}_1 = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \boldsymbol{\tau}_{y,1} &= \boldsymbol{\tau}_1 \end{aligned} \quad (\text{B-5})$$

This allows us to rewrite equation (B-4):

$$\begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_{y,2} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{y,21} \dot{\mathbf{y}}_2 \\ \mathbf{H}_{y,22} \dot{\mathbf{y}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{y,1} \\ \mathbf{C}_{y,2} \end{bmatrix} \quad (\text{B-6})$$

To calculate \mathbf{q} from \mathbf{y} , we define the loop closure function $\gamma(\mathbf{y})$, i.e. $\mathbf{q} = \gamma(\mathbf{y})$. The joint velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$ are found by differentiating the loop closure function:

$$\dot{\mathbf{q}} = \mathbf{G}\dot{\mathbf{y}} \quad (\text{B-7})$$

$$\ddot{\mathbf{q}} = \mathbf{G}\ddot{\mathbf{y}} + \mathbf{g} \quad (\text{B-8})$$

with the velocity mapping or Jacobian \mathbf{G}

$$\mathbf{G} = \frac{\partial \gamma}{\partial \mathbf{y}} \quad (\text{B-9})$$

and the convective acceleration vector \mathbf{g}

$$\mathbf{g} = \dot{\mathbf{G}}\dot{\mathbf{y}} \quad (\text{B-10})$$

Note that it follows from $\mathbf{y}_1 = \mathbf{q}_1$ that \mathbf{G} is block-diagonal and that \mathbf{G}_{11} is an (2×2) identity matrix :

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{22} \end{bmatrix} \quad \text{with} \quad \mathbf{G}_{11} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{B-11})$$

This allows us to calculate \mathbf{H}_y , \mathbf{C}_y and $\boldsymbol{\tau}_y$ as follows:

$$\begin{aligned} \mathbf{H}_y(\mathbf{y}) &= \mathbf{G}^\top \mathbf{H}(\mathbf{q}) \mathbf{G} \\ \mathbf{C}_y(\mathbf{y}, \dot{\mathbf{y}}) &= \mathbf{G}^\top (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{H}(\mathbf{q}) \mathbf{g}) \\ \boldsymbol{\tau}_y &= \mathbf{G}^\top \boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \mathbf{G}_{22}^\top \boldsymbol{\tau}_2 \end{bmatrix} \end{aligned}$$

which we can use to calculate our unknown independent joint accelerations $\ddot{\mathbf{y}}_2$ as follows:

$$\ddot{\mathbf{y}}_2 = \mathbf{H}_{y,22}(\mathbf{y})^{-1} (\boldsymbol{\tau}_y - \mathbf{C}_{y,2}(\mathbf{y}, \dot{\mathbf{y}}))$$

Having $\ddot{\mathbf{y}}_2$, the dependent joint accelerations in $\ddot{\mathbf{q}}$ can be calculated using equation (B-8) and the independent joint torques $\boldsymbol{\tau}_1$ can be found accordingly

$$\boldsymbol{\tau}_1 = \mathbf{H}_{y,21}(\mathbf{y})\ddot{\mathbf{y}}_2 + \mathbf{C}_{y,1}(\mathbf{y}, \dot{\mathbf{y}})$$

Appendix C

Energy flows

With an increasing level of model complexity, the model becomes more prone to mistakes and the comprehensibility deteriorates. We compute and compare all the energy flows in Skippy with the following purposes:

- Providing a check to verify that our model is not incorrect, by summing energies and work done on and by the system. We refer to this study as the energy audit.
- Improving our understanding of the system, by observing the various energy levels throughout time, which could possibly tell us what to focus on to further improve the performance of Skippy.

This appendix focusses on the energy audit and expressions of all energies involved. For the energy audit we should have that the sum of all system energies plus work done by the system W_{out} (heat generation) minus work done on the system W_{in} (by the motor) remains constant throughout time:

$$\frac{d(E_{sys} + W_{out} - W_{in})}{dt} = 0 \quad (C-1)$$

in which E_{sys} is the sum of all internal system energies, W_{out} is the work done by the system and W_{in} the work done on the system. The energy defined by the sum of these variables should be constant.

The system

It is important to define the system to properly distinguish between E_{sys} , W_{out} and W_{in} . The system, denoted *sys*, consists of all the moving bodies, the springs, the ‘disembodied’ rotational inertia representing the out-of-plane rotational inertia of the motor’s rotor, coupling and ball screw spindle, and the motor’s electrical armature. Heat developed in bodies or the armature circuit is not included in the system, because it is considered ‘lost’ or ‘useless’ energy, and as such is classified as work done by the system W_{out} . The only source of work done on the system W_{in} is electrical energy that is fed to the armature circuit. Heat loss in other electric circuits (e.g. motor driver electronics), or heat loss as a result of conversion from chemical to electrical energy in the batteries, is not considered.

System energies

The system energy consists of all kinetic energies E_k and potential energies E_p :

$$E_{sys} = E_k + E_p = (E_{k,x} + E_{k,y} + E_{k,\phi} + E_{k,h}) + (E_{p,m} + E_{p,a} + E_{p,g})$$

| | | |
|--------------|--|--|
| $E_{k,x}$ | $= \frac{1}{2} m \dot{x}^2$ | horizontal translational kinetic energy of the CoM |
| $E_{k,y}$ | $= \frac{1}{2} m \dot{y}^2$ | vertical translational kinetic energy of the CoM |
| $E_{k,\phi}$ | $= \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{H}(\mathbf{q}) \dot{\mathbf{q}} - E_{k,x} - E_{k,y}$ | rotational kinetic energy |
| $E_{k,h}$ | $= \frac{1}{2} I_h \dot{\phi}_h^2$ | rotational driveline energy |
| $E_{p,m}$ | $= \int_{l_z} f_s(x_s) dx_s$ | main spring energy |
| $E_{p,a}$ | $= \int_0 f_{\tau 4}(q_4) dq_4$ | ankle spring energy |
| $E_{p,g}$ | $= m g y$ | gravitational energy |

$\mathbf{H}(\mathbf{q})$ is the joint-space inertia matrix, see Appendix B. m and $[x \ y]$ are the system's total mass and coordinates of its CoM respectively.

$E_{k,xy} = E_{k,x} + E_{k,y}$ is the sum of all translational kinetic energy of all bodies in the system. Translational kinetic energy is the most interesting value for Skippy as it is a measure for jumping performance and speed, which is why it is taken apart from other kinetic energies. The rotational kinetic energy in the driveline $E_{k,h}$ is the sum of kinetic energy of rotating parts in the driveline, which includes the motor rotor, spindle and coupling. The total kinetic energy E_k minus $E_{k,h}$ equals the total kinetic energy of all bodies $E_b = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{H}(\mathbf{q}) \dot{\mathbf{q}}$. E_ϕ is the total kinetic energy of all bodies minus the translational kinetic energy.

All system energies are calculated analytically, including the spring potential energies $E_{p,m}$ and $E_{p,a}$, as their profiles are readily available.

Work done on the system

The electrical work W_{el} done on the system equals

$$W_{in} = W_{el} = \int_0 P_{el} = \int_0 \min(0, UI) dt \quad (\text{C-2})$$

This equation implies that the electrical power flow is always positive or zero. It is consistent with the assumption that the motor driver electronics is non-regenerative. The motor cannot work as a dynamo, which implies that if the motor is doing negative work, then this work is lost as heat rather than being fed back into the battery. Resulting heat generation is considered as work done by the system instead, which is explained subsequently.

Work done by the system

Skippy has various sources of heat generation. Most heat generation results from resistor losses W_J in the motor armature circuit. The total heat losses are

$$W_{out} = W_J + W_T + W_{reg} + W_{bs}$$

$$\begin{aligned} W_J &= \int_0 J^2 R dt && \text{motor winding resistance heat} \\ W_T &= \int_0 |\dot{\phi}_h| J_z c_T dt && \text{motor friction torque heat} \\ W_{reg} &= \int_0 \max(0, UI) dt && \text{regenerative heat} \\ W_{bs} &= \int_0 |F_{fric} \dot{x}_n| dt && \text{ball screw heat} \end{aligned}$$

The largest source of heat generation is W_J . The heat that occurs from the friction torque in the motor W_T is very small and in most cases negligible. Motor heat W_{JT} is defined as the sum of W_J and W_T . W_{reg} is an additional source of heat generation that occurs if the motor is doing negative work (i.e. $\dot{W}_M < 0$), which results from the previously stated assumption that the motor driver electronics is non-regenerative. Idiomatically, we refer to this heat as regenerative heat. Note that W_{reg} remains constant if the motor is delivering positive work.

The only source of modelled heat generation that does not occur in the motor or its electric circuit is heat generation in the ball screw W_{bs} . The springs are modelled to be purely elastic and do not generate heat in this model. The fact that real springs cannot be purely elastic has been taken into account in the design constraint that only 50% of the necessary energy to perform a 4 m hop can be retrieved from the previous hop, to be stored as potential energy in the spring (see section 2-1-5).

Energy audit

The energy audit defined in equation (C-1) confirms that energy is preserved. Note that all types of work done by the system are positive and monotonically increasing, since heat is considered permanently lost. Also work done on the system is positive and monotonically increasing, due to the nature of its (non-regenerative) definition in equation (C-2). System energies are positive but not necessarily monotonically increasing. If E_{sys} is decreasing, it implies that more heat is generated than electrical energy fed to the system.

We make a cumulative plot of all energies E_{sys} and work done by the system W_{out} minus work done on the system W_{in} as a function of time to verify that it remains constant throughout time. Note however that $-W_{in}$ is negative. For the cumulative plot it is desired that all quantities be positive. As such, we plot $W_{-in} = \max(W_{in}) - W_{in}$ instead of $-W_{in}$. $\max(W_{in})$ can be seen as a reservoir of energy that is being depleted during one particular simulation. As an example, the cumulative plot that corresponds to the data analysed in section 4-5-4 is displayed in figure C-1. We confirm that it is constant throughout time, for which it passes the energy audit. All other simulations mentioned in the main matter also pass the energy audit.

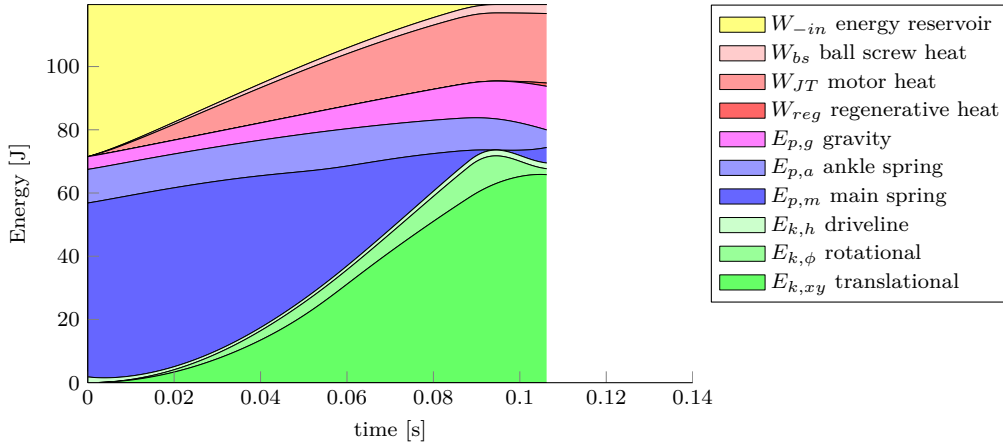


Figure C-1: The energy audit confirms preservation of energy.

Subsystems

Work done by the motor W_M is defined as mechanical work done by the motor W_{mech} plus motor friction torque heat W_T and motor winding resistance heat W_J in the motor's armature.

$$W_M = W_{mech} + W_T + W_J$$

Note that the energy flow of mechanical work done by the motor W_{mech} is neither included in W_{out} nor in W_{in} , as it is a flow *within* the system, not an incoming or outgoing flow. Yet, we are able to compute it by integrating the mechanical power that is delivered by the motor:

$$W_{mech} = \int_0 T_{mech} \dot{\phi}_h dt \quad (C-3)$$

This quantity can be used for additional verifications, for which we divide our system in two subsystems. We define sys, k as the system excluding the motor (and its armature) with

$$\begin{aligned} E_{sys,k} &= E_{sys} \\ W_{out,k} &= W_{bs} \\ W_{in,k} &= W_{mech} \end{aligned}$$

Congruently, we can define a subsystem that considers *only* the motor and its armature, sys, e :

$$\begin{aligned} E_{sys,e} &= 0 \\ W_{out,e} &= W_{mech} + W_J + W_T + W_{reg} \\ W_{in,e} &= W_{el} \end{aligned}$$

Both systems confirm energy preservation. Figure C-2 plots the energy audit of sys, e corresponding to the same data analysed in figure C-1. W_J and W_T are plotted separately to show how tiny W_T is.

Note

The energy audit plots can be confusing or difficult to read, as they contain the cumulative results of all system energies *and* work done by the system *minus* work done on the system. The plots are generated for the purpose of the energy audit, which is a valuable check for the correctness of simulations. Easier readable energy plots are used in the main matter of the thesis, which do not indicate energy preservation. In these plots, cumulative system energies are plotted above zero, cumulative work done by the system is plotted below zero and work done on the system is not plotted.

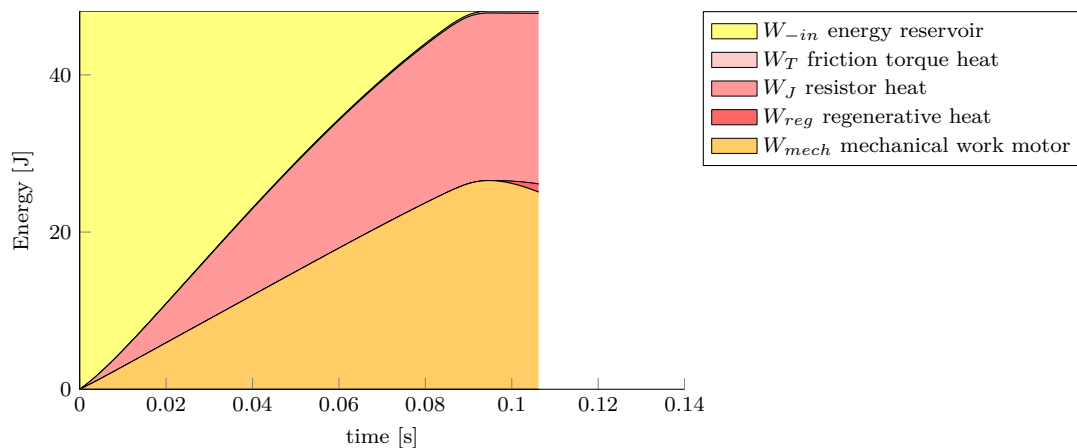


Figure C-2: The energy audit of the motor confirms preservation of energy. It can also be seen that the heat development W_T as a result from the friction torque is very small.

Bibliography

- Azad, M. (2014). *Balancing and Hopping Motion Control Algorithms for an Under-actuated Robot*. PhD thesis, The Australian National University.
- Berkemeier, M. and Fearing, R. (1999). Tracking fast inverted trajectories of the underactuated acrobot. *Robotics and Automation, IEEE Transactions*, 15(4):740–750.
- Berkemeier, M. D. and Fearing, R. S. (1998). Sliding and hopping gaits for the underactuated acrobot. *IEEE Trans. Robotics & Automation*, 14(4):629–634.
- Billingsley, J., Visala, A., and Dunn, M. (2008). Robotics in agriculture and forestry. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, chapter 46, pages 1065–1077. Springer.
- Boston Dynamics (2012). *SandFlea – Jumping Robot*. Boston Dynamics. datasheet v1.0.
- Boston Dynamics (2015). <http://www.bostondynamics.com/>. Accessed 05/10/2015.
- Buehler, M., Playter, R., and Raibert, M. (2005). Robots step outside. In *Int. Symp. Adaptive Motion of Animals and Machines (AMAM)*.
- Collins, S., Ruina, A., Tedrake, R., and Wisse, M. (2005). Efficient bipedal robots based on passive-dynamic walkers. *Science*, 307(5712):1082–1085.
- de Boer, T. (2012). *Foot Placement in Robotic Bipedal Locomotion*. PhD thesis, Delft University of Technology.
- Featherstone, R. (2008a). Closed loop systems. In *Rigid Body Dynamics Algorithms*, chapter 8, pages 141–169. Springer.
- Featherstone, R. (2008b). Hybrid dynamics and other topics. In *Rigid Body Dynamics Algorithms*, chapter 9, pages 171–193. Springer.
- Featherstone, R. (2008c). *Rigid Body Dynamics Algorithms*. Springer.

- Featherstone, R. (2012a). Analysis and design of planar self-balancing double-pendulum robots. *Romansy 19 – Robot Design, Dynamics and Control*, 70(1):64–73.
- Featherstone, R. (2012b). Spatial vector and rigid-body dynamics software. <http://royfeatherstone.org/spatial/v2/index.html>. Accessed 16/09/2015.
- Featherstone, R. (2014). Skippy: A versatile 3D hopper. Dynamic Walking Conference.
- Featherstone, R. (2015a). A new simple model of balancing in the plane. In *Int. Symp. Robotics Research (ISRR)*.
- Featherstone, R. (2015b). Quantitative measures of a robot’s ability to balance. In *Proceedings of Robotics: Science and Systems*, Rome, Italy.
- Festo (2014). *Bionic Kangaroo*. Festo. datasheet.
- Franklin, G. F., Powel, J. D., and Emami-Naeini, A. (2010). *Feedback Control of Dynamic Systems*. Pearson Education, Inc., 6 edition.
- Grimes, J. A. and Hurst, J. W. (2012). The design of ATRIAS 1.0: A unique monopod, hopping robot. In *International Conference on Climbing and Walking Robots (CLAWAR)*, pages 548–554.
- Grizzle, J., Moog, C., and Chevallereau, C. (2005). Nonlinear control of mechanical systems with an unactuated cyclic variable. *Automatic Control, IEEE Transactions*, 50(5):559–576.
- HDAG (2014). *Engineering Data CSF-Mini Gearboxes*. Harmonic Drive AG. Catalog.
- Hirose, H. and Masato, K. (2007). Honda humanoid robots development. *Philosophical Transactions of the Royal Society*, 365:11–19.
- Hobbelen, D. G. E. and Wisse, M. (2007). A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm. *IEEE Transactions on Robotics*, 23(6):1213–1224.
- Karszen, D. (2013). *Robotic Bipedal Running*. PhD thesis, Delft University of Technology.
- Karszen, J. G. D. (2007). Design and construction of the cornell ranger, a world record distance walking robot. Technical report, Cornell University.
- Kemp, C. C., Fitzpatrick, P., Hirukawa, H., Yokoi, K., Harada, K., and Matsumoto, Y. (2008). Humanoids. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, chapter 56, pages 1307–1333. Springer.
- Khan, H., Featherstone, R., Caldwell, D. G., and Semini, C. (2015). Bio-inspired knee joint mechanism for a hydraulic quadruped robot. In *International Conference on Automation, Robotics and Applications*.
- Kuswadi, S., Ohnishi, A., Takahashi, A., Sampei, M., and Nakaura, S. (2003). A one linear actuator hopping robot: modeling and control. *Advanced Robotics*, 17(8):709–737.
- Maxon Motor (2014). Program 2014/15 – high precision drives and systems. <http://www.maxonmotor.nl/maxon/view/catalog/>. Catalog.

- McGeer, T. (1990). Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62–82.
- Misumi (2015). *Misumi Catalog – Couplings, Motors*. Misumi. pages 960 –1008.
- Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., and Erkmen, A. M. (2008). Search and rescue robotics. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, chapter 50, pages 1151–1173. Springer.
- NSK (2008). *Precision Machine Components – NSK Linear Guides, Ball Screws, Monocarriers*. Motion & Control NSK. Catalog.
- Pfeifer, S., Riener, R., and Vallery, H. (2012). An actuated transfemoral prosthesis with optimized polycentric knee joint. In *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*.
- Raibert, M. H. (1986). *Legged Robots that Balance*. MIT Press, Cambridge, Massachusetts.
- Saranli, U., Buehler, M., and Koditschek, D. E. (2001). Rhex: A simple and highly mobile hexapod robot. *International Journal of Robotics Research*, 20:616–631.
- SBC (2015). *Ball Screw – Technical Data*. SBC. Catalog.
- Schwind, W. J. and Koditschek, D. E. (1997). Characterization of monoped equilibrium gaits. In *International Conference on Robotics and Automation*, volume 3, pages 1986–1992.
- Seyfarth, A., Geyer, H., Günther, M., and Blickhan, R. (2002). A movement criterion for running. *Journal of Biomechanics*, 35:649–655.
- Silva, M. F. (2007). A historical perspective of legged robots. *Journal of Vibration and Control*, 13:1447–1486.
- Spong, M. (1995). The swing up control problem for the acrobot. *Control Systems, IEEE*, 15:49–55.
- Steinmeyer (2015). Rolled ball screws - the alternative in linear technology. Catalog.
- Yoshida, K. and Wilcox, B. (2008). Space robots and systems. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, chapter 45, pages 1031–1063. Springer.

Glossary

List of Acronyms

| | |
|-----------------|---------------------------------|
| CoM | centre of mass |
| COTS | commercial off-the-shelf |
| DoF | degree of freedom |
| emf | electromotive force |
| EoM | equations of motion |
| FD | forward dynamics |
| GRF | ground reaction force |
| ID | inverse dynamics |
| IIT | Istituto Italiano di Tecnologia |
| RoG | radius of gyration |
| RoM | range of motion |
| SEA | series elastic actuation |
| TU Delft | Delft University of Technology |