PHD

Foot placement for running robots

Bhatti, Jawaad

Award date:
2016

Awarding institution:
University of Bath

Foot placement for running robots

submitted by

Jawaad Bin-Sajid Bhatti

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mechanical Engineering

January 2016

COPYRIGHT

Signature of Author ....

Jawaad Bin-Sajid Bhatti

Summary

Rubble-strewn corridors, stairs and steep natural terrain all present a challenge for wheels and tracks. Legs are a solution in these cases because foot placement allows the traversal of discontinuous terrain. Legged robots, however, currently lack the performance needed for practical applications. This work seeks to address an aspect of the problem, foot placement while running.

A novel hopping height controller for a spring-loaded legged robot is presented. It is simple and performs well enough to allow control of the ballistic trajectory of hops and therefore foot placement. Additionally, it can adapt to different ground properties using the result from previous hops to update control gains. A control strategy of extending the leg at a fixed rate during the stance phase and modulating the rate of extension on each hop was used to control the hopping height. The extension rate was then determined by a feed-forward + proportional control loop. This performed sufficiently well allowing the ballistic trajectory of hops to be controlled.

In simulation, the spring-loaded inverted pendulum (SLIP) model was extended to include actuation and losses due to friction. The control strategy was developed using this model then, in a planar simulation, the controller was run to perform foot placement while running over a series of platforms which vary in their horizontal and vertical spacing.

To experimentally validate and further develop the control strategy, a one-legged hopping robot, constrained to move vertically, was used. The leg had 2 links, hydraulically actuated hip and knee joints and a spring-loaded foot. Results showed that the controller developed could be used to perform hops of randomly varying size on grounds with different properties and while running on a treadmill at different speeds.

As an aside, the dynamics of hydraulic actuators presented a problem for foot repositioning during flight using a simple PID controller. This was solved through the novel implementation, in hydraulics, of a ‘zero-vibration’ (ZV) filter in a closed-loop. Simulation and experimental results demonstrating this are presented.

Acknowledgement

I am grateful to my supervisors Prof. Andrew Plummer, Dr. Pejman Iravani and Prof. M. Necip Sahinkaya for their expertise and guidance. Thanks are also due to the Istituto Italiano di Technologia for generously donating the HyQ robot leg used in this work.

Finally, thanks are due to my sisters, to my mum and to my dad for their infinite love and support.

Notation

Symbol	Description
c	Damping coefficient
d_i	Displacement of actuator i
$\mathbf{d}$	Vector of actuator displacements i.e. (d_1, d_2)
f	General function
g	Acceleration due to gravity: $9.81\ \mathrm{m\,s^{-2}}$
$\mathbf{g}$	Gravity vector: $(0, -g)\ \mathrm{m\,s^{-2}}$
h_i	Height at apex where i can be b for body or f for foot
k	Spring stiffness
K	Control gains e.g. PID: K_P, K_I, K_D
L	Spring length
L_0	Unstrained spring length
m	Mass
n	Hop count where nth hop begins at touch-down (inclusive)
$\mathbf{p}$	Position for example $\mathbf{p}_f$ is the foot position
P	Pressure
q	Leg extension velocity during stance
Q	Flow rate
$\mathbf{r}$	Relative position e.g. body relative to foot $\mathbf{r}_{bf} = \mathbf{p}_b - \mathbf{p}_f$
u_n	Velocity horizontally at touch-down on nth hop
v_n	Velocity upwards at touch-down on nth hop
V_i	Control voltage for ith valve
$\mathbf{V}$	Vector of control voltages: (V_1, V_2)
X	Spring extension
α, β	Cartesian foot velocity control coefficients
δ	Gain tuning rate coefficient in Eq. 6.28 and Eq. 6.29
ζ	Damping ratio
θ_{td}	Angle of leg at touch-down
θ_{pd}	Neutral touch-down angle to maintain steady forward velocity
φ	Angle swept by leg between touch-down and lift-off
ω_n, ω_d	Angular frequency: natural and damped

Contents

Chapter 1

Introduction

1.1 Motivation

Legged animals have found their footing on nearly the entire surface of our world. This ubiquity implies that legs are an effective way to cross rough, natural terrain. It may therefore be worthwhile to develop machines that can walk and run well.

The absolute size of a legged machine or animal affects its dynamics. This is because the relative strength of physical forces changes with size. Electrostatic forces, fluid viscosity and energy storage density can become dominant concerns at the small scale of insects, less than 10^{-2} m. Whereas for bigger animals and machines, greater than 10^{-1} m, inertial forces and structural stresses become more important. The scope of this work will be limited to mammal-like or mammal-sized machines. Such robots are the stuff of science fiction but pragmatic reasons for pursuing this technology are emerging. In agriculture and forestry, machines have been built with careful foot placement in mind to avoid damage to a field that might otherwise result from driving over it. In defence, machines are being field tested to see if they permit supply chains to cross natural terrain; and, perhaps banally, to realistically test the durability of hazard suits. In the exploration of Earth and space, a low gravity hopper was deployed on a moon of Mars. In service robotics and disaster response, development work is ongoing to make legged machines able to negotiate obstacles such as stairs found in human environments.

Any practical legged machine requires a coherent design of the mechanics, electronics and software. The legs of such a machine consist of masses, inertias, actuators and elastic components with parameters tuned to give walking or running dynamics that are efficient and controllable. These components will be subjected to periodic impacts, free swinging and loading phases so actuators capable of handling this are required. Energy will be constantly transformed between gravitational, elastic and kinetic forms during locomotion. This has to be choreographed by a computer controller taking input from a suite of sensors. Simply balancing requires proprioception and inertial sensors. To do something useful additional sensing and computational power is necessary. For example one of the main areas where legs might be an advantage over other modes of transport would be when traversing rough terrain. To achieve this requires the ability to map some of the surrounding terrain to avoid obstacles. Rougher terrain might require the ability to make some sense of the local environment, detect and dynamically plan foot placement, and react to slips and trips.

Designing and building a practical legged machine is a complicated and interesting mechatronics challenge. This work is being simplified and catalysed by a number of ongoing developments:

- Robotics in general is benefiting from the accelerating development of computer technology and consumer electronics. This has made digital self-contained sensors of various kinds available: high-resolution cameras, depth cameras and inertial sensors for example.

- Computer vision is becoming easier to implement. This is likely to be a necessity for many legged locomotion applications where there is a need to detect obstacles and avoid unsuitable foot placement spots. The visual effects and video games industries have driven the development of graphical processors which are now being used for computer vision and SLAM (simultaneous location and mapping). Meanwhile the open-source movement has simplified algorithm development.

- There has been a long running interest in legged machines from the US military. The Defence Advanced Research Projects Agency (DARPA) continues to sponsor much research in the area. DARPA is currently incentivising research by holding competitions with multiple millions of dollars of prizes.

1.2 Hypothesis

Aside from the work required to build a practical machine, the fundamental mechanics problem of how to structure and control a legged machine requires further work. It will be seen in the literature review chapter of this thesis that researchers have not converged onto one paradigm. The contrast is perhaps greatest between walking and running. The focus of this thesis will be on an aspect of the fundamental mechanics: how to control foot placement while running? To be more specific, the following research hypothesis is made and supported:

> The foot placement of a robot designed for steady-state controlled passive dynamic running can be controlled by appropriately moving actuators during the ground contact phase.

The concept of 'controlled passive dynamics' is a paradigm of walking and running. This paradigm has been successfully applied to construct efficient and impressive walking and running machines discussed in more detail in the literature review chapter. It also has explanatory power when applied to animal locomotion. Controlled passive dynamic approaches take the following general form:

- A machine which walks or runs indefinitely without requiring control or an energy source can be conceived theoretically.
- A real machine based on this concept would be affected by energy losses and disturbances meaning it would not walk indefinitely like the theoretical model so actuators are added.
- A method of control is then devised to maintain steady locomotion indefinitely by actuating the system to excite the underlying passive dynamics, compensate for losses and reject disturbances.

This paradigm has largely been applied to achieve steady running but the work here will focus on how to make quick and accurate changes in the trajectory of a passive dynamic runner. This will sometimes be referred to as agile locomotion.

1.3 Aims and objectives

The objective of this work is to develop a control method for modulating the action of the foot during the stance phase to achieve a desired flight duration. It is argued that if this is coupled to a forward velocity control method then the size of hops, and therefore foot placement, can be controlled. This argument will be supported by analysis and simulation work as well as by experiments with a hopping leg on a treadmill. A limitation of the work done here is that the control techniques are not comprehensively demonstrated on an untethered, unconstrained field robot. This is left for future work.

1.4 Contributions

The main contribution of this work is a novel method for controlling the flight duration of a running robot. Specific original contributions to knowledge are contained in each chapter. These are summarised below:

- Chapter 2 presents a review of legged robotics work is undertaken. It is shown that researchers have approached the problem of legged locomotion from different starting points. This has led to machines with different mechanical and controller designs, with different strengths and limitations. Areas requiring further research are identified including agile locomotion.

- Chapter 3 extends the spring-loaded inverted pendulum (SLIP) running model to develop a novel method for controlling the ballistic trajectory of a running machine on each hop. The basic method is extended to adapt to different ground properties. With it, foot placement control in 2D simulations is also shown.

- The hydraulically actuated spring-loaded leg used to experimentally validate work done in this thesis is presented in chapter 4. During the flight phase of each hop, the foot has to be quickly repositioned before touchdown. A method for achieving this without vibrations, namely closed-loop signal shaping (CLSS), is developed in chapter 5. Implementation of CLSS with hydraulics is novel.

- Chapter 6 experimentally demonstrates how foot motion during stance can be modulated to perform hops with varying flight duration demands. The demanded flight duration can be met even if varied randomly on each hop. Results show the controller can self-tune to adapt to changes in ground properties. By running on a treadmill, it is shown how the technique can be made to work over a range of running speeds.

1.5 Publications

The following publications have been made as a result of this research:

- J. Bhatti, P. Iravani, A. Plummer, and M. N. Sahinkaya, “Instantaneous control of a vertically hopping leg’s total step-time,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on.* IEEE, 2013, pp. 1–6.

- J. Bhatti, A. R. Plummer, P. Iravani, and M. N. Sahinkaya, “Implementation of closed loop signal shaping in a hydraulic system,” in *The 13th Mechatronics Forum International Conference*, 2012.

- J. Bhatti, P. Iravani, A. R. Plummer, and M. N. Sahinkaya, “Towards running robots for discontinuous terrain,” in *Advances in Autonomous Robotics.* Springer Berlin Heidelberg, 2012, pp. 461–462.

- J. Bhatti, A. R. Plummer, M. N. Sahinkaya, P. Iravani, E. Guglielmino, and D. G. Caldwell, “Fast and adaptive hopping height control of single-legged robot,” in *ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis (ESDA2012)*, 2012.

- J. Bhatti and A. Plummer, “Hydraulic running robots: the prospects for fluid power in agile locomotion,” *Proceedings of the twelfth Scandinavian international conference on fluid power*, 2011.

Chapter 2

Background to dynamic legged robots

This chapter looks at the literature on legged robots. It focusses largely on robots which are mammal-like and mammal-sized. Legged machines which have been built but intentionally neglected here include: robots for climbing; those which are small insect-like or with multiple legs; those very large, having a mass on the order of 1000 tonne. The design and control of such machines differs from the ones mentioned in this section.

The goal of this chapter is to:

- Summarise what has been achieved by roboticists working on legged locomotion.
- Understand how running and hopping robots are designed and controlled.
- Identify areas of research.

A review of the literature on legged machines shows that legged locomotion is not one line of research. Different paradigms prevail among walking and running robots, for example. These can then be subdivided further. Humanoid walking bipeds can be under precise kinematic control in which trajectories are generated offline and online for each joint. They may also be based on passive dynamic walking mechanisms in which joints are underactuated and stability is maintained through small control actions. Successful running robots have utilised elastic components to a greater degree in order to store energy from one step to the next. The design and control of running robots varies too. They can have multiple actuators per leg in order to control the footfall of each step or be self-stabilising and underactuated with only one actuator per leg. They can have one telescopic spring or a biomimetic set of articulated links and springs. Although different paradigms have prevailed in these lines of research there is also some cross-fertilisation.

Thus far researchers have largely focussed on steady-state locomotion so the subject of agile running is identified for this thesis. This involves developing control techniques to allow a running robot to rapidly execute desired changes in the direction and size of its stride.

2.1 History of legged robot research

The desire to mimic the form and motion of animate beings – to build 'automatons' – goes at least as far back as the ancient Greeks and Chinese. Around 480 BCE the Chinese master carpenter Lu Ban invented a mechanical horse to entertain his mother. On a slope, the four-legged wooden horse and carriage would move down on their own [1]. The ancient Greeks sparked an interest in automatons which continued into the Middle Ages and experienced a revival during the Renaissance. A good survey of early legged machines can be found at cyberneticszoo.com [2]. The earliest walking machines consisted of cam and linkage based mechanisms [3]. This restricted them to walking with a fixed gait.

In 1968, under the supervision of Bob McGhee and Rajko Tomovic, Andrew

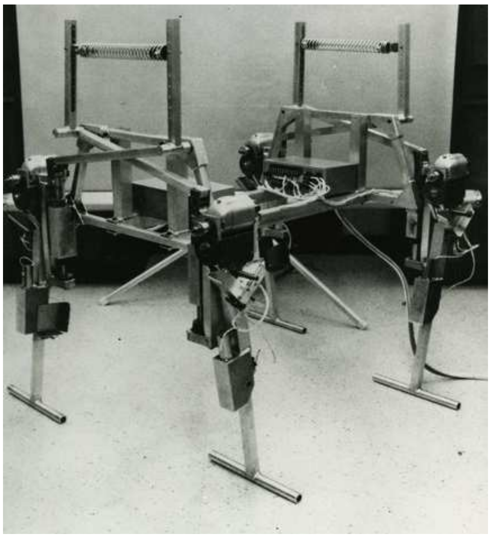


Figure 2-1: Phoney Poney also called the Californian Horse [4].

Frank built the “Phoney Pony”, which can be seen in Fig. 2-1. This quadruped was capable of moving very slowly with statically stable crawl, walk and trot gaits. The Phoney Pony was an interesting development because it employed computer control before microprocessors. Joint states were employed in state-machine feedback control with the computer sitting in an adjacent building.

In 1968 General Electric made a Walking Truck, Fig. 2-2, also known as the ‘Cybernetic Anthropomorphous Machine’ (CAM). This was an experimental hydraulic quadruped vehicle built for rough terrain locomotion for the US Army. The walking truck weighed 3000 lb and could achieve $2.2\,\mathrm{m\,s^{-1}}$ (5 mph). A human operator suspended within the vehicle would use both arms and legs to interface with force-feedback controls. The coordination required for locomotion with four legs could be achieved but this became mentally strenuous for the operator after 15 min. Nevertheless, a legged vehicle capable of changing gait had been built.

In 1985 Ohio State University, with DARPA funding, built the prototype Adaptive Suspension Vehicle (ASV) [6]. The ASV was a hexapedal vehicle featuring a computer controller. Other features included force controlled hydraulic servos and flywheel regenerative braking. Each pantograph leg had 3 hydraulic

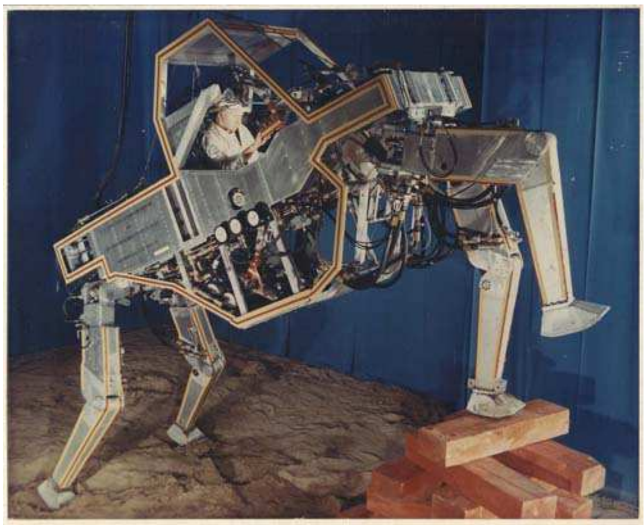


Figure 2-2: General Electric Walking Truck or Cybernetic Anthropomorphous Machine, CAM. [5]

actuators and each actuator fed back position, velocity and differential pressure signals to the computer controller. The ASV could be switched between different modes by the operator. In a basic locomotion mode, closed loop control of the leg positions was used to produce a fixed alternating tripod gait. In the most advanced mode, a scanning rangefinder was used to determine foot placement locations and, instead of a fixed gait, an algorithm was used for terrain adaptive foot placement. The controller acted to maintain the body orientation parallel to, and a fixed height above a smooth average slope. Foot placement adapted to terrain to filter the body from short-wavelength variations in the terrain. The operator input desired body forward velocity, sideways velocity and rate of change of heading through joystick signals. The ASV had on-board power from a 50 kW motorcycle engine, weighed 2700 kg and had a maximum speed of 3.6 $\mathrm{m\,s^{-1}}$. Computer coordination of the legs left the operator in charge of navigation and path planning.

Merely aiming to maintain static stability restricts robot locomotion to slow, constant velocities and designs with large bases. A significant development in

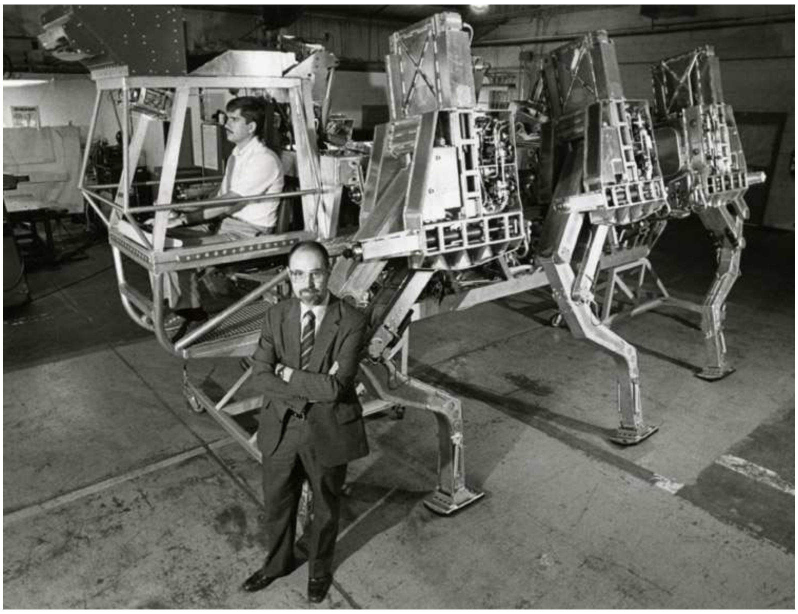
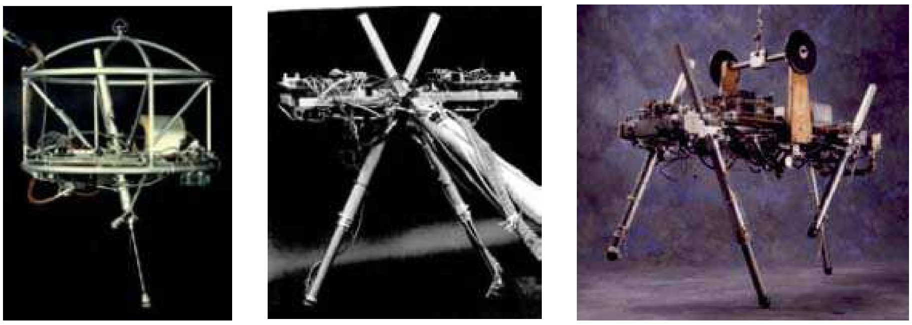


Figure 2-3: Ohio State University's Adaptive Suspension Vehicle (ASV), nicknamed the "Walker". [7]

legged robotics were the first running robots [8]. Running robots, those with a flight phase as part of their gait, were first built in the 1980s and early 1990s at the MIT Leg Lab founded by Marc Raibert. These robots featured legs with springs endowing a tendency to hop by storing and releasing energy during ground contact. The Leg Lab began by building one-legged hopping machines with a point contact foot. Such a machine cannot remain statically balanced. A '3-part' computer control method was used to maintain dynamic balance while hopping. Leg repositioning in-between hops, thrusting downwards during ground contact and applying hip-torques during ground contact were used to control the running velocity, hopping height and body orientation respectively. This basic control technique was extended successfully from one- to two- and four-legged running robots (Fig. 2-4) [9]. The Leg Lab robots had off-board power so were restricted to move mostly on a flat lab floor. Nevertheless, they could maintain balance when disturbed and achieve speeds of up to 5.8 $\mathrm{m\,s^{-1}}$ (13 mph). The quadruped shown in Fig. 2-4(c) could run with a trotting, pacing or bounding gait. A 3D biped



(a) (b) (c)

Figure 2-4: Leg lab robots.

could even perform somersaults without losing balance. Raibert's book 'Legged Robots That Balance', published 1986, remains authoritative on the subject of hopping and running robots [9]. Other notable Leg Lab robots include Jerry Pratt's Spring Flamingo, a planar biped walker with series-elastic actuated hip, knee and ankle. The Leg Lab robots are an example of how relatively simple control laws can be used to manage complex dynamics.

Marc Raibert and his MIT colleagues have gone on to continue the work they started at the Leg Lab by founding the company Boston Dynamics in 1992. In 2005 Boston Dynamics, working with Foster-Miller, NASA's Jet Propulsion Lab and Harvard University, unveiled their robot 'BigDog' [10–15]. BigDog, Fig. 2-5, is a four-legged robot with a dynamically balancing gait able to traverse a variety of outdoor terrain and recover from disturbances. The project was funded by DARPA and the US Marine Corp who are interested in rough-terrain platforms for carrying equipment and weapons. BigDog actively balances and reacts to disturbances to maintain dynamic balance in a way that looks uncannily similar to quadruped mammals. Though there are many nuances as to how this is achieved, the underlying concepts are those developed at the MIT Leg Lab in the 1980s. BigDog weighs 109 kg and is approximately 1 m tall. It is hydraulically actuated and powered by an on-board 15 hp internal combustion engine. Each actuator is fitted with a load-cell and position sensor. Each leg has 4 actuated and

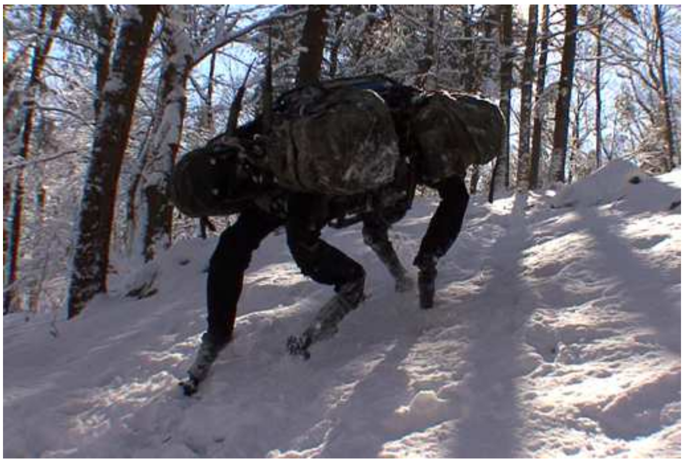


Figure 2-5: BigDog [11].

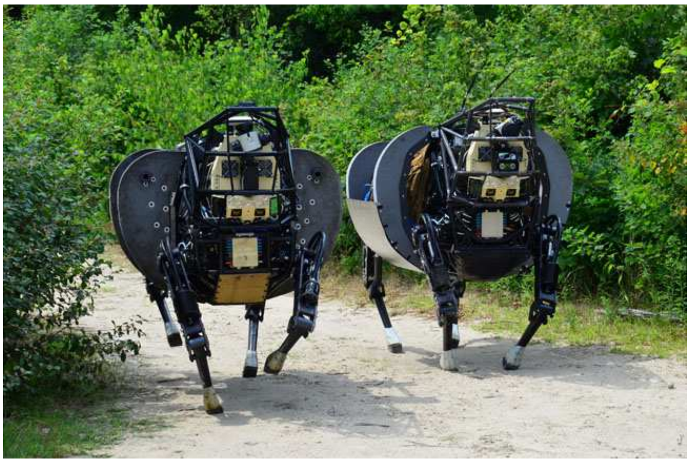
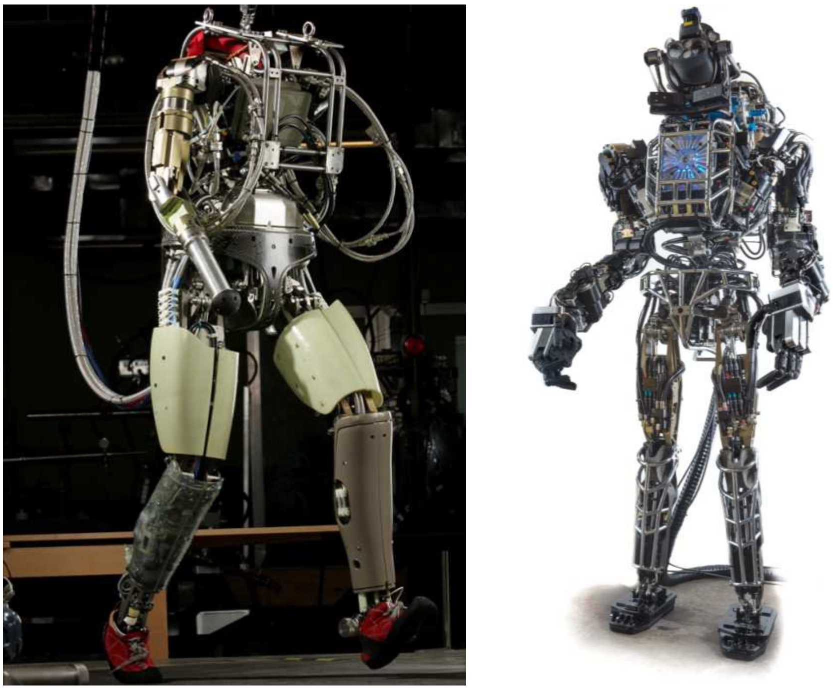


Figure 2-6: AlphaDog [11].

1 passively compliant degree of freedom. Alongside proprioception and an IMU, BigDog also has LIDAR, stereo vision and GPS to allow autonomous navigation and obstacle avoidance. Boston Dynamics' successor to BigDog is the AlphaDog also called the Legged Squad Support System (LS3), Fig. 2-6. AlphaDog is a more rugged version of BigDog currently being field tested for military use. It is quieter, able to carry a greater payload, has greater range and able to self-right after falling over.

Some of the most advanced and dynamic mammal-like legged machines today



(a) (b)

Figure 2-7: PETMAN (a) and its derivative ATLAS (b) [11]. As part of the DARPA Robotics Challenge, Boston Dynamics was commissioned to build 6 ATLASs at a cost of $10.9 million.

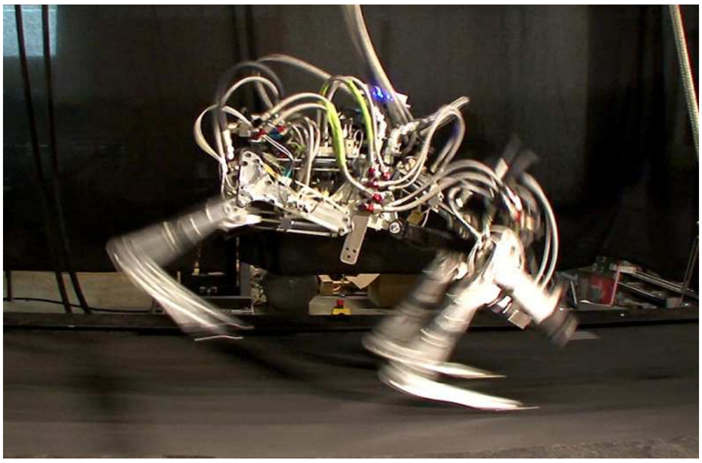


Figure 2-8: Boston Dynamics' Cheetah robot [11].

are Boston Dynamics' AlphaDog, PETMAN and Cheetah. PETMAN, shown in Fig. 2-7(a), is an anthropomorphic robot that walks and performs movements in order to stress test clothing designed to protect against hazardous chemical exposure. PETMAN balances itself and simulates the temperature, humidity and sweating of a person [11]. The Cheetah, shown in Fig. 2-8, is a planarised, tethered robot that can achieve 29 mph running on a treadmill. Its successor, the WildCat, will aim to achieve high speed running outdoors and untethered.

Among animals, bipedal walking is relatively rare but humanoid bipedal robots have attracted researchers, especially from Japan, since the 1970s. Walking – locomotion in which at least one foot is always in contact with the ground – presents a particular challenge because a high centre of gravity and small base of support results in a tendency to tip over and fall. Humans find walking intuitive making it easy to underestimate its complexity. Firstly, walking requires the coordination of multiple redundant degrees of freedom. Active balance is also required because static stability is insufficient in order to walk with a reasonable speed and robustness, even in a straight line on a flat lab floor. In practical applications walking surfaces are likely to be slippery, uncertain, cluttered and discontinuous. Negotiating obstacles such as stairs, narrow footpaths, crowds and outdoor terrain requires planning, self-awareness, communication and learning. Bipedalism research is a window into intelligence in general. One motivation behind research into humanoid robots is that they could function as generalists in human environments. They could use tools and negotiate environments designed for humans, working alongside people and freeing them from menial labour. It is also thought people would find communication and collaboration with humanoid robots more intuitive and comfortable. A well-known humanoid bipedal walking robot is Honda's Asimo. Asimo stays balanced while walking or climbing stairs by playing back optimised recordings of humans which are then modulated by applying Zero Moment Point Control (ZMP) in order to maintain balance [16].

Defence, commercial and academic interests are continuing to advance the development of legged machines. In December 2013 DARPA hosted the first round of their robotics grand challenge to catalyse the development of semi-autonomous robots. The challenge provides prize money and an obstacle course which includes: driving a vehicle; traversing rough terrain; opening a door; using power tools; and locating and closing a valve. For the challenge DARPA commissioned Boston Dynamics to build a number of Atlas robots, Fig. 2-7. This was provided to some of the competing teams. Around the same time as the start of the challenge Google acquired Boston Dynamics. Since then Google has acquired several teams that have done well in the DARPA challenges.

2.2 Balancing while walking

The problem of remaining balanced while walking has been approached in a few different ways:

- In position controlled robots by employing so-called 'Zero Moment Point' (ZMP) control to generate dynamically balanced walking trajectories. [16]

- In force controlled robots such as the Spring Flamingo using the concept of

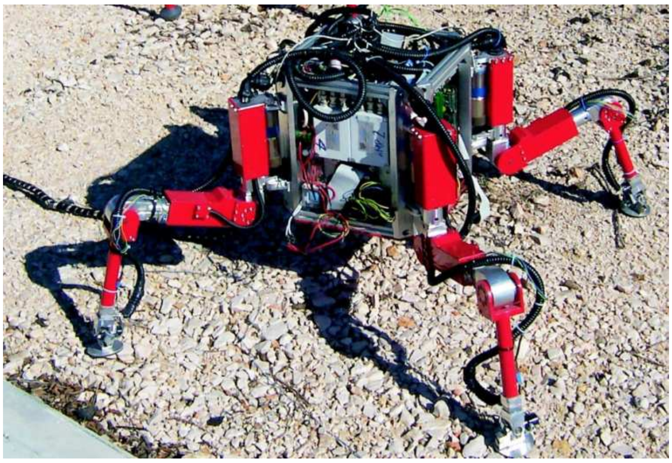


Figure 2-9: SILO4, an example of a statically stable robot [18].

Virtual Model Control [17] .

- Using the concepts of controlled passive dynamics .

2.2.1 Static stability and ZMP-CoP control

A fundamental task for legged machines is maintaining some sort of balance. One approach to this is to keep the robot positioned so that it always maintains static balance. This is a purely kinematic problem and static balance can be achieved by keeping the centre of mass (CoM) directly above a support region which can be determined as explained in [19]. For the case of level, flat terrain the support region is the convex envelope for the contact points between the ground and the robot. The SILO4 robot shown in Fig. 2-9 essentially moves in this way. As long as the robot moves slowly enough that inertial forces can be ignored, kinematic control to maintain static balance will be sufficient to prevent tipping over. In order to allow movement at higher speeds, the controller can be extended to keep a certain safety margin between the edges of the support region and the projection of the CoM on the ground. This is feasible for SILO4 because its feet form a large support region (relative to the height of the CoM). Maintaining static balance becomes much more challenging for robots with a small support region. For bipeds, having a small support region means disturbances can more easily knock them off-balance. The earliest biped robots walked by maintaining static balance. The small base and high CoM of bipeds restricts them to a very slow walking speed if control is implemented with a view to maintaining static balance. For example, the first of Honda's E-series bipeds, the E0 in 1986, could only walk in a straight line and take a step every 30 s [20]. The E1 had improved mechanical design but still only achieved 0.07 $\mathrm{m\,s^{-1}}$ (0.25 $\mathrm{km\,h^{-1}}$).

Pseudo-statically stable walkers have been built which, when swinging one foot forwards, fall out of static stability and tip forwards. Stability is then regained through passive stabilisation as the robot lands on the other foot.

On flat ground, a robot with no acceleration will tip over if the point at which the weight vector intersects the ground lies outside the support envelope. For an object with acceleration, tipping will occur if the projection of the resultant of the inertial and gravity forces (and all external forces) lie outside the support envelope. This projected point is the so-called Zero Moment Point (ZMP). The Centre of Pressure (CoP) is the point where the resultant ground reaction force intersects the plane of support. The CoP lies within the support envelope. It can be shown that the ZMP and CoP coincide [21]. If the robot trajectory is designed to maintain the calculated ZMP-CoP within the support envelope, tipping will be avoided. The concept of ZMP-CoP has been widely used [22]. However, a limitation is that it is only defined for flat terrain. Less clearly defined extensions to the concept, virtual- or pseudo-ZMP-CoP, have to be made in order to extend the concept to non-flat support planes. ZMP-CoP gives a condition to prevent tipping but various walking trajectories can satisfy it. Furthermore, mere playback of trajectories is not robust against disturbances. One approach is to record and optimise trajectories from humans to increase stability based on ZMP-CoP concept then modulate playback in order to maintain balance against

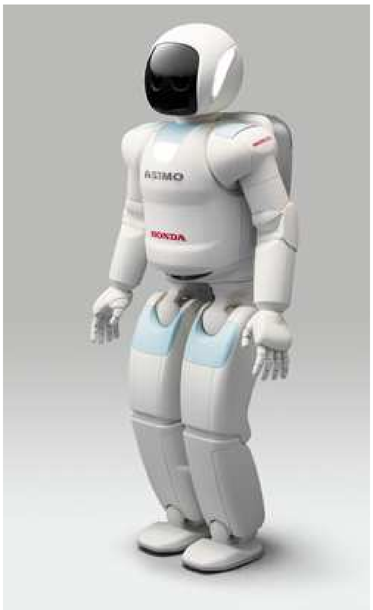


Figure 2-10: Honda humanoid robot, Asimo [23]. Asimo weighs 48 kg and is 1.3 m tall.

disturbances. This seems to be the approach taken by Honda who have been researching humanoid bipeds since 1986. The Honda E2, 1989, employed dynamic balance using the ZMP-CoP concept and the E4, 1991, could walk at 1.3 $\mathrm{m\,s^{-1}}$ (4.7 $\mathrm{km\,h^{-1}}$) [20]. Further development means Honda's latest 2011 Asimo biped (Fig. 2-10) can negotiate stairs, turn, run at 2.5 $\mathrm{m\,s^{-1}}$ (9 $\mathrm{km\,h^{-1}}$), walk over slightly uneven terrain and maintain balance against some external disturbances. Asimo is one of the most advanced humanoid bipeds but it cannot yet negotiate outdoor environments and rough or cluttered terrain.

2.2.2 Virtual model control

The concept of 'Virtual Model Control' (VMC) was applied by Jerry Pratt [17] on the Spring Flamingo and Spring Turkey bipedal, planar walking robots. In VMC, control laws for walking and balancing are conceptualised as virtual mechanical components which act to keep the robot balanced. For example Fig. 2-11 shows how a virtual walking trolley keeps the robot balanced in the sagittal

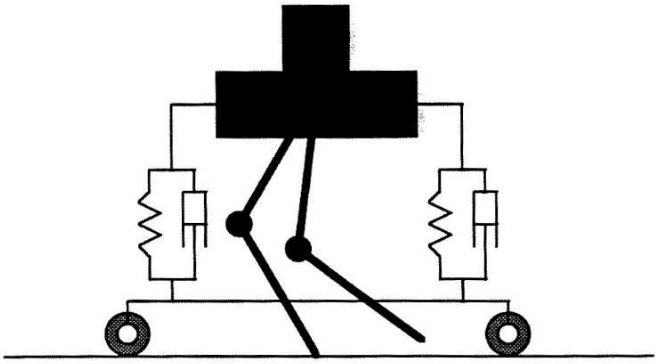


Figure 2-11: Virtual Model Control [24]. The control action for maintaining balanced is conceptualised as a virtual walking trolley. Force-controlled joints then emulate the virtual spring-dampers. Parameters for the virtual components are manually tuned.

plane. The joints of the robot are series elastically actuated so they can operate in a force-control loop which is necessary in order to emulate the virtual components. Moving the robot forwards was similarly conceptualised using virtual components.

2.2.3 Controlled passive dynamic walking

Passive dynamic walkers are unactuated mechanisms of links and joints whose parameters have been carefully selected through analysis, simulation and tuning so they can stably walk down slopes with a gait that resembles human walking. They were first studied by McGeer [26]. There are a variety of such mechanisms with differently designed joints at the knees and ankles. Because they have no actuators or control electronics, passive dynamic walkers cannot walk on level ground and have limited ability to remain stable on varying terrain.

A number of researchers have added actuators to 'passive' dynamic walkers to build walking robots (Fig. 2-12). This gives the advantage of a human-like walking gait and a specific mechanical cost of transport comparable to human walking, an order of magnitude reduction over bipeds such as Asimo [25]. In

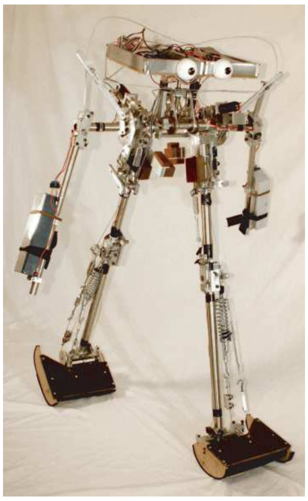


Figure 2-12: The Cornell biped, an example of a controlled passive dynamic walking robot. [25]

this thesis, the paradigm of beginning with a legged mechanism tuned for passive dynamics and then actuating it will be referred to as controlled passive dynamics (CPD). CPD walkers are not merely passive walkers with the ability to walk indefinitely. The addition of actuation may give insights to resolve important questions about the role of different muscles in human walking. The role of ankle push-off in human walking, for example, is somewhat controversial but important in the design of powered prosthetics. This may be resolved by building and studying controlled passive dynamic walkers [25, 27].

2.3 Balancing while running

2.3.1 3-part running control and virtual legs

The one-legged robot shown in Fig. 2-4(a) is incapable of static balance because there is practically no support region. Instead the robot must hop to stay upright. The dynamic balancing problem cannot be approached in this case as an extension to static balancing. It has been tackled most effectively by Marc Raibert starting at the MIT Leg Lab in 1980 [9]. The monopod robot has a springy leg giving the robot a natural hopping motion.

The mechanics of a one-legged hopping robot are non-linear and difficult to analyse mathematically so the problem is simplified and tackled as if 3 aspects can be decoupled reducing it to 3 simpler control problems:

- Control of the hopping height. Hopping can be achieved by exciting a leg spring in series with a telescopic leg actuator. By applying the appropriate vertical thrust or displacement during the stance phase, it is possible to input energy vertically and thus control the hopping height. Actuators act to add vertical energy to the system in order to maintain a steady hopping height.

- Control of the horizontal velocity. The horizontal velocity of the robot after a hop is affected by the angle of the leg before impact. A control loop can be set up which adjusts the leg angle during flight in order to increase or decrease the horizontal velocity and maintain balance.

- Control of the body orientation. During stance, by applying a hip torque it is possible to control body orientation and remove angular momentum.

This dynamic balance approach to locomotion begins with a mechanism with a natural repetitive hopping or stepping dynamic. Actuators then perturb the system in order to sustain and modify the natural motion. The three-part controller works under the assumption and to the extent that these three components are largely uncoupled. During the 1980s and early 1990s the Leg Lab built increasingly sophisticated dynamic running machines including [9], [28]:

- Planar one-legged hopper implementing the three-part controller.

- 3D one-legged hopper with modified and extended three part controller, Fig. 2-4(a).

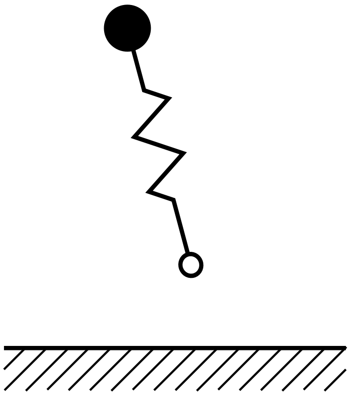


Figure 2-13: SLIP model.

- Planar biped capable of running and somersaults, Fig. 2-4(b).
- Quadruped which could run with several gaits including trotting, pacing and bounding Fig. 2-4(c).
- 3D biped capable of running and performing somersaults.

Bipedal running at the LegLab was essentially achieved by applying the one-legged control scheme but alternating between legs in the flight phase. For this reason bipedal running is referred to as a "one-foot gait" in [9].

In order to achieve running with a quadruped, the concept of a "virtual leg" was developed [29]. In gaits such as the trot, pace and bound, where sets of legs enter stance simultaneously and different sets do not overlap in their stance phase, the sets can be treated as though they were one "virtual leg". In this way, quadruped locomotion can effectively be achieved by using the one-leg three-part controller with some extensions in order to control body pitching and rolling.

2.3.2 Self stabilisation

Self-stabilisation is a feature of some gaits which reject disturbances to a steady gait without any feedback control action being required. Whether a gait is self-stabilising depends on the model and the particular parameters. Fig. 2-13 shows

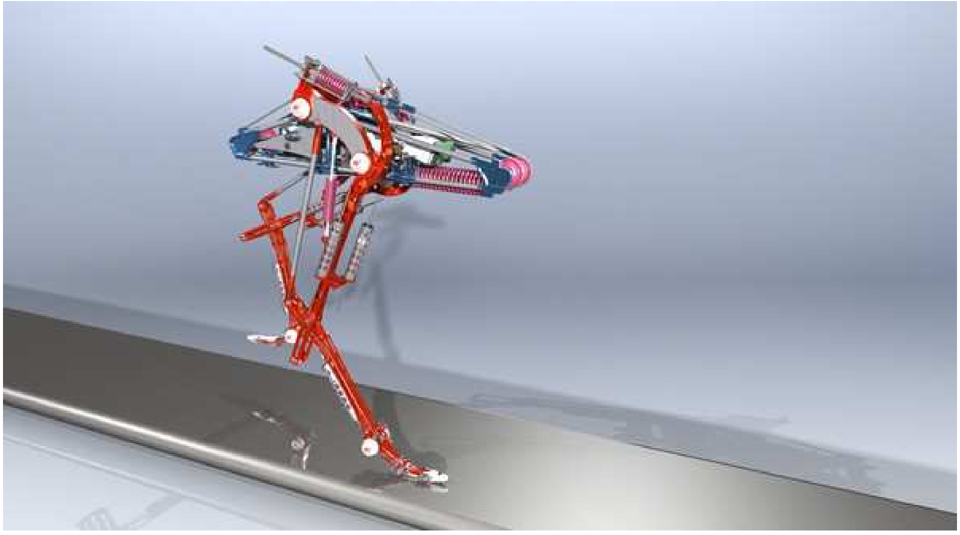


Figure 2-14: IHMC FastRunner. A DARPA funded project. Achieves stable running in simulation at 22 mph. [30]

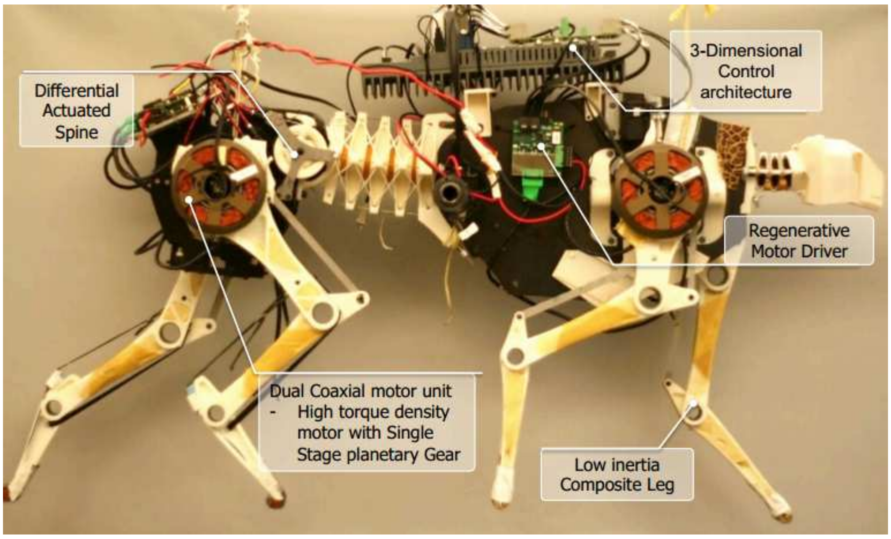

Differential Actuated Spine
3-Dimensional Control architecture
Regenerative Motor Driver
Dual Coaxial motor unit
- High torque density motor with Single Stage planetary Gear
Low inertia Composite Leg

Figure 2-15: MIT Cheetah. [31]

the SLIP (spring-loaded inverted pendulum) model of a running animal or robot. The model consists of a point mass hopping on a spring. The point mass represents the body and the spring represents a leg. Self-stabilisation even in this simple model is not well understood. Simulations of the SLIP model conducted as part of this thesis show that for non-dimensional parameters similar to those of human running, the SLIP model is self-stabilising. This suggests that the mechanical design of running machines, the distribution of masses and stiffnesses of springs, could not only make feedback control easier but remove the need for it altogether in certain circumstances. Indeed robots built to be biomimetic, such as the MIT Cheetah [31] or IHMC FastRunner [30], turn out to require no control action to remain balanced while running. Although control is simplified, mechanical design becomes much more important. Both robots have numerous leg segments, biomimetically placed elastic components and only one-actuator per leg making them highly underactuated. In the case of the FastRunner, a sinusoidal input is provided by the hydraulic actuators which excites the entire running motion. This is a very different paradigm to that followed for many bipedal humanoid robots, such as Asimo, where joints are under full control.

Passive dynamics and elastic components mean that actuator power in the FastRunner is efficiently used to excite the running motion and not lost to do work against limb inertia. The FastRunner is however only efficient and self-stabilising around its operating speed of 20 mph. Currently, FastRunner has to be running in air above the treadmill at around its nominal operating speed. It will then be lowered and dropped onto the moving treadmill when running at the correct speed. The timing of this is important so FastRunner's foot touches down at the correct point in its gait. As of November 2015, results of the FastRunner's treadmill running experiments could not be found.

2.4 Agile running

Legs may have a significant advantage over wheels or tracks when tackling rough terrain because a continuous support surface is not required. This means that terrain with isolated footholds can be traversed if foot placement can be controlled. For slow moving rigid robots with large bases of support or multiple legs, foot placement is purely a kinematic problem. If a robot is required to cross rough terrain quickly or jump large gaps or heights then the problem involves dynamics. For a hopping robot, foot placement can be achieved by taking the right control action during ground contact in order to launch into the flight phase with a ballistic trajectory that will lead to the next foot placement spot.

Research on springy-legged hopper type robots has mainly focussed on achieving stable, steady running. The goal has been to approach a desired hopping height and running speed over a number of hops in a way that is robust to disturbances, for instance unforeseen changes in the ground height. Agile manoeuvring, meaning the ability to perform rapid changes in speed and direction as desired has been the explicit or even implicit goal of relatively few researchers on springy legged robots.

When foot placement surfaces are limited it can become necessary to vary the size of the step on each step in order to avoid poor spots. This problem has been tackled most directly by Hodgins [32]. Hodgins experimented with a biped robot featuring prismatic legs as shown in Fig. 2-16. The robot was planarised to run in circles by a pivoting boom. The legs consisted of a hydraulic actuator and pneumatic spring in series. The robot hopped on one leg at a time, alternating the leg that contacted the ground during each flight phase. Raibert's 3-part controller formed the basis of control. The thrust of the hydraulic actuator was varied in order to achieve hops of different sizes but specific details on how thrust was controlled were not provided. Hodgins cites studies suggesting runners on treadmills and long-jumpers control step-length by varying the vertical impulse

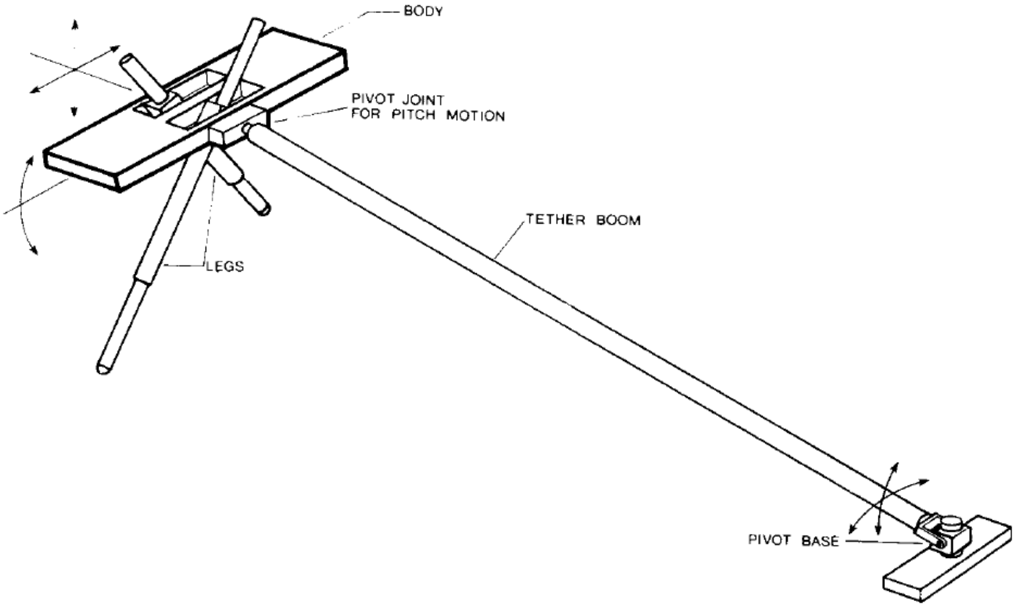


Figure 2-16: Hodgins' planar biped. [32]

imparted to the ground as well. She also cites studies suggesting that runners on flat level terrain adjust their step-length by modifying both vertical and horizontal impulse. In Hodgins' work [32], three methods for adjusting step-length are investigated. The step-length is changed by varying one of the following whilst fixing the other two:

1. Forward speed
2. Hopping height
3. Stance duration

The authors note that the best method may be to vary all three but this is not investigated in their work. When stationary, changes in the hopping height have no effect on the step-length. At low speeds, they have little effect. At that point, step-length control requires the ability to change horizontal velocity to achieve foot placement. When moving with speed, it is possible to achieve changes in step length by varying both the horizontal and vertical components of take-off velocity (the speed and hopping height effectively). Additional constraints are required for there to be one optimal solution to step-length control. Experiments conducted by Hodgins to demonstrate the step-length control methods included stepping on a pattern of ground targets and climbing up and down a flight of three stairs.

In the literature research done here, Hodgins' work seems to be the only example of an experimental implementation on a hopping robot of step-length control. Some directly following topics of further research raised are:

- Improved methods for changing the hopping height and forward running speed on each hop. In Hodgins' work, controller parameters were fine-tuned specifically to the apparatus at a specific nominal running speed and hopping height.
- Given that step-length is a function of forward speed and hopping height, how might combining the two be used to optimise performance?
- The application of hip torques during ground contact, not investigated in Hodgins' work, may also be used to vary step-length.
- How might the work be extended to a 3D robot?

Aside from Hodgins' work, no other seems to investigate step-length or foot placement with spring-loaded hopping robots experimentally. However some research has contributed implicitly. Lebaudy et al. used a 1D hopping prismatic leg consisting of a DC motor and leadscrew with mechanical spring in series to develop a hopping height controller to achieve rapid changes in hopping height [33,34]. The authors understood that the apex height of the $(n+1)$th hop is a function of the apex height at the nth hop and a 'pseudo-control signal', in their case the motor voltage V_m:

$$h_{n+1} = f(h_n, V_m) \tag{2.1}$$

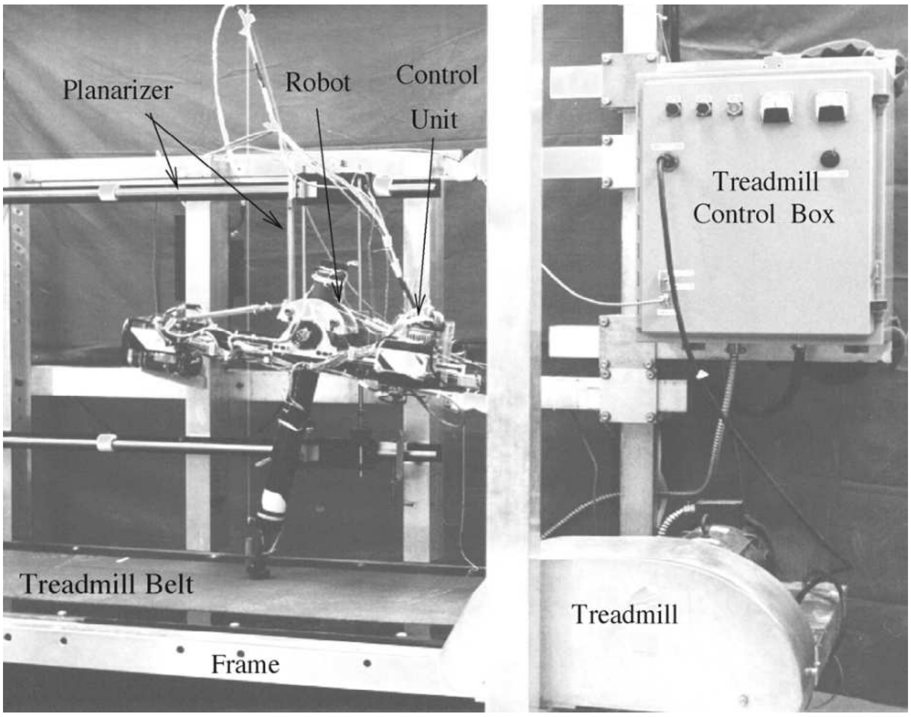


Figure 2-17: ARL Monopod 2 [35]

For the function f in Eq. 2.1, a 9 parameter, non-linear algebraic expression is fitted to a set of experimental data in which random actuation signals were output [34]. Solving for V_m in Eq. 2.1 could not be done so a 'pseudo-inverse' solution is produced. Using this pseudo-inverse function, changes in the demanded apex hopping height could be met within one hop within the limits of the pseudo-inverse function. The work done in this dissertation demonstrates a significantly simpler method which achieves similar results. See chapter 6.

Ahmadi et al. also made an implicit contribution to step-length/foot placement control [35]. The height control method employed by Ahmadi et al. may be capable of matching varying desired hopping heights on each step. The authors' purpose though was to achieve efficient steady-state locomotion so the performance of the robot when varying heights are demanded was not assessed. They built a one-legged planar running robot with a leg spring and also a hip spring. The hip spring, tuned to oscillate at the stride frequency, adds leg swinging to the robot's passive dynamics in addition to hopping. The hopping and swinging motions are excited and controlled by DC motors. The combination of electric actuation, leg spring and hip spring led to a substantial increase in the efficiency of ARL Monopod II, shown in Fig. 2-17, when compared with ARL Monopod I. Controlling hopping height was viewed in terms of controlling the system's vertical energy. Somewhat similar to Eq. 2.1, the energy at the next apex, $(n+1)$th, is the energy at the current apex, nth, plus actuator input and minus losses:

$$E_{\text{apex}}^{n+1} = E_{\text{apex}}^{n} + E_{\text{act}}^{n} - E_{\text{loss}}^{n} \tag{2.2}$$

A measure of the work done by the actuator was obtained and controlled by employing a method which integrated an estimate of the spring force with the product of the actuator velocity during stance. This control method could allow instantaneous changes in height if the values output for E_{act} and estimated for E_{loss} are accurate. The energy loss on each hop E_{loss} is adaptively updated based on previous hops' E_{apex}.

2.5 Summary

People have long been fascinated by walking machines but until the development of computer control, such machines were restricted to fixed gaits. Computer control allowed actuators to coordinate legs so that walking machines could adapt to terrain and recover balance. In the 1980s, parallel lines of research began on hopping and running legged machines. Among these, some featured elastic components which stored and released energy giving rise to passive dynamic locomotion. Passive dynamics, which is also a feature of efficient animal running, can be harnessed with relatively simple or no feedback control at all in order to achieve steady running locomotion. Legged machines continue to improve but most research on running as opposed to walking has remained focussed on improving the robustness and efficiency of steady locomotion. Running animals, however, also demonstrate the ability to quickly change direction. For example rock wallabies can negotiate rocky, mountainous terrain with apparent ease by varying the size and direction of each hop to quickly jump from one good foothold to the next while keeping an eye out for birds of prey. To emulate this ability requires the development of control algorithms which allow a legged machine designed for passive dynamic running to perform rapid and precise steps or hops. The work done in this thesis seeks to contribute here. Work will be presented on the development of an algorithm that allows a hopping legged machine to traverse terrain by hopping over a series of targeted foot placement spots. The focus will be on the basic control problem although the realisation of any useful machine would require much more including methods to sense good foot placement spots and plan routes across rough terrain.

There are a great number of different topics in legged locomotion which are ripe for further research. Some of the ones identified in this review include:

- Biomimicry
- Agile manoeuvres
- Gait transitions
- Targeted foot placement
- Field robots
- Turning
- Energy efficiency
- Under-actuation
- Speed
- Self-stabilisation

Chapter 3

Foot placement control in 2D: modelling and simulation

This chapter analyses the problem of foot placement and presents simulation results. A video of the simulation results can be seen here:

`https://youtu.be/uvrJPWUkTwQ`

The simulation model used is an extended version of the spring-loaded inverted pendulum (SLIP). An approach to height control is developed. This is then applied to perform hopping across a set of platforms which vary in their horizontal and vertical separation. The general approach to height control developed in this chapter is validated experimentally in later chapters.

First, in section 3.1, 1-dimensional models of the hopping machine as a mass-spring-damper system are used to formulate a function for control of the hopping height. In section 3.2, the 2-dimensional SLIP model is used to formulate a forward velocity controller. Finally, section 3.3 demonstrates how the height controller can be used to control foot placement when the task is to hop across isolated platforms which vary in their horizontal and vertical spacing.

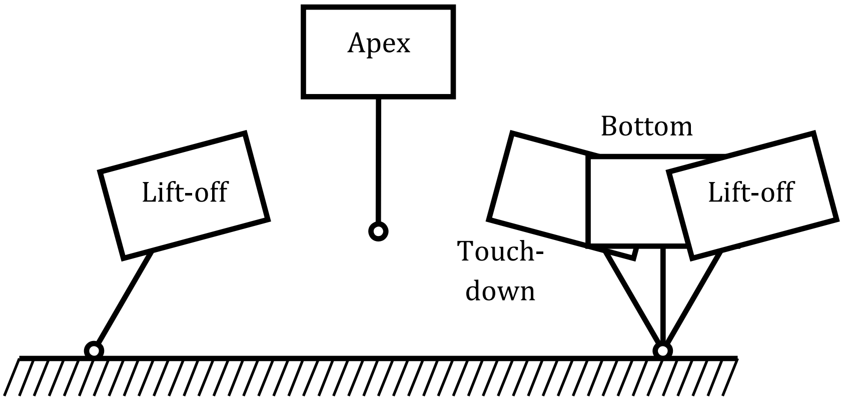


Figure 3-1: Terminology for stages of a hop. The hopper is in flight when it is not in contact with the ground, otherwise it is in stance. The nth hop begins in stance with touch-down and the $(n + 1)$th hop begins at the next touch-down.

The terminology used here for key instances during a hop is shown in Fig. 3-1. Lift-off is when the machine stops being in contact with the ground. Apex is when, during the flight phase, the CoG has zero vertical velocity. Ground contact or stance phase begins with touch-down. To count hops, the convention will be that a hop begins at touch-down.

3.1 Controlling 1D hopping

Actively balancing legged robots have been around since the 1980s when, at the MIT Leg Lab, Raibert and his collaborators developed one, two and four legged machines capable of running. These employed legs which featured a passive elastic component in series with the actuator. Significant progress was made by treating the problem as one of stabilisation and control of the passive dynamics. This was done using a 3-part controller:

1. Hopping height control: the basic hopping motion has to be excited by an actuator.

2. Horizontal velocity control: the foot has to be repositioned before landing

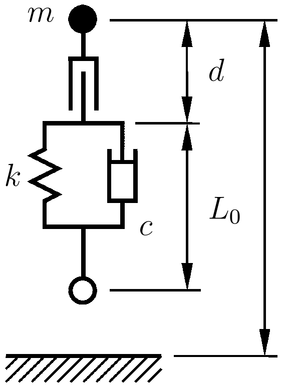


Figure 3-2: Model of a hopping machine with massless foot.

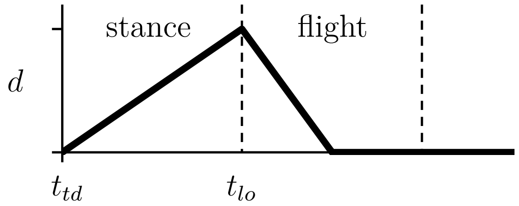


Figure 3-3: Actuator motion following touch-down. It is extended with constant velocity q during stance. The motion in flight is unimportant as long as the actuator returns to home $d = 0$ before the next touch down.

in order to maintain balance and, change speed and direction.

3. Body orientation: hip torques are applied during the stance phase in order to stabilise the body orientation.

This work will focus on improving the first of these: height control.

The problem of adjusting step length has been addressed before in work by Hodgins [32]. In it a planar, telescopic legged hopping robot which is hydraulically actuated with a pneumatic spring in series is used. The flight duration of hops is controlled by extending the actuator at different velocities. The work here provides a different analysis of the problem using spring-damper models. The experimental work in later chapters demonstrates instantaneous hopping height control using an articulated leg with a spring foot.

A model for a robot leg is shown in Fig. 3-2. Here a point mass representing the body is connected to an actuator in series with a spring-damper. The foot is modelled as a massless point. The ground has perfectly inelastic and rigid collisions with the foot. This model resembles a telescopic leg with a spring in series but can also model, for instance, the behaviour of an articulated leg with elasticity in the joints. In this 1-dimensional model, the length of the spring does not affect dynamics so can be set to zero to simplify the equations of motion giving:

$$\begin{aligned} \text{for } h-d<0\text{: } \ddot{h} &= -g - \left(2\zeta\omega_n(\dot{h}-\dot{d}) + \omega_n^2(h-d)\right) \\ \text{otherwise: } \ddot{h} &= -g \end{aligned} \tag{3.1}$$

where $d = d(t)$ is the actuator extension, $\omega_n = \sqrt{k/m}$ is the natural frequency of the mass-spring system, and $\zeta = c/\left(2\sqrt{mk}\right)$ is the damping ratio.

If the actuator is kept stationary, $d(t) = 0$, and the mass is dropped from an initial height the mass will bounce and will lose energy on each hop. Energy needs to be put into the system to maintain hopping. This can be done by moving the actuator during stance; in the case here by extending it at a constant velocity q. The actuator is then retracted back to its starting position during the flight phase ready for the next hop as shown in Fig. 3-3. The actuator must return to its home position $d = 0$ during flight before the next touch-down. The motion it takes to do this can be arbitrary because the parabolic trajectory of the body will not be affected by actuator motion.

Consider at $t = 0$ touch-down has just occurred, with $d(0) = 0$, in the stance phase. Assuming that accelerations during stance are significantly greater than g, gravity can be neglected, $g \approx 0$, and Eq. 3.1 can be simplified:

$$\ddot{h} + 2\zeta\omega_n(\dot{h}-\dot{d}) + \omega_n^2(h-d) = 0 \tag{3.2}$$

The assumption that $\ddot{h} \gg g$ and therefore g can be neglected can be checked given values for the duration of flight T_f and stance T_s. The ground touch-down vertical velocity implied by a flight duration of T_f is $v_{td} = \frac{gT_f}{2}$. For steady-state running the lift-off velocity will be the same and therefore the impulse during stance will be mgT_f. This implies a mean force of $mg\frac{T_f}{T_s}$, therefore a mean acceleration relative to g during stance of $\frac{\bar{a}}{g} = \frac{T_f}{T_s}$. For human running this ratio of flight to stance varies between 2 to 4.

The actuator will be extended at a constant velocity q throughout stance so:

- $d = qt$

Impact with the ground occurred with a speed v_1 giving the initial conditions:

- $h(0) = 0$
- $\dot{h}(0) = -v_1$

The above can be solved giving the motion during stance, $0 \leq t < t_{lo}$:

$$h(t) = -\left(\frac{v_1 + q}{\omega_d}\right) e^{-\zeta\omega_n t} \sin \omega_d t + qt \tag{3.3}$$

$$\dot{h}(t) = -\left(\frac{v_1 + q}{\omega_d}\right) e^{-\zeta\omega_n t} \left(\omega_d \cos \omega_d t - \zeta\omega_n \sin \omega_d t\right) + q \tag{3.4}$$

Lift-off will occur at $t = t_{lo}$. There is no closed form solution for t_{lo} but it is close to half the period of oscillation, $t_{lo} \approx \frac{\pi}{\omega_d}$. The lift-off speed, which will be equivalent to the touch-down speed of the next hop v_2, can be found by substituting $t = t_{lo}$ into Eq. 3.4:

$$v_2 = \dot{h}(t_{lo}) \approx \dot{h}\left(\frac{\pi}{\omega_d}\right) \tag{3.5}$$

$$\Rightarrow \;\; v_2 \approx C_R(v_1 + q_1) + q_1 \tag{3.6}$$

where C_R is defined as:

$$C_R = \exp\left(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}\right) \tag{3.7}$$

The change in speed from touch-down to lift-off is $\Delta v_1 = v_2 - v_1$. By substituting $v_2 = v_1 + \Delta v_1$ into Eq. 3.6 and rearranging for q_1 it can be seen that:

$$q_n = K_L v_n + K_\Delta \Delta v_n \tag{3.8}$$

$$K_L = \frac{1 - C_R}{1 + C_R} \tag{3.9}$$

$$K_\Delta = \frac{1}{1 + C_R} \tag{3.10}$$

A control logic is provided by Eq. 3.8. It gives the actuator extension velocity q_n needed during the stance phase of the nth hop given a desired change in speed between touch-down and lift-off Δv_n. This can also be written in terms of the desired touch-down velocity v_{n+1} of the next, $(n+1)$th, hop:

$$q_n = K_1 v_n + K_2(v_{n+1} - v_n) \tag{3.11}$$

The gains in Eq. 3.11 can be initially set from Eq. 3.9 and Eq. 3.10 so $K_1 = K_L$ and $K_2 = K_\Delta$ and then tuned. The optimum values will differ from K_L and K_Δ due to the assumptions made to simplify the problem in the analysis here.

Motion during the flight phase is parabolic which means there are simple relationships between the touch-down speed v_n, flight time T_{fn} and hopping height h_n:

$$v_n = \frac{1}{2} g T_{fn} \tag{3.12}$$

$$v_n = \sqrt{2 g h_n} \tag{3.13}$$

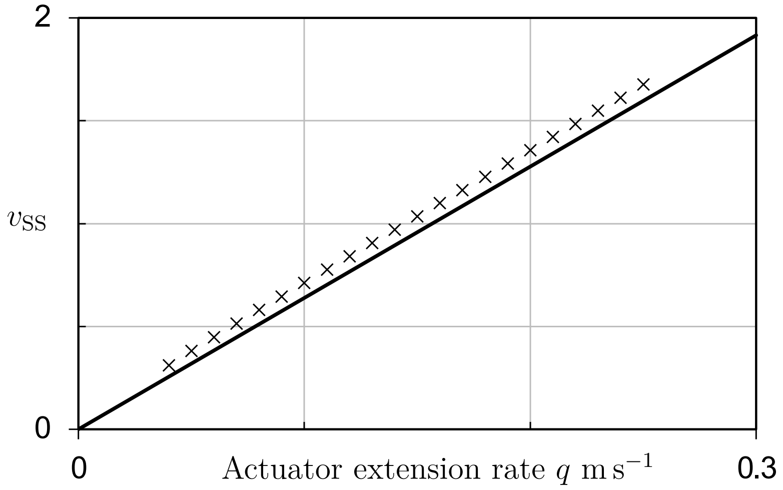


Figure 3-4: Steady-state lift-off velocity v_{SS} m s^{-1} for different actuator extension rates in Fig. 3-2 model. Line ($q = K_L v_{\mathrm{SS}}$; where $K_L = 0.157$) shows analytical results with simplifying assumptions. Crosses show actual values from simulation. Parameters are listed in Table 3.1.

Table 3.1: Parameters for simulations of Fig. 3-2.

	Parameter	Value	
$h(0)$	Initial height	0.1	m
$\dot{h}(0)$	Initial velocity	0	m s^{-1}
ω_n	Mass-spring natural frequency	62.8	rad s^{-1}
ζ	Damping ratio	0.1	

This means that the control logic of Eq. 3.11 can similarly be written in terms of T_{fn} and $\sqrt{h_n}$:

$$q_n = K_a\sqrt{h_n} + K_b\left(\sqrt{h_{n+1}} - \sqrt{h_n}\right) \tag{3.14}$$

$$q_n = K_\alpha T_{fn} + K_\beta(T_{f(n+1)} - T_{fn}) \tag{3.15}$$

It should be noted that q_n can take a negative value. This results in the leg retracting to remove energy from the system to reduce lift-off speed more than would be otherwise possible with damping alone.

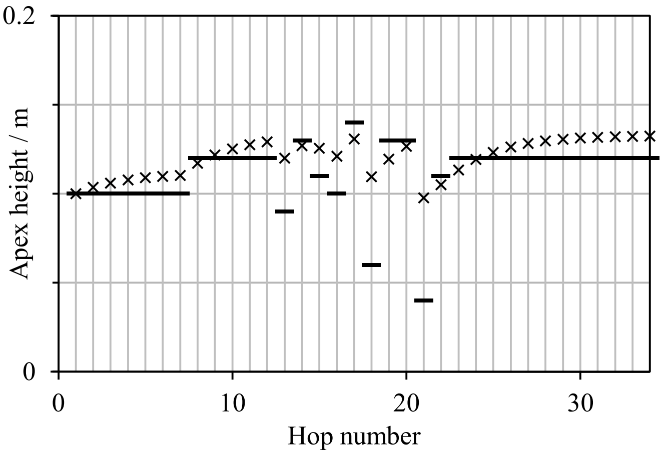


Figure 3-5: Simulation results for open loop height controller where $q = K_a\sqrt{h_d}$. Dashes: demand height, crosses: actual height.

3.1.1 Steady-state, open loop controller

The energy losses due to friction increase with higher impact speeds v_n. If the actuator action is kept constant so $q_n = q$ then the hopper will tend towards steady-state hopping, $v_n = v_{n+1} = v_{\mathrm{SS}}$, where the energy input by the actuator on each hop will be in equilibrium with the losses. The steady-steady relationship between input and output can be found from Eq. 3.8:

$$q = K_L v_{\mathrm{SS}} \tag{3.16}$$

As before, this may be written in terms of root apex height $\sqrt{h}$ or flight time T_f. It can also be seen that Eq. 3.16 contains no feedback of any variables. This type of height controller can be considered open-loop, requiring only the synchronisation of actuator outputs to touch-down events.

The analytical result from Eq. 3.16 (line) is plotted against simulation (crosses) in Fig. 3-4. Parameters for the simulation are listed in Table 3.1. The difference between simulation and analytical results can be attributed to the simplifying assumptions made while analysing stance:

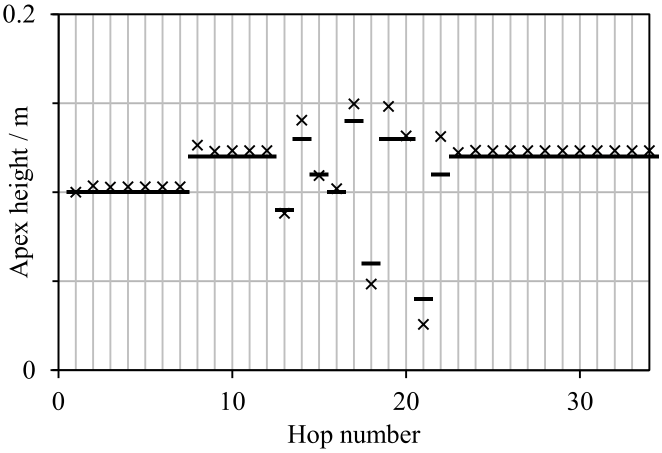


Figure 3-6: Height tracking performance for dynamic controller $q = K_1\sqrt{h_0} + K_2(\sqrt{h_d} - \sqrt{h_0})$. Dashes: demand height, crosses: actual height. $K_1 = 0.70$, $K_2 = 2.56$

- Gravity was neglected.
- Lift-off time t_{lo} was approximated.

Results for variable height demand with the open loop controller are plotted in Fig. 3-5. The apex height demand h_d is held constant at 0.1 m for the first 7 hops, increased to 0.12 m for hops 8 to 12, varied randomly between 0.04 m and 0.14 m for hops 13 to 22 and then kept constant at 0.12 m for the rest. During periods of constantly held demand the hopping height is converging to a constant value. There is a steady-state error because of simplifying assumptions made in deriving the controller gain $K_a = \sqrt{2g}K_L$. This controller is not able to track a rapidly changing height demand so would not be suitable for controlling foot placement.

3.1.2 Dynamic, adaptive, closed-loop controller

Simulation results for hopping height control using Eq. 3.14 are plotted in Fig. 3-6. The gains have been set analytically so:

- $K_a = \sqrt{2g}K_L$

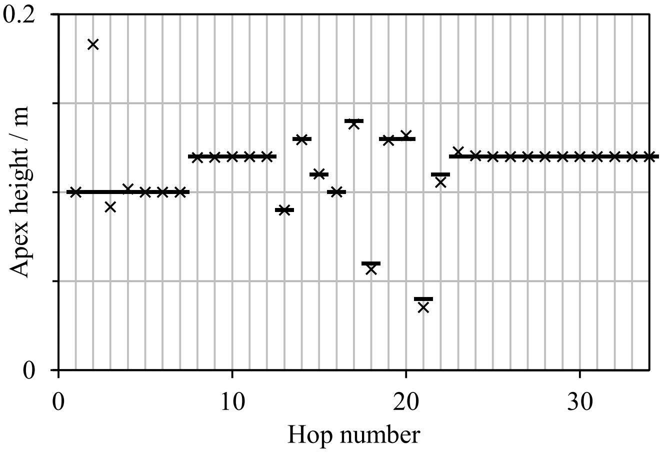


Figure 3-7: Height tracking performance for adaptive dynamic controller $q = K_1\sqrt{h_0}+K_2(\sqrt{h_d}-\sqrt{h_0})$. Dashes: demand height, crosses: actual height. Initially gain values are chosen to be $K_1 = 1.5$, $K_2 = 3.0$.

- $K_b = \sqrt{2g}K_\Delta$

This feedback controller offers better performance than the steady state controller of Fig. 3-5. Tracking is much faster and steady state errors are also reduced. The remaining error can be further reduced by tuning the control gains. The controller can be extended to make the gains self-tuning. If the touch-down speeds and actuator actions for previous hops are known then these can be used to improve controller gains.

Writing Eq. 3.11 for the previous two hops in matrix form gives:

$$\underbrace{\begin{bmatrix} v_{n-1} & \Delta v_{n-1} \\ v_{n-2} & \Delta v_{n-2} \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{pmatrix} K_1 \\ K_2 \end{pmatrix}}_{\mathbf{k}} = \underbrace{\begin{pmatrix} q_n \\ q_{n-1} \end{pmatrix}}_{\mathbf{q}} \tag{3.17}$$

where $\Delta v_n = v_{n+1} - v_n$. Solving for the gains $\mathbf{k}$ requires $|\mathbf{V}| \neq 0$. This can be computed:

$$|\mathbf{V}| = v_{n-1}v_{n-2} \underbrace{\left(\frac{\Delta v_{n-2}}{v_{n-2}} - \frac{\Delta v_{n-1}}{v_{n-1}} \right)}_{\rho} \tag{3.18}$$

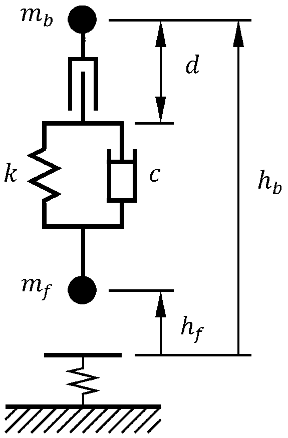


Figure 3-8: Model of hopping machine with foot mass and elastic ground.

The solution will be ill-conditioned if $|\mathbf{V}| \approx 0$. This can be avoided by checking a threshold condition, $|\rho| > 0.01$, is met. If below the threshold then the gains are left unchanged. In simulation it was also found that the controller can fail if set so $K_2 \approx 0$. A solution then is to not update the value of K_2.

The results of a simulation where the gains were self-tuned in this way are plotted in Fig. 3-7. Initial values of controller gains are selected to be poor. This results in hops 2 and 3 with large errors. Thereafter, the controller has enough information from previous hops to keep the gains correctly tuned. The self-tuning results in better performance than the analytically derived gains in Fig. 3-6.

3.1.3 Changing ground properties

Self-tuning control gains are useful when running over ground with changing properties. This can be demonstrated by simulations of the model shown in Fig. 3-8. This is similar to the previous model (Fig. 3-2) but includes a foot mass m_f in addition to the body mass m_b and a non-rigid ground. Here the ground which is a hard material is modelled as a non-linear spring-damper. The equations of

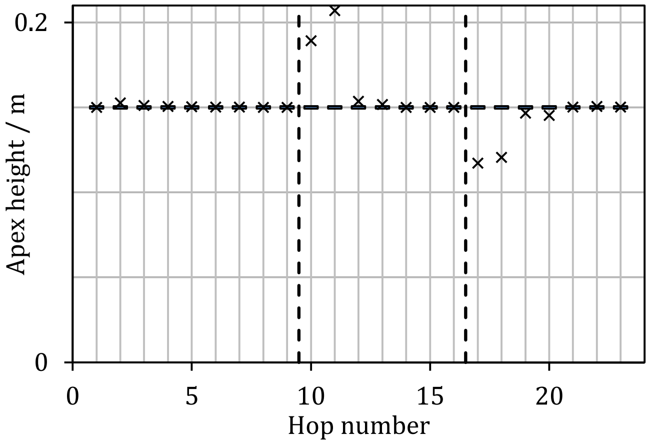


Figure 3-9: Results for Fig. 3-8 model with changing ground properties. Ground is soft after hop 9 and returns to hard after hop 16. Height demand is kept constant and adaptive controller is used.

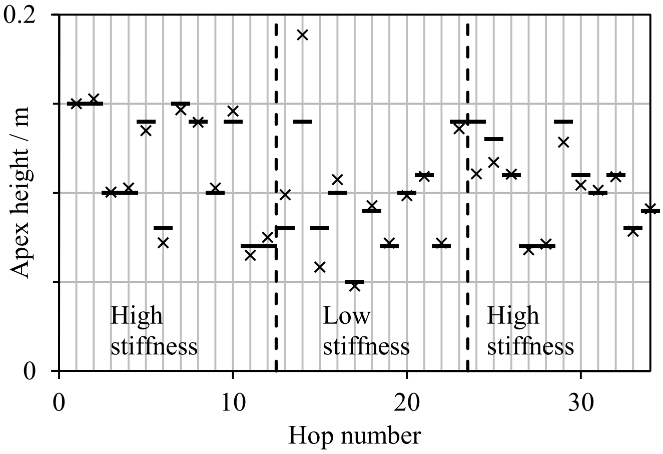


Figure 3-10: Fig. 3-8 model changing to soft ground after hop 12 and returning to hard ground after hop 23. Randomly varying height demand with adaptive controller.

Table 3.2: Fig. 3-8 model simulation parameters.

	Parameter	Value	
m_b	Body mass	10	kg
m_f	Foot mass	1	kg
k	Spring stiffness	8000	$\mathrm{N\,m^{-1}}$
c	Damping coefficient	30	$\mathrm{N\,s\,m^{-1}}$
$h_b(t=0)$	Initial body height	0.15	m
$h_f(t=0)$	Initial foot height	0.15	m
	High stiffness model:		
F_0	Reference spring force	10000	N
δ_0	Reference spring displacement	0.01	m
c_{gr}	Damping coefficient	10	$\mathrm{N\,s\,m^{-1}}$
	Low stiffness model:		
F_0	Reference spring force	100	N
δ_0	Reference spring displacement	0.01	m
c_{gr}	Damping coefficient	10	$\mathrm{N\,s\,m^{-1}}$

motion are:

$$F\left(h_f, \dot{h}_f\right) = F_0 \left(\frac{-h_f}{\delta_0}\right)^{\frac{3}{2}} + c_{\mathrm{gr}}(-\dot{h}_f) \tag{3.19}$$

$$X = h_b - d - h_f \tag{3.20}$$

$$T = c\dot{X} + kX \tag{3.21}$$

$$m_b\ddot{h}_b = -m_b g - T \tag{3.22}$$

$$m_f\ddot{h}_f = -m_f g + T + F\left(h_f, \dot{h}_f\right)[h_f < 0] \tag{3.23}$$

The parameters used in simulations are listed in Table 3.2.

Results for 2 simulations are plotted in Fig. 3-9 and Fig. 3-10. In the first

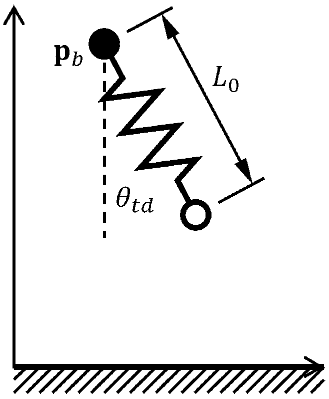


Figure 3-11: SLIP model of leg.

	Parameter	Value	
$\mathbf{p}_b(0)$	Initial position	$(0.00, 0.84)$	m
$\dot{\mathbf{p}}_b(0)$	Initial velocity	$(3.00, 0.00)$	$\mathrm{m\,s^{-1}}$
L_0	Unstrained spring length	1.00	m
ω_n	Spring natural frequency	27.8	$\mathrm{rad\,s^{-1}}$
θ_{td}	Touch-down angle	Varying	rad

Table 3.3: Initial conditions and parameters for SLIP hopper simulation.

simulation, the demand hopping height is kept constant. The ground properties are changed after hop 9 and 16. Within a couple of hops, the gains are tuned to the new ground. The second simulation is similar but presents a more challenging height demand. It is randomly varied between 0.05 m and 0.15 m. With a variable demand, the controller can still adapt to changing ground properties within a few hops.

3.2 Forward velocity control

3.2.1 The SLIP model

Running in 2D requires a method to maintain balance and control horizontal velocity. The SLIP model, shown in Fig. 3-11, can be used to analyse the problem of horizontal velocity control. This is a successful and frequently employed math-

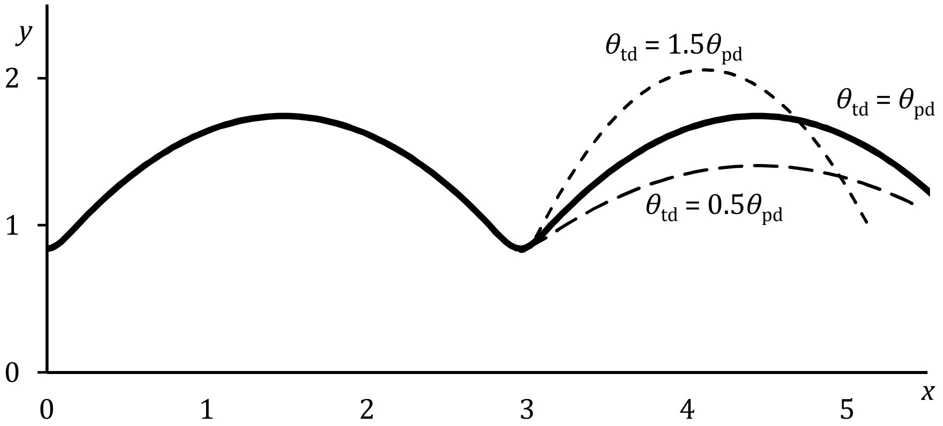


Figure 3-12: SLIP model passive dynamic motion in vertical y and horizontal x axes, SI units, for different touch-down angles showing effect of changing touch-down angle. Parameters in Table 3.3.

ematical model of a running animal's leg. It consists of a point mass representing the body and a massless elastic spring for the leg. During flight the leg can be repositioned without affecting the angular momentum or trajectory of the centre of mass of the system. If the angle of the leg is kept at θ_{td} relative to vertical until touch-down then the foot position during flight is:

$$\mathbf{p}_f = \mathbf{p}_b + L_0 \begin{pmatrix} \sin\theta_{td} \\ -\cos\theta_{td} \end{pmatrix} \tag{3.24}$$

and the equation of motion is:

$$\ddot{\mathbf{p}}_b = \mathbf{g} \tag{3.25}$$

Upon contact with the ground the foot is locked in position. The position of the body relative to the foot is then:

$$\mathbf{r}_{bf} = \mathbf{p}_b - \mathbf{p}_f \tag{3.26}$$

and the equation of motion while $|\mathbf{r}_{bf}| < L_0$ is:

$$\ddot{\mathbf{p}}_b = \mathbf{g} - \omega_n^2 \left(|\mathbf{r}_{bf}| - L_0\right) \hat{\mathbf{r}}_{bf} \tag{3.27}$$

where:

$$\hat{\mathbf{r}}_{bf} = \frac{\mathbf{r}_{bf}}{|\mathbf{r}_{bf}|} \tag{3.28}$$

The trajectory for the body, $\mathbf{p}_b$, that results from integrating the above equations is plotted in Fig. 3-12. The initial conditions and parameters are listed in Table 3.3. These are for the bottom of a hop. In this simulation lift-off occurs when the leg is at an angle of 0.297 rad. Due to the symmetric, frictionless nature of this model setting the touch-down angle for the next hop to match this angle will result in stable passive dynamic hopping. This is the 'passive dynamic' or neutral angle. If the touch-down angle on each hop is kept at the neutral angle, $\theta_{td} = \theta_{pd}$, the robot will maintain hopping at a fixed apex height and forward velocity.

So for a given running speed there will be a neutral angle θ_{pd} to maintain steady hopping. In Fig. 3-12 it can be seen that increasing the touch-down angle causes the hopping height to increase and therefore horizontal velocity to decrease. Similarly, decreasing the angle from θ_{pd} reduces height and increases forward velocity. This means horizontal velocity can be controlled by perturbing the touch-down angle about the neutral angle θ_{pd}. For the SLIP hopper there exist self-stabilising regimes where disturbances in forward velocity will self-correct but this will not be the case in general. The hopper will lose/gain velocity and gain/lose height until it topples.

Precisely computing θ_{pd} requires numerical integration. However an estimate can be made by assuming the body mass moves with a constant horizontal velocity u_n during stance for a time T_s as shown in Fig. 3-13. The distance moved horizontally during stance will then approximately be $u_n T_s$. The motion will

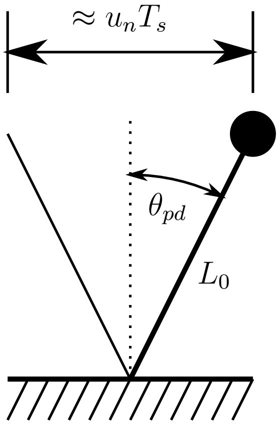


Figure 3-13: Estimation of the netural touch-down angle which results in no change to forward velocity.

be vertically symmetric for steady hopping. The angle swept by the leg during stance will therefore be $2\theta_{pd}$. The neutral angle is then:

$$\theta_{pd} \approx \sin^{-1}\left(\frac{u_n T_s}{2L_0}\right) \tag{3.29}$$

3.2.2 Control about the neutral angle

For a range of initial running speeds u_0, the effect on the running speed caused by a neutral angle deviation $\Delta\theta$ have been plotted in Fig. 3-14. It can be seen that the gradient $\frac{\Delta\theta}{\Delta u}$ is independent of running speed for small $\Delta\theta$. This means that the value of $\frac{\Delta\theta}{\Delta u}$ can be derived from the case of zero forward speed which is easier to analyse because $u_0 = 0$ and $\theta_{pd} = 0$. The case of touch-down with vertical touch-down velocity v_0 and forward velocity set to zero is illustrated in Fig. 3-15(a). Anti-clockwise rotation is positive by definition so the leg in Fig. 3-15(a) is at a deviation of $-\Delta\theta$ from the neutral angle θ_{pd} upon impact. The following assumptions are made for analysis:

- Small angles.
- Accelerations during the stance phase are high relative to g and the change in gravitational potential energy is also relatively small so $g \approx 0$.

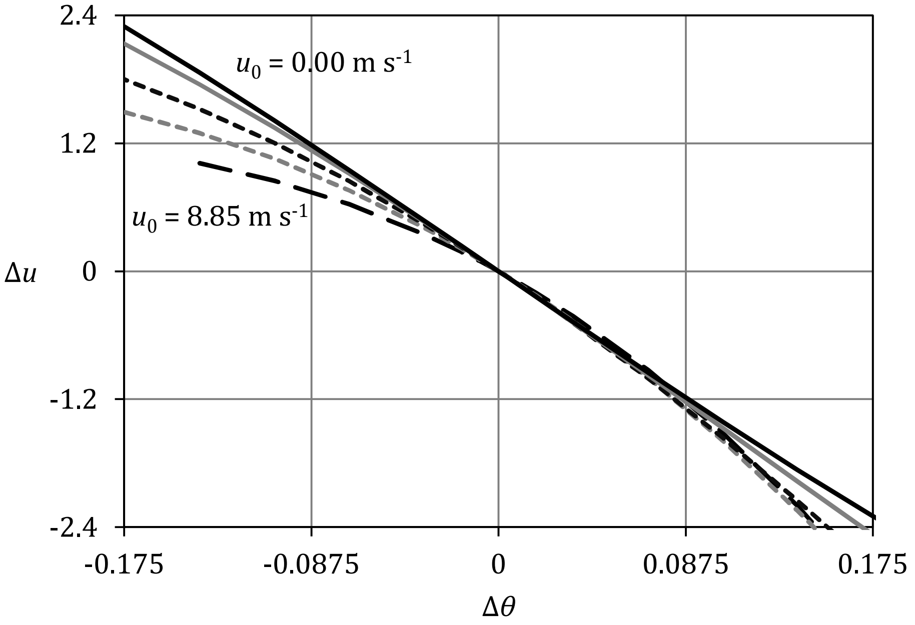


Figure 3-14: Effect of changing touch-down angle by $\Delta\theta$ rad about the neutral angle on forward velocity at different velocities: 0.00, 1.12, 3.36, 5.57 and $8.85\,\mathrm{m\,s^{-1}}$. The change in forward velocity is Δu $\mathrm{m\,s^{-1}}$.

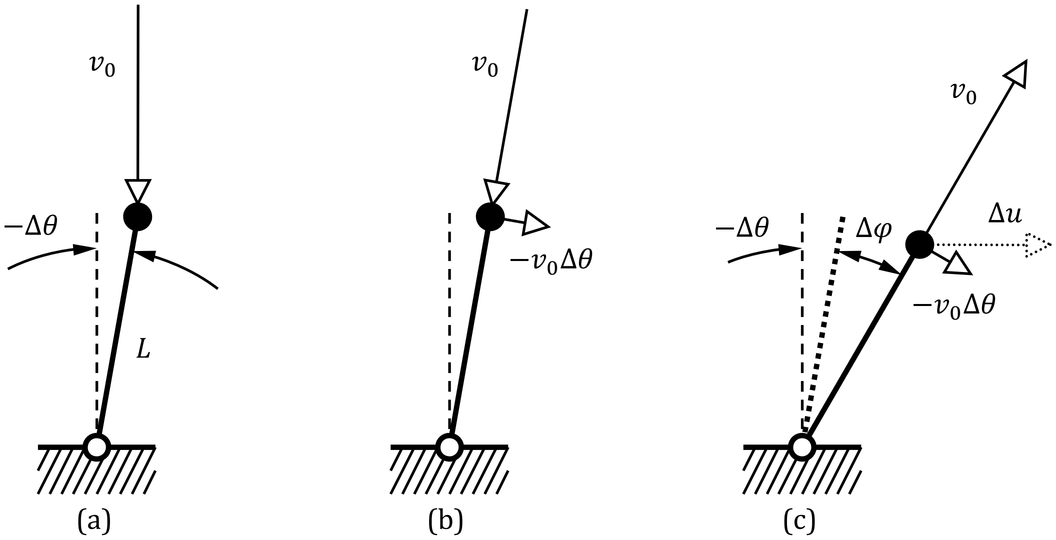


Figure 3-15: Touch-down with no forward speed and small clockwise leg deflection $\Delta\theta$. (a) At touch-down. (b) At touch-down with velocity resolved radially and tangentially. (c) At lift-off with the angle swept through stance labelled $\Delta\varphi$.

- Longitudinally, the leg is a stiff spring.

The velocity of the body at impact in Fig. 3-15(a) is $(0, -v_0)$ in Cartesian coordinates. Using small angle assumptions for $\Delta\theta$ the velocity at impact can be resolved into radial and tangential components: radially $v_0 \cos(-\Delta\theta) \approx v_0$; tangentially $v_0 \sin(-\Delta\theta) \approx -v_0\Delta\theta$. This is illustrated in Fig. 3-15(b).

In Fig. 3-15(c) the leg is shown at lift-off, having swept an additional angle $\Delta\varphi$ during the stance phase due to its initial angular momentum. The angle $\Delta\varphi$ can be approximated by assuming the angular velocity is approximately constant throughout stance, i.e. the radial velocity remains fixed at $-v_0\Delta\theta$ and leg compression is neglected. The angle turned by the leg is given by:

$$\Delta\varphi = \left(\frac{-v_n \Delta\theta}{L} \right) T_s \tag{3.30}$$

where T_s is the time between touch-down and lift-off.

Since the initial horizontal velocity of the body mass was zero, the change in horizontal velocity Δu_n is the horizontal component of the body velocity in Fig. 3-15(c):

$$\Delta u_n = (v_n) \sin\left(-\Delta\theta + \Delta\varphi\right) + (-v_n \Delta\theta) \cos\left(-\Delta\theta + \Delta\varphi\right) \tag{3.31}$$

Applying small angle assumptions:

$$\Delta u_n \approx -v_n \Delta\theta + v_n \left(-\Delta\theta + \Delta\varphi\right) \tag{3.32}$$

$$\Rightarrow \frac{\Delta\theta}{\Delta u_n} = -\left(2v_n + \frac{T_s}{L} v_n^2\right)^{-1} \tag{3.33}$$

Although derived here for the case where initial horizontal velocity $u_n = 0$, the result is applicable for non-zero forward velocities as found in Fig. 3-14.

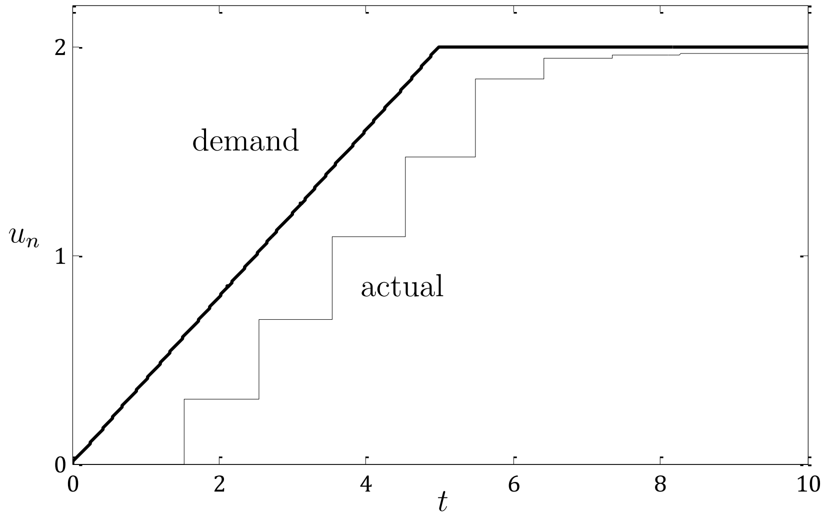


Figure 3-16: Ramp input for demand speed from 0 to $2\,\mathrm{m\,s^{-1}}$. SI units.

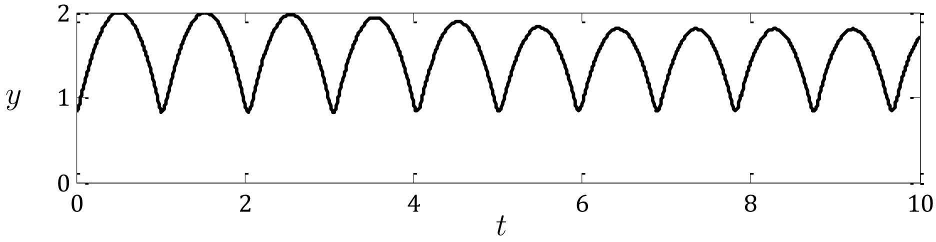


Figure 3-17: Height of body in ramp speed demand simulation. SI units.

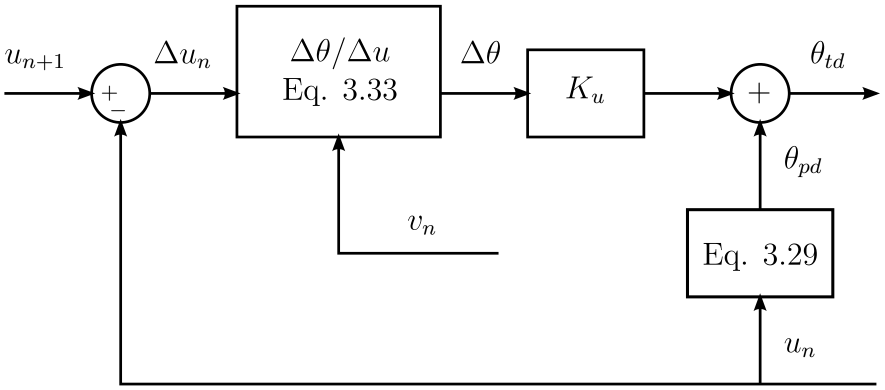


Figure 3-18: Forward velocity control loop. This is executed once per hop after the apex.

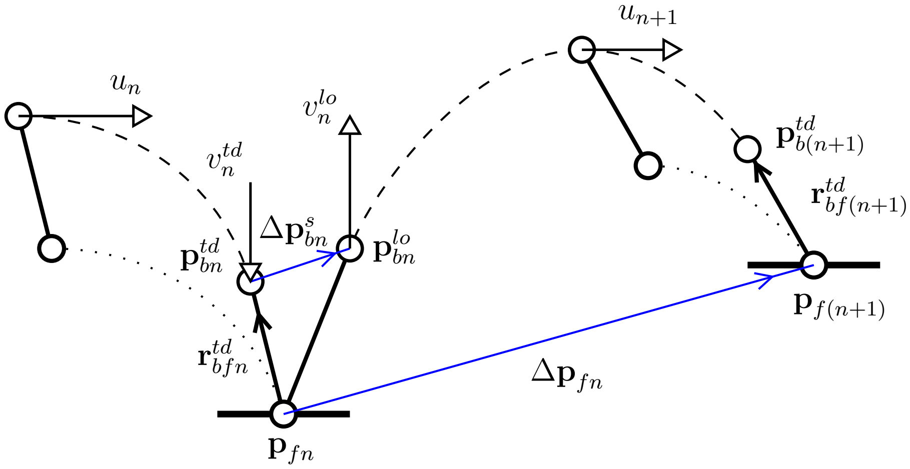


Figure 3-19: Parabolic flight of one-legged hopper as it jumps from one foot placement spot to the next. Going from left to right.

Figure 3-18 shows the closed loop forward speed control loop which utilises the equations developed in this section. This is a novel formulation of a speed controller because of the $\Delta\theta/\Delta u$ block which makes the gain K_u non-dimensional. This formulation may be useful for non-dimensional analysis of SLIP hoppers. Results for $K_u = 1$ are plotted in Fig. 3-16. The vertical position of the hopper is plotted in Fig. 3-17. Conservation of energy means height is lost as velocity is gained.

3.3 Foot placement by height control

To show how foot placement can be achieved work was carried out to simulate an extended SLIP hopper in the plane. The hopper was tasked with traversing isolated platforms placed at varying distances and heights. Results presented here show how this was achieved with a height controller. To begin with, a single hop is analysed.

Hop	State	Body: Position	Velocity	Foot position
$n-1$	Apex		$\begin{pmatrix} u_n \\ 0 \end{pmatrix}$	
n	Touch-down	$\mathbf{p}_{bn}^{td}$	$\begin{pmatrix} u_n \\ -v_n^{td} \end{pmatrix}$	$\mathbf{p}_{fn}$
n	Lift-off	$\mathbf{p}_{bn}^{lo}$	$\begin{pmatrix} u_{n+1} \\ v_n^{lo} \end{pmatrix}$	$\mathbf{p}_{fn}$
n	Apex		$\begin{pmatrix} u_{n+1} \\ 0 \end{pmatrix}$	
$n+1$	Touch-down	$\mathbf{p}_{b(n+1)}^{td}$	$\begin{pmatrix} u_{n+1} \\ -v_{n+1}^{td} \end{pmatrix}$	$\mathbf{p}_{f(n+1)}$

Table 3.4: States of hopper illustrated in Fig. 3-19 from left to right and the meaning of the symbols used.

3.3.1 Analysis of a planar hop

The trajectory of a one-legged hopper as it makes a leap from one platform (foot placement spot) to another is illustrated in Fig. 3-19. Table 3.4 explains some of the labels. The hopper begins at the apex of a hop before landing on one platform then launching itself to land at another. The hopper has to control its velocity at lift-off to achieve a ballistic trajectory which will land its foot on the next hop at a desired location $\mathbf{p}_{f(n+1)}$.

Landing at the desired location means the body must go from $\mathbf{p}_{bn}^{lo}$ to $\mathbf{p}_{b(n+1)}^{td}$. Flight will displace the body by:

$$\begin{aligned} \Delta\mathbf{p}_{bn}^{f} &= \mathbf{p}_{b(n+1)}^{td} - \mathbf{p}_{bn}^{lo} \\ &= -\Delta\mathbf{p}_{bn}^{s} - \mathbf{r}_{bfn}^{td} + \Delta\mathbf{p}_{fn} + \mathbf{r}_{bf(n+1)}^{td} \end{aligned} \tag{3.34}$$

At the beginning of the nth hop only $\mathbf{r}_{bfn}^{td}$ and $\Delta\mathbf{p}_{fn}$ in the above can be known. These are the body position relative to the foot and the relative position of the desired foot placement spot respectively. Simplifying assumptions can be made to calculate Eq. 3.34. If radical changes in horizontal speed or heading are not being made then it can be said that the leg orientation for hop $(n+1)$ will be similar to the nth:

$$\mathbf{r}^{td}_{bf(n+1)} \approx \mathbf{r}^{td}_{bfn} \tag{3.35}$$

The displacement through stance $\Delta\mathbf{p}^{s}_{bn}$ is a function of the system dynamics and control input. A precise computation would require iterative simulation since the control input also has to be determined. If the hopper is running at a steady velocity an approximate assumption can be made that it will maintain a steady horizontal velocity u_n for the duration of stance $T_s \approx \frac{\pi}{\omega}$:

$$\Delta\mathbf{p}^{s}_{bn} = \begin{pmatrix} \Delta x^{s}_{bn} \\ \Delta y^{s}_{bn} \end{pmatrix} \approx \begin{pmatrix} u_n T_s \\ 0 \end{pmatrix} \tag{3.36}$$

Substituting assumptions Eq. 3.35 and Eq. 3.36 into Eq. 3.34 and given $\Delta\mathbf{p}_{fn} = (\Delta x_{fn}, \Delta y_{fn})$ gives:

$$\Delta\mathbf{p}^{f}_{bn} \approx \begin{pmatrix} \Delta x_{fn} - \Delta x^{s}_{bn} \\ \Delta y_{fn} \end{pmatrix} \tag{3.37}$$

The flight time is given by:

$$T_{fn} = \frac{\Delta x_{fn} - \Delta x^{s}_{bn}}{u_{n+1}} \tag{3.38}$$

This can be computed if the assumption is made that forward speed for the next hop will be equal to that demanded $u_{n+1} \approx u_d$. Alternatively it could be assumed that the running speed is steady $u_{n+1} \approx u_n$. Then looking at the vertical component of the ballistic trajectory, the required lift-off velocity can be computed:

$$v^{lo}_{n} = \frac{\Delta y^{f}_{bn} + \frac{1}{2} g T^{2}_{fn}}{T_{fn}} \tag{3.39}$$

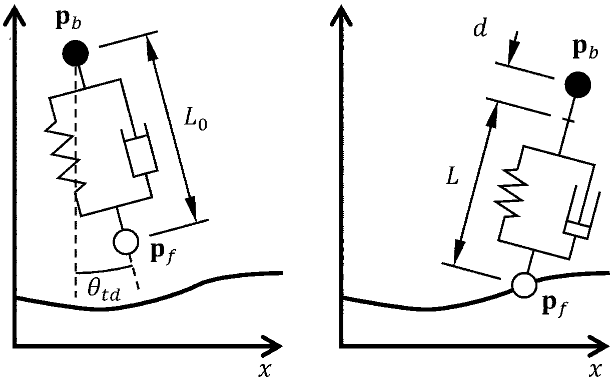


Figure 3-20: Model of hopping robot in flight (left) and during stance (right).

3.3.2 Planar extended SLIP model

The model shown in Fig. 3-20 will be used to develop and test a platform hopping control algorithm. The equations of motion outlined here were integrated using MATLAB Simulink. A variable step 'ode45' solver was used. There is a discontinuous change in forces on each hop at touch-down and lift-off so hit crossing blocks were implemented to ensure that the solver computes a time step immediately before and after touch-down/lift-off.

The model consists of a body and leg. The body is modelled as a simple point mass. The leg consists of a telescopic actuator and a spring-damper in series.

There are two control inputs to the system:

- The displacement of the telescopic leg actuator d.
- The angle of the leg upon touch-down θ_{td}.

There are two phases to the hopping model. During the flight phase, motion is ballistic:

$$\ddot{\mathbf{p}} = \mathbf{g} \tag{3.40}$$

At the apex of a hop, the leg touch-down angle θ_{td} is set. This is used to control the horizontal velocity. The position of the foot in flight is given by:

$$\mathbf{p}_f = \mathbf{p}_b + (d + L_0) \begin{pmatrix} \sin\theta_{td} \\ -\cos\theta_{td} \end{pmatrix} \tag{3.41}$$

Touch-down occurs when the foot $\mathbf{p}_f$ contacts the ground which is defined by $y_{\text{gr}} = y_{\text{gr}}(x)$. Upon touch-down the foot $\mathbf{p}_f$ is fixed at its position on contact throughout the stance phase. The relative position of the body from the foot is $\mathbf{r}_{bf} = \mathbf{p}_b - \mathbf{p}_f$. The equations of motion during the stance are:

$$\ddot{\mathbf{p}}_b = \mathbf{g} - f_L \hat{\mathbf{r}}_{bf} \tag{3.42}$$

$$f_L = \omega_n^2 (L - L_0) + 2\zeta\omega_n \dot{L} \tag{3.43}$$

$$L = |\mathbf{r}_{bf}| - d \tag{3.44}$$

$$\dot{L} = \dot{\mathbf{r}}_{bf} \cdot \hat{\mathbf{r}}_{bf} - \dot{d} \tag{3.45}$$

where $\hat{\mathbf{r}}_{bf} = \frac{\mathbf{r}_{bf}}{|\mathbf{r}_{bf}|}$ and f_L is the mass specific tension in the leg (tension per unit mass of body). Flight begins as soon as the leg goes into tension: $f_L \geq 0$.

3.3.3 Control method

To control the model above and execute foot placement the following are deployed:

- The closed-loop lift-off velocity control method from section 3.1.

- The forward velocity control method from section 3.2.

- The ballistic trajectory analysis here in section 3.3.

Combining these presents a test for many of the simplifying assumptions made in this chapter. All controller calculations will be made at the apex of hops.

Forward velocity control

The forward velocity controller from Fig. 3-18 can be shown to be:

$$\theta_{td} = \theta_{pd} + \frac{\Delta\theta}{\Delta u} K_u (u_{n+1} - u_n) \tag{3.46}$$

To achieve a stable steady-state running velocity an integral action was added and the proportional gain tuned down $K_u = 0.5$:

$$\theta_{td} = \theta_{pd} + \frac{\Delta\theta}{\Delta u} K_u (u_{n+1} - u_n) + C_u \tag{3.47}$$

where C_u is an offset provided by an integration action with the gain K_{uI}. It is updated on each hop:

$$C_{u(n+1)} = C_{u(n)} + K_{uI}(u_{n+1} - u_n) \tag{3.48}$$

Lift-off velocity control

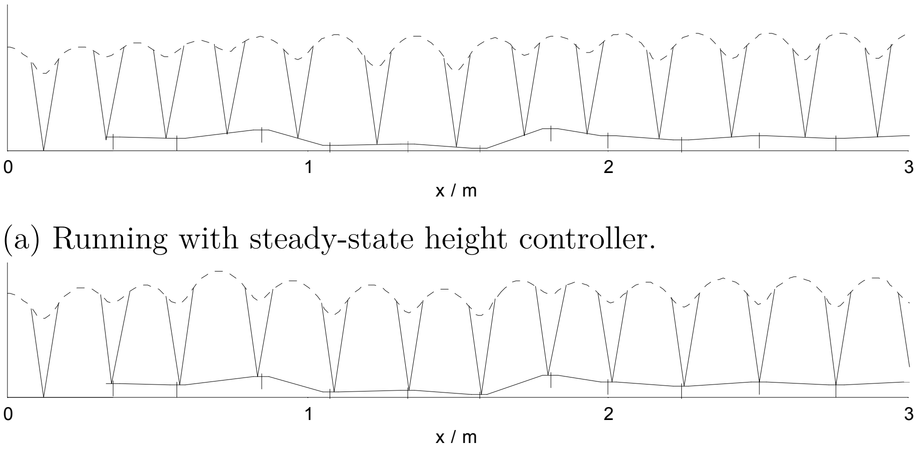
To achieve a ballistic trajectory to the next foot placement spot Eq. 3.39 is used to calculate the required lift-off velocity v_n^{lo}: This feeds into the closed-loop lift-off velocity controller, Eq. 3.8, to give the actuator extension velocity required during stance:

$$q_n = K_1 v_n^{td} + K_2 \left(v_n^{lo} - v_n^{td} \right) \tag{3.49}$$

The gains K_1 and K_2 are first allowed to self-tune. This is done by running on flat ground with random variations in the actuation q_n. Once tuned, these controller gains are kept fixed for all other simulations.

	Parameter	Value	
$\mathbf{p}_b(0)$	Initial body position	(0.00, 0.35)	m
$\dot{\mathbf{p}}_b(0)$	Initial horizontal velocity	(0.80, 0.00)	$\mathrm{m\,s^{-1}}$
L_0	Leg resting length	0.30	m
ω_n	Leg natural frequency	29	$\mathrm{rad\,s^{-1}}$
ζ	Leg damping ratio	0.1	

Table 3.5: Initial conditions and parameters for platform hopping simulations. These are based on rough estimates of hopping marsupial biomechanics.



(a) Running with steady-state height controller.

(b) Running with step by step control of height.

Figure 3-21: Running on rough terrain without, (a), and with, (b), attempt to control landing spots. Both axes are scaled equally.

3.3.4 Simulation results

Table 3.5 shows the parameters used in all simulations. The values selected are rough estimates derived from the analysis of video footage of hopping marsupials such as rock wallabies. All simulations begin at the apex of a hop giving the initial conditions:

$$\begin{aligned} \mathbf{p}_b\left(0\right) &= \begin{pmatrix} 0 \\ h_0 \end{pmatrix} \\ \dot{\mathbf{p}}_b\left(0\right) &= \begin{pmatrix} u_0 \\ 0 \end{pmatrix} \end{aligned} \tag{3.50}$$

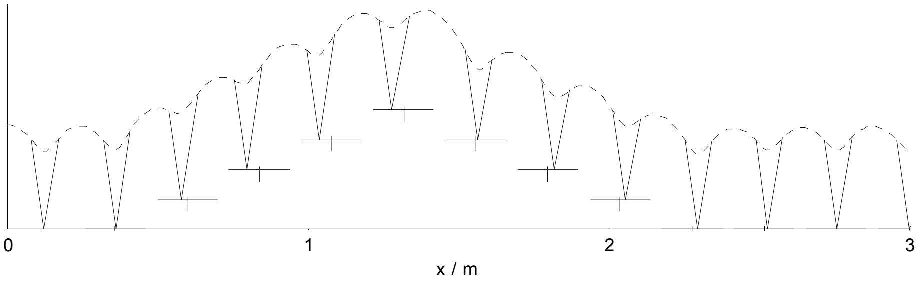


Figure 3-22: Robot trajectory when ascending and descending platforms. Both axes are scaled equally.

Terrain for the robot to run on is generated by placing 0.05 m flats evenly at distances of 0.24 m then displacing them randomly in the horizontal and vertical directions by up to ±0.04 m. The flats are then connected by straight lines. This forms a piecewise linear function which is y_{gr}. Results for the robot hopping over this terrain are plotted in Fig. 3-21. Vertical lines are used to show target foot placement spots.

Results without any foot placement controller are plotted in Fig. 3-21(a). Here the simple open-loop control method of extending the actuator with the same velocity, $q_n = 0.15\ \mathrm{m\,s^{-1}}$, on each hop is used. Results with the foot placement controller enabled are plotted in Fig. 3-21(b). Comparing the two, it can be seen that the height controller manages to keep the foot landing within ±0.025 m of the target.

Some terrain, steps for instance, may be impossible to traverse without controlling foot placement. In Fig. 3-22 equally spaced steps at distances of 0.24 m and heights of 0.1 m are attempted. It can be seen that the changes in height, which are relatively large when compared with the hopping height of approximately 0.05 m and leg length of 0.3 m increase the error in the foot landing position. The foot lands before the targeted spot when ascending the steps and ahead of the targeted spot when descending. This may be due to a weakening of the assumptions made in developing the hop controller. For example, it can be seen touch-down and lift-off do not occur at the same height. A more aggressive actuator action is required going up and down the stairs to appropriately affect the vertical energy of the system. Improved foot placement performance could be achieved by:

- Increasing the complexity of the assumptions made: for example by developing a better estimate for the body displacement between touch-down and lift-off.
- Changing the design of the robot: for example by increasing the stiffness of the leg.

3.4 Conclusion

To better model a hopping machine with a spring-loaded leg this chapter proposed the addition of two missing features to the SLIP model: losses due to friction; and energy input from actuation. This was done by adding a damper in parallel and actuator in series. From this a novel hopping height control method was developed. Modulation of the actuator extension rate during stance could be used to control the ballistic trajectory of hopping. By analysing the equations of motion of the model, it was found that a simple feed-forward plus proportional gain controller formed around the flight duration, lift-off velocity or root apex height was all that was needed to do this. Indeed, the ballistic trajectory could be controlled with sufficient accuracy to allow foot placement control while running in simulations.

The control algorithm developed was a simple linear function mapping output to input, that is lift-off velocity to actuator velocity. This meant the coefficients of that linear function, the controller gains, could be found by fitting to the results of previous hops. In this chapter gains were determined by fitting over the two most recent hops but the gains can also be found by fitting over several hops or by a continuous updating method which is demonstrated in chapter 6. The controller gains can be said to be self-tuning. This allows the running machine to adapt, to different ground properties for instance, to maintain accurate foot placement. Adaptation to different ground properties by self-tuning is validated in chapter 6.

The chapters following this one will experimentally validate the simple control scheme introduced here. A hydraulically actuated and spring-loaded robot leg with 2-links will be used to perform controlled hops while running on a treadmill. The non-linear dynamics and added complexities of the real machine will be handled. Then control of the ballistic trajectory of a running machine, sufficiently accurate for foot placement, will be shown. There is little doubt though that a lot more can be learned about animal and machine locomotion from continued modelling and simulation work. Even the familiar and apparently simple SLIP model is more complex than it seems. No solution is available for the neutral angle for instance. In regions of the parameter space it self-stabilises if perturbed which is a feature exploited by animal biomechanics. This region does not appear to have been mapped. Collectively, the modelling done here and the experimental work in later chapters lends credence to the more general hypothesis that walking and running machines with favourable passive dynamics for steady-state locomotion can be made to perform accurate and fast changes in direction with relatively simple control actions.

Chapter 4

Experimental system

In order to experimentally validate aspects of the control techniques developed in this dissertation, a test rig was constructed. The test rig consisted of a 2-link hydraulically actuated leg from the HyQ robot which was modified with the addition of a springy foot. A hydraulic supply, control valves, control computer and sensors were also added. The leg was constrained to hop vertically on a treadmill. This section provides a description of the leg and rig design as well as sizing calculations.

4.1 Overview of experimental rig

A schematic drawing of the experimental rig used in this thesis is shown in Fig. 4-1. A single leg from the HyQ robot has been approximately constrained to hop vertically on a treadmill using a pivoting beam. The leg consists of two links and a springy foot. The springy foot gives the leg a natural hopping motion. The leg is actuated by hydraulic actuators as shown. Sensors measure joint and beam angles θ_1, θ_2 and θ_b. Additionally an accelerometer is positioned above the 'hip' joint as shown. Key parameters for this experimental setup have been listed in Table 4.1. A photo of the rig can also be seen in Fig. 4-2.

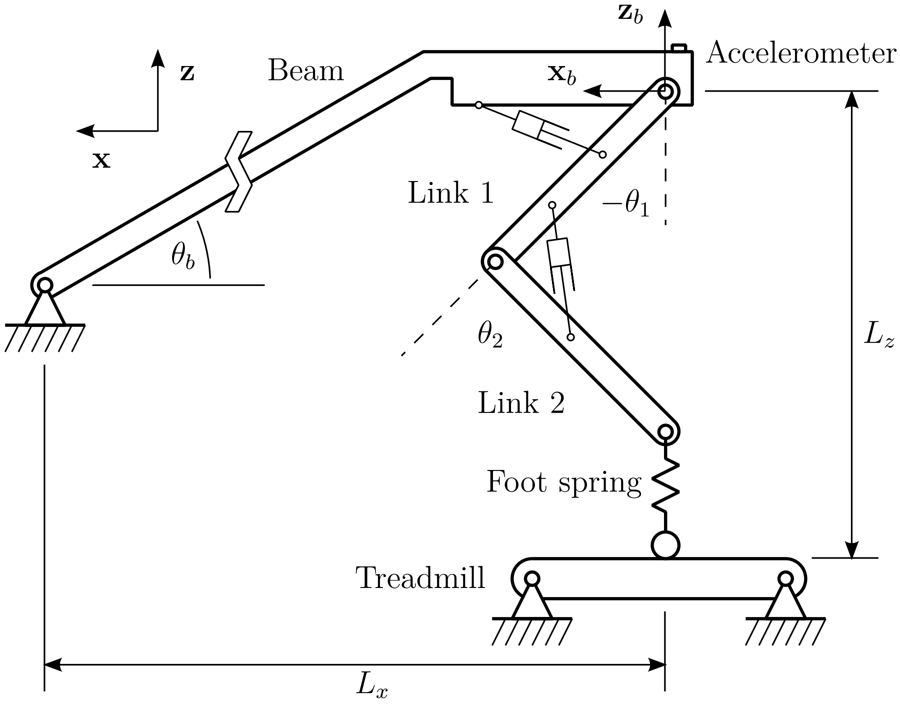


Figure 4-1: Schematic of experimental rig: emulates a two-link hydraulically actuated springy leg constrained to hop vertically. Degrees of freedom include the beam angle, hip angle, knee angle, spring displacement and treadmill motion.

Parameter	Value	
Link 1 length, Hip-knee	0.35	m
Link 1 mass	1.772	kg
Link 2 length, Knee-foot	0.33	m
Link 2 mass	0.808	kg
Aluminium box beam width	38.1	mm
Aluminium box beam thickness	3.2	mm
Total mass	18	kg
Approximate foot stiffness	10000	$\mathrm{N\,m^{-1}}$
Hip-beam pivot distance	2	m

Table 4.1: Experimental rig key parameters

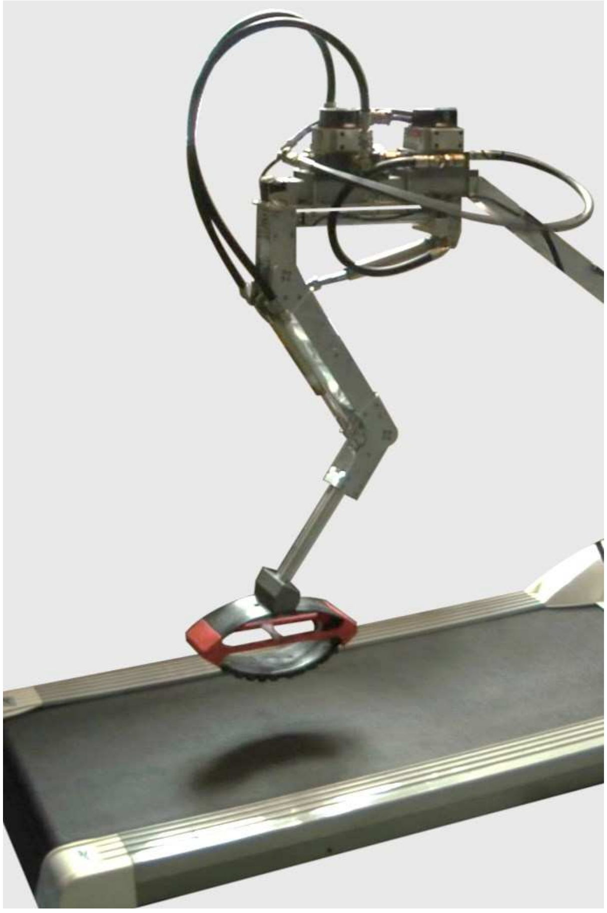


Figure 4-2: Photograph of experimental rig.

4.2 Mechanical design

4.2.1 Rig sizing

The robot leg used for experimentation was kindly donated by the Italian Institute of Technology. Detailed specifications can be found in [36]. The leg was originally designed for the HyQ robot [37]. For this work it was modified by adding a springy foot. This can be seen in the photo of the rig, Fig. 4-2.

The hopping rig was designed to approximately restrict the leg to hop vertically. The aim was to allow testing of the various hopping height controllers as well as experiments designed for example to explore strategies to cope with uncertain or shifting terrain. It can be seen in Fig. 4-1 that the leg is fixed at one end of a beam which is pivoted at the ground at the other end. For a long beam, the tilting of the leg will be negligible and the motion of the leg will be approximately restricted to the vertical.

The foot was added to introduce a passive hopping dynamic. In normal animal running, the stance duration tends to be approximately 30% of the total step-time T. This is a rough estimate based on observations of animal and human running on video. Although hops of different sizes can be executed by the hopping rig, this 30% duty cycle running can be used to determine a nominal hopping height for the running robot. Alternatively if the nominal hopping height is specified, a 30% duty cycle can be used to specify the robot's physical parameters.

The approximate stiffness of the foot spring and mass of the robot are $k = 10$ kN m^{-1} and $m = 18$ kg respectively. The approximate stance time then is half the period and is given by:

$$T_s \approx \pi\sqrt{\frac{m}{k}} = 0.13 \text{ s} \tag{4.1}$$

The flight time T_f is related to the hopping height by the relationship:

$$T_f = 2\sqrt{\frac{2h}{g}} \tag{4.2}$$

Given that $T = T_f + T_s$ and setting $T_s = 0.3T$, combining with Eq. 4.1 and Eq. 4.2 gives the following nominal values:

$$\begin{aligned} T &= 0.43 \text{ s} \\ T_f &= 0.30 \text{ s} \\ T_s &= 0.13 \text{ s} \\ h &= 0.11 \text{ m} \end{aligned} \tag{4.3}$$

A pivoting beam was used to achieve a constraint of vertical hopping approximately. The longer the constraining beam, the lower the change in the beam angle $\Delta\theta$ for a given change in the robot's height Δh. The nominal hopping height, Eq. 4.3, gives $\Delta h = 0.11$ m. Larger hops were expected during experiments so for the purpose of sizing the constraining beam a value of $\Delta h = 0.2$ m was used. As long as angle changes of the pivoting beam remain small $\Delta\theta \approx 0.1$ rad, the leg is approximately constrained to vertical hopping. The length of the beam L_x, shown on Fig. 4-1, follows from the hopping height and the corresponding change in angle of the beam. Rotating the beam by a small angle changes the height given by:

$$\Delta h \approx L_x \Delta\theta \Rightarrow L_x = 2 \text{ m} \tag{4.4}$$

The length of the leg links were 0.35 m and 0.33 m and joint 2 had a limit of 20°. This gave a maximum leg extension, from hip to ankle, of 0.66 m. For the purpose of rig sizing, a leg extension of 0.56 m, as shown in Fig. 4-2, was used as the nominal standing or 'home' position. The foot had a height of 0.14 m giving a standing height $L_y \approx 0.7$ m. These values, L_x and L_y, give the basic dimensions of the beam.

The leg could be made to hop on the spot or on a treadmill running with a set speed. Hopping on the treadmill requires that the foot reposition on each flight phase and sweep backwards while in contact with the ground. This presents an additional disturbance for the hopping height controller outlined in later chapters. Excluding the treadmill, the rig had the following degrees of freedom:

- Joint angles θ_1 and θ_2 on Fig. 4-1. These are hydraulically actuated revolute joints. Extension and retraction of the hydraulic cylinders corresponded to extension and flexion of the leg.
- Displacement of the spring foot.
- Beam pivot angle. This provided an approximately vertical degree-of-freedom for body.

4.2.2 Kinematics

In later chapters it will be necessary to compute what actuator velocities $\dot{\mathbf{d}}$ are required in order to achieve a specified foot velocity given in Cartesian coordinates $\dot{\mathbf{p}}$. For a given leg position a matrix can be computed to transform from one to the other:

$$\dot{\mathbf{d}} = \mathbf{F}(\boldsymbol{\theta}) \cdot \dot{\mathbf{p}} \tag{4.5}$$

The derivation of the matrix $\mathbf{F}$ will be outlined in this section.

As shown in Fig. 4-1, the leg consists of two links whose lengths are L_1 and L_2. The joints for θ_1 and θ_2 will be referred to as the ‘hip’ and ‘knee’ joints respectively. The distal point of link 2 is the foot position $\mathbf{p}$. In body fixed coordinates with origin at the hip:

$$\mathbf{p} = \begin{pmatrix} -L_1 \sin\theta_1 - L_2 \sin(\theta_1 + \theta_2) \\ -L_1 \cos\theta_1 - L_2 \cos(\theta_1 + \theta_2) \end{pmatrix} \tag{4.6}$$

Constant	Value	
a_1	0.3219	m
b_1	0.045	m
a_2	0.3218	m
b_2	0.045	m
ϵ_{11}	6.24	deg
ϵ_{21}	8.04	deg
ϵ_{22}	6.0	deg

Table 4.2: Constants for Eq. 4.9.

Defining $\boldsymbol{\theta} = (\theta_1, \theta_2)$ the Jacobian matrix is then defined as:

$$\mathbf{J}(\boldsymbol{\theta}) = \begin{bmatrix} \dfrac{\partial \mathbf{p}}{\partial \theta_1} & \dfrac{\partial \mathbf{p}}{\partial \theta_2} \end{bmatrix} \tag{4.7}$$

Joint velocities can thus be related to Cartesian foot position velocities:

$$\begin{aligned} \dot{\mathbf{p}} &= \mathbf{J}(\boldsymbol{\theta}) \cdot \dot{\boldsymbol{\theta}} \\ \Rightarrow \dot{\boldsymbol{\theta}} &= \mathbf{J}^{-1}(\boldsymbol{\theta}) \cdot \dot{\mathbf{p}} \end{aligned} \tag{4.8}$$

From the geometry of the system it is possible to derive equations for the actuator displacements in terms of the joint angles θ_1 and θ_2:

$$\mathbf{d} = \begin{pmatrix} d_1(\theta_1) \\ d_2(\theta_2) \end{pmatrix} = \begin{pmatrix} \sqrt{a_1^2 + b_1^2 - 2a_1 b_1 \cos\left(\pi/2 + \theta_1 + \epsilon_{11}\right)} \\ \sqrt{a_2^2 + b_2^2 - 2a_2 b_2 \cos\left(\pi - \theta_2 - \epsilon_{21} - \epsilon_{22}\right)} \end{pmatrix} \tag{4.9}$$

where constants are listed in Table 4.2. This leg geometry results in the leverage of the actuators changing with joint angle as plotted in Fig. 4-3 and Fig. 4-4.

Differentiating Eq. 4.9 gives the following matrix:

$$\mathbf{D}(\boldsymbol{\theta}) = \begin{bmatrix} \dfrac{\partial d_1}{\partial \theta_1} & 0 \\ 0 & \dfrac{\partial d_2}{\partial \theta_2} \end{bmatrix} \tag{4.10}$$

The cylinder displacement velocities can be computed, for a given leg position,

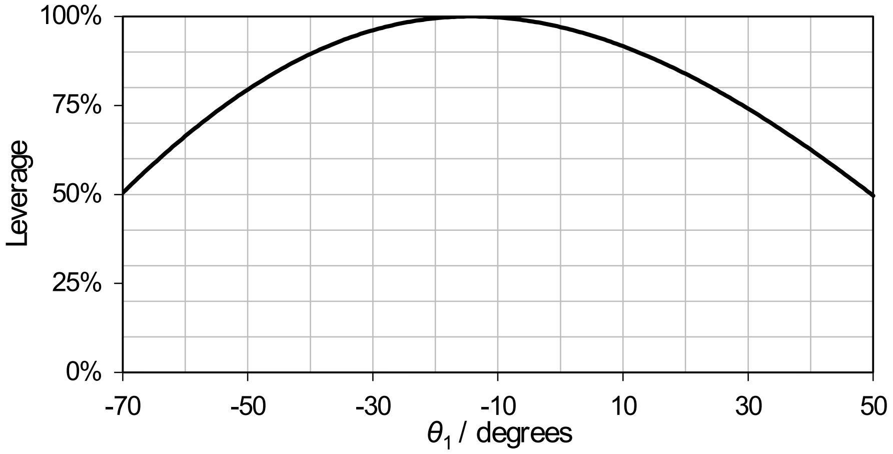


Figure 4-3: Variation of hip actuator lever arm with joint angle as proportion of maximum.

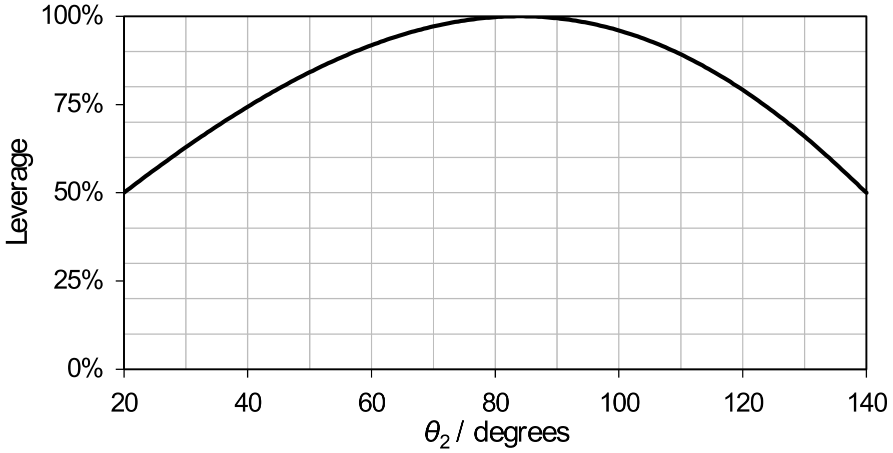


Figure 4-4: Variation of knee actuator lever arm with joint angle as proportion of maximum.

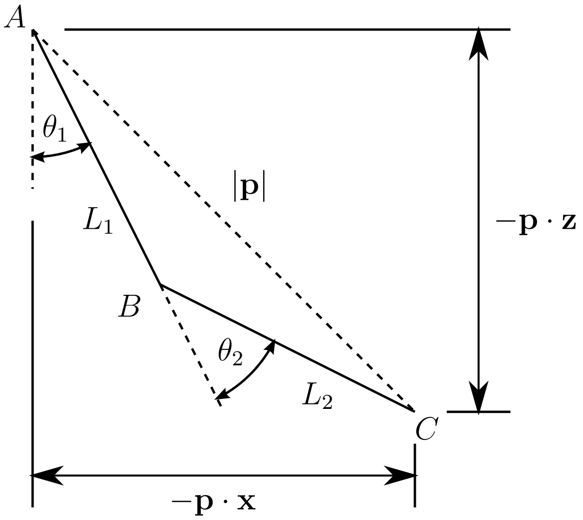


Figure 4-5: Geometry of two-links.

from given joint velocities:

$$\dot{\mathbf{d}} = \mathbf{D}(\boldsymbol{\theta}) \cdot \dot{\boldsymbol{\theta}} \tag{4.11}$$

Substituting Eq. 4.8 into Eq. 4.11 gives actuator velocities from foot velocity:

$$\begin{aligned} \dot{\mathbf{d}} &= \underbrace{\mathbf{D}(\boldsymbol{\theta})\mathbf{J}^{-1}(\boldsymbol{\theta})}_{\mathbf{F}(\boldsymbol{\theta})} \cdot \dot{\mathbf{p}} \\ \Rightarrow \dot{\mathbf{d}} &= \mathbf{F}(\boldsymbol{\theta}) \cdot \dot{\mathbf{p}} \end{aligned} \tag{4.12}$$

4.2.3 Inverse kinematics

In later chapters, it will be necessary to compute the joint positions $\boldsymbol{\theta} = (\theta_1, \theta_2)$ given Cartesian foot position $\mathbf{p}$. This can be done as follows.

From Fig. 4-5 using the cosine rule it can be seen that:

$$\angle BAC = \cos^{-1}\left(\frac{L_1^2 + |\mathbf{p}|^2 - L_2^2}{2L_1 |\mathbf{p}|}\right) \tag{4.13}$$

$$\angle ABC = \cos^{-1}\left(\frac{L_1^2 + L_2^2 - |\mathbf{p}|^2}{2L_1 L_2}\right) \tag{4.14}$$

Model	UniMeasure, Inc. VPA-15
Sensitivity	64.74 mV/V/Inch
Velocity	224.32 mV/100 Inch/min

Table 4.3: String potentiometer details.

and then it is clear that:

$$\begin{aligned} \tan\left(\theta_1 + \angle BAC\right) &= \frac{-\mathbf{p}\cdot\mathbf{x}}{-\mathbf{p}\cdot\mathbf{z}} \\ \Rightarrow \theta_1 &= \tan^{-1}\left(\frac{-\mathbf{p}\cdot\mathbf{x}}{-\mathbf{p}\cdot\mathbf{z}}\right) - \angle BAC \end{aligned} \tag{4.15}$$

and that:

$$\begin{aligned} \theta_2 + \angle ABC &= \pi \\ \Rightarrow \theta_2 &= \pi - \angle ABC \end{aligned} \tag{4.16}$$

4.3 Sensors

A number of sensors are installed on the experimental rig for the purpose of data-logging and feedback control. The following sensors are used:

- String potentiometer used to measure the beam pivot angle θ_b and $\dot{\theta}_b$. See Table 4.3.
- Rotary position encoders for measuring θ_1 and θ_2. See Table 4.4.
- Load cells to measure actuator forces F_1 and F_2. See Table 4.5.
- Accelerometer at position shown in Fig. 4-1 to measure C_x, C_y, C_z. See Table 4.6.

The string potentiometer is mounted in the same plane as the beam as shown in Fig. 4-6. The string is fixed to the beam at a distance L from the pivot and exits the string potentiometer at the point $\mathbf{p}_2$. The length of the string l_s is fed back to the controller as a signal voltage. From l_s and knowing the geometry of

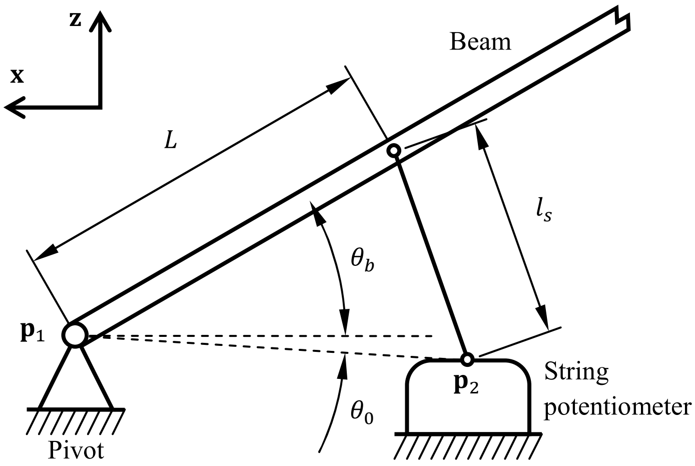


Figure 4-6: String potentiometer.

the system, it is possible to calculate the beam angle θ_b. First let $\mathbf{r}$ be the relative position of the pivot $\mathbf{p}_1$ from the string potentiometer exit point $\mathbf{p}_2$:

$$\begin{aligned} \mathbf{r} &= \mathbf{p}_1 - \mathbf{p}_2 \\ \Rightarrow r &= |\mathbf{r}| \end{aligned} \tag{4.17}$$

then it can be seen that:

$$\tan\theta_0 = \frac{\mathbf{r}\cdot\mathbf{z}}{\mathbf{r}\cdot\mathbf{x}} \tag{4.18}$$

and applying the cosine rule at the pivot $\mathbf{p}_1$ gives the beam angle θ_b as a function of the string length l_s:

$$\cos(\theta_b + \theta_0) = \frac{L^2 + r^2 - l_s^2}{2Lr} \tag{4.19}$$

Finally combining Eq. (4.18) and Eq. (4.19) gives θ_b as a function of the string length l_s:

$$\theta_b = \cos^{-1}\left(\frac{L^2 + r^2 - l_s^2}{2Lr}\right) - \tan^{-1}\left(\frac{\mathbf{r}\cdot\mathbf{z}}{\mathbf{r}\cdot\mathbf{x}}\right) \tag{4.20}$$

Model	AVAGO AEDA-3300-BE1
Counts/rev	80 000
Principle of operation	Optical disc
Output	A, B, I digital channels
Power supply	5 V
Sample rate	40 MHz

Table 4.4: Relative position encoders at joints 1 and 2

Model	burster Subminiature Load Cell Model 8417
Range	±5 kN
Accuracy	±0.5 % of full range
Principle of operation	Strain guages
Output	Analogue signal

Table 4.5: Load cells fitted to actuators 1 and 2

4.4 Actuation

In the experimental setup the leg was actuated by 2 identical hydraulic actuators, one for each joint. Their properties are listed in Table 4.7. The leg supplied by IIT came fitted with the 2 actuators and 2 joint position encoders. All other sensors were added as part of the work done here.

A circuit diagram of the hydraulics can be seen in Fig. 4-7. A 5 L/min electric motor driven pump was used to power the circuit. Due to the relief valve feeding back to tank, the pump acted as a constant 160 bar pressure source. Proportional control valves (see Table 4.8) were used to direct flow and control the actuators. Each valve was controlled by a current amplifier which generated a current in proportion to signal voltages V_1 and V_2 in the range $\pm 10V$ to position the valve spools. The control voltages were supplied by the controller.

Model	HiTechnic LEGO NXT Acceleration Sensor
Range	$\pm 2g$
Signal	3-axes analogue

Table 4.6: Accelerometer fitted at location shown in Fig. 4-1.

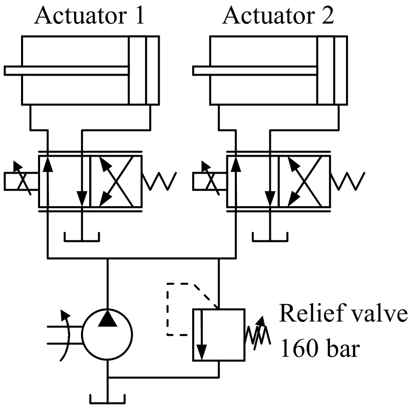


Figure 4-7: Hydraulic circuit diagram.

Model	Hoerbiger LB6-1610-0080-4M
Type	Single-ended
Stroke	80 mm
Maximum pressure	16 MPa
Piston diameter	16 mm
Rod diameter	10 mm
Maximum lever arm	45 mm

Table 4.7: Hydraulic actuators' properties

Model	Moog 20MA Series
Rating	2.5 lpm

Table 4.8: Hydraulic valve specifications

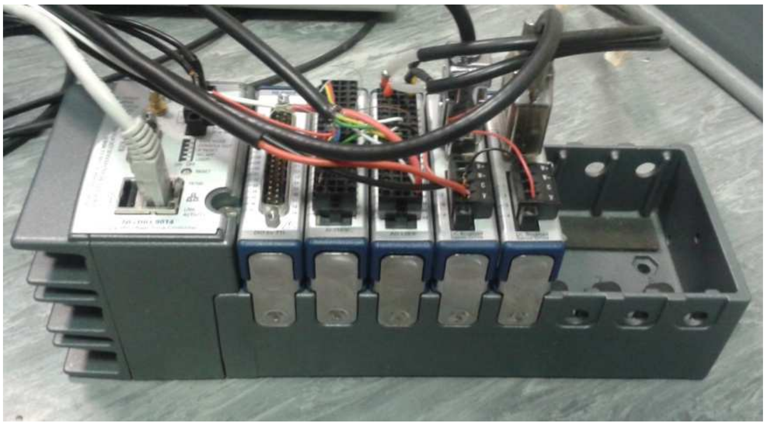


Figure 4-8: CompactRIO

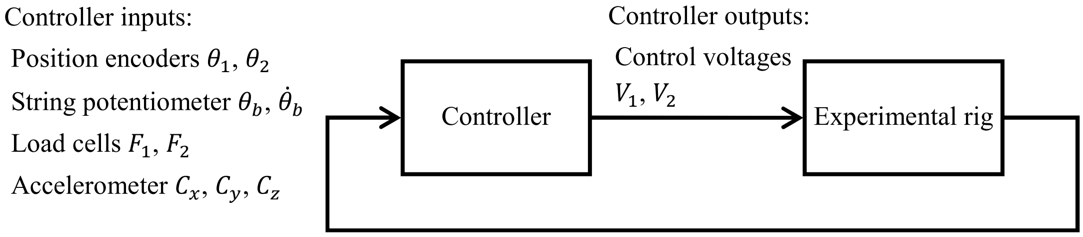

Controller inputs:
Position encoders θ_1, θ_2
String potentiometer θ_b, $\dot{\theta}_b$
Load cells F_1, F_2
Accelerometer C_x, C_y, C_z
Controller outputs:
Control voltages
V_1, V_2
Controller
Experimental rig

Figure 4-9: Controller inputs and outputs.

4.5 Controller

All digital and analogue signals were wired into a National Instruments cRIO-9014 'CompactRIO' programmable controller [38]. Additionally the CompactRIO also outputted the signal voltages used to control hydraulic valves. A summary of controller inputs and outputs can be seen in Fig. 4-9. The CompactRIO's inputs and outputs can be varied by plugging in different modules. The controller was programmed using 'LabVIEW' software. The CompactRIO and the various modules that were plugged into it can be seen in Fig. 4-8. Modules used included:

- Motor encoder modules (NI 9505) to read joint position encoders.
- Analogue input module (NI 9205) for string potentiometer, accelerometer and, force transducer.

Sensor signal	Cut-off frequency (Hz)	Analogue input
Accelerometer	500	AI1
String potentiometer position	1000	AI19
String potentiometer velocity	50	AI20
Load cell	200	AI6

Table 4.9: Analogue signals from the accelerometer, potentiometer and load cells are sampled at a rate of 20 kHz. The FPGA is programmed to perform Butterworth low-pass filtering on these signals at the cut-off frequencies shown in this table.

- Analogue output module (NI 9264) to send reference signal to valve amplifier. The valve amplifier generated current in proportion to the input reference voltage.

The CompactRIO included a FPGA (field-programmable gate array) and CPU. A FPGA is best suited for fast, parallel but relatively simple calculations. For this reason, in this project, the FPGA was used for sensor signal acquisition and filtering. More specifically the following tasks were programmed at the FPGA level:

- Count quadrature encoders from digital signals at 40 MHz sample rate, giving joint positions.
- Read string potentiometer analogue signal voltage and apply low-pass filters.
- Read accelerometer analogue signal voltages and apply low-pass filters.
- Read load cells signal and apply low-pass filters.

Low-pass filters were applied as listed in Table 4.9 in order to reduce high frequency electrical noise.

In addition to receiving sensor information, the controller had a module installed to output analogue signals. Signal voltages in the range ±10 V were passed to an amplifier circuit which produced a proportional current in order to position the valve spools.

All computations other than low-level signal acquisition were passed onto the CompactRIO's 400 MHz CPU. The CPU was programmed to run calculations in a 200 Hz timed-loop.

Chapter 5

Vibration control during leg swing

A hopping or running robot needs to reposition its foot forwards during the flight phase in a running cycle. This repositioning must be done with sufficient speed and accuracy so that upon touch down the foot has the desired position and velocity. For the experimental system used in this work (see chapter 4), a simple first attempt at controlling foot position was made by implementing PD position control on the two leg joints (hip and knee). It was found that the combined dynamics of the mechanical and hydraulic systems meant that tuning of PD gains was inadequate to achieve the required performance. High PD gains led to vibration and instability.

Sufficient performance was ultimately achieved using the control method described in this chapter. A video showing the results can be seen here:

`https://youtu.be/DJPPp5URkrU`

A P (proportional gain only) closed-loop controller with the addition of a feedforward comb filter was used. A feedforward comb filter adds a delayed version of a signal to itself leading to destructive (and constructive) interference. The technique of applying such a filter to the input reference signal of a position control system has been called ‘zero-vibration’ or ZV input shaping [39]. In open-loop form the ZV signal shaper cannot reject the vibrations induced from the initial foot lift-off conditions or other disturbances so a closed-loop for was developed.

First, the reasons why a PD controller is inadequate are discussed. Then, a simple simulation model of a single hydraulic cylinder acting on an inertial load is used to demonstrate the effect of closed loop signal shaping (CLSS) on reducing vibrations. To the author’s knowledge, the application of ZV signal shaping in a hydraulic system is novel [40]. Finally, the CLSS control loop is also implemented in the real experimental system and CLSS is validated experimentally.

5.1 Why PD is inadequate

The use of a simple P controller to control leg position in the air was observed to give rise to steady, limit cycle oscillations as the gain K_P was increased. Sufficiently responsive performance could not be achieved by tuning a simple P controller. It should be noted that disturbing the leg when in open-loop control led to vibrations of a much higher frequency that rapidly decayed away. This implies that the limit cycle oscillations that occured with the P controller were the consequence of the dynamics of the controller and the non-linear dynamics of the hydraulics. Oil compressibility is known to introduce resonant vibrations which can limit dynamic performance [41, 42]. This was found to be the case for the experimental system used here. The behaviour of the electro-hydraulic system meant satisfactory position control performance of the foot was also not achievable using a simple PD (proportional + derivative) controller in the air while hopping. The relatively poor performance of a PD controller when applied to a hydraulic actuator can be understood by making a comparison with, for example, a DC electric motor.

The relationship between the torque T produced by an electric motor and the position θ of an inertial, viscous and elastic load can be modelled as a second-order system. Such a system can be represented by the following equation:

$$\frac{\theta}{T} = G_1 = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{5.1}$$

When a PD (proportional K_P + derivative K_D) controller is applied to this system the closed loop poles are given by the following characteristic equation:

$$s^2 + (2\zeta\omega_n + K_D K\omega_n^2)s + (\omega_n^2 + K_P K\omega_n^2) = 0 \tag{5.2}$$

It can be seen that increasing the differential gain K_D, the gain on the velocity error in the case of an electric motor, has a similar effect on the dynamics of the system as increasing the damping ratio ζ. When tuning the controller gains K_P and K_D, oscillations induced by a high K_P can be damped by increasing K_D.

Now consider a hydraulic cylinder controlled by a proportional valve. Assuming valve dynamics are fast enough to be neglected, the flow into the cylinder will be proportional to the valve opening. Due to the compressibility of the hydraulic fluid and the inertia of the load oscillations will be induced by step changes in the input flow rate. The transfer function between the input flow Q and output cylinder velocity $\dot{x}$ can be approximately modelled as a second order system [41,42]:

$$\frac{sx}{Q} = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{5.3}$$

In terms of position, the open loop transfer function is third order:

$$\frac{x}{Q} = G_2 = \frac{K\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} \tag{5.4}$$

In this case, applying a PD control would result in the following characteristic equation:

$$s^3 + (2\zeta\omega_n)s^2 + (\omega_n^2 + K_D K \omega_n^2)s + (K_P K \omega_n^2) = 0 \tag{5.5}$$

Comparing the characteristic equation of the motor, Eq. 5.2, with the characteristic equation of the hydraulic cylinder, Eq. 5.5: it can be seen that unlike in the case of the motor, K_D does not appear alongside the damping term ζ i.e. as a coefficient of s^1. The effect of changing K_D is therefore not similar in the two cases. This explains why PD control does not perform well in position control of hydraulics. The gain K_D cannot be used to damp oscillations. Generally PD control of a third-order system might be problematic as there are only two gains available while three are needed to change each coefficient. The addition of acceleration feedback can be used to address this problem. Acceleration feedback however can be noisy to obtain either through numerical differentiation or by the use of accelerometers or load cells. An alternative approach to feeding back acceleration in a servo-hydraulic system is outlined in the following sections. Oscillations can be reduced through the application of a command shaping filter.

5.2 A solution: Closed-loop signal shaping

Command shaping is a well established and easy to implement method for eliminating vibrations from a control system [43]. The most common form this takes is open-loop input shaping. Input shaping is a control technique where the input reference signal to a system is modified in order to reduce oscillations in the response. Input shaping has been demonstrated to be useful in preventing oscillations when positioning cranes and flexible beams [44–46]. The application of shaping algorithms within a closed-loop has been labelled ‘closed-loop signal shaping’ (CLSS). A signal shaper within a closed-loop gives the advantage of removing oscillations due to external disturbances.

For second order systems, the ZV signal shaper can remove transient vibra-

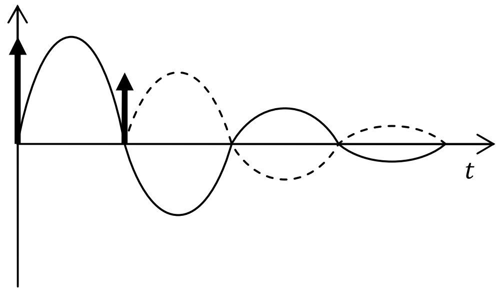


Figure 5-1: Second order system response to impulses.

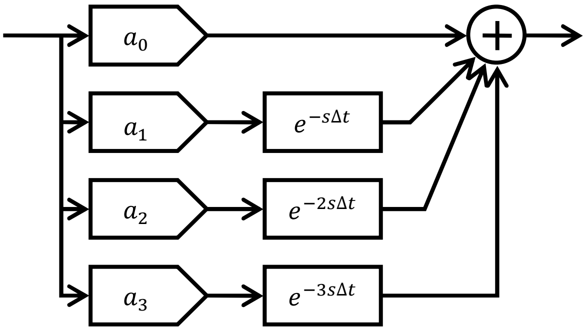


Figure 5-2: ZV signal shaper block diagram.

tions fully. As an example consider the response of a second-order system with transfer function $K\omega_n^2/(s^2 + 2\zeta\omega_n s + \omega_n^2)$ to an impulse as illustrated in Fig. 5-1. By applying a second impulse of an appropriate magnitude and at an appropriate time, it is possible to remove all oscillations except half of the first cycle. The ZV shaper transforms a single pulse into the two shown.

The ZV shaper functions by delaying a portion of the command signal by half the period of the system's oscillation frequency. The delayed part generates transient oscillations in anti-phase. The superposition of the responses to the immediate and delayed command signals results in the removal of transient oscillations. Figure 5-2 shows the block diagram for such signal shapers. For a simple ZV shaper only one delay is required so $a_2 = a_3 = 0$. Given the constraint that $\sum a_i = 1$, for a second-order system the values a_0, a_1, and Δt can then be determined through mathematical analysis [43] to be:

$$a_0 = \frac{1}{1+k} \tag{5.6}$$

$$a_1 = \frac{k}{1+k} \tag{5.7}$$

$$\Delta t = \frac{\pi}{\omega_d} \tag{5.8}$$

$$\omega_d = \omega_n \sqrt{1-\zeta^2} \tag{5.9}$$

$$k = e^{\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)} \tag{5.10}$$

where the damped natural frequency is $\omega_d = \omega_n\sqrt{1-\zeta^2}$.

Signal shapers with multiple delays such as ZVD (2 delays) and ZVDD (3 delays) work similarly but can reduce vibrations over a greater bandwidth of frequencies at the cost of response speed due to the extra delays. The ZVD shaper is obtained by using the following values:

$$a_0 = \frac{1}{1+2k+k^2} \tag{5.11}$$

$$a_1 = \frac{2k}{1+2k+k^2} \tag{5.12}$$

$$a_2 = \frac{k^2}{1+2k+k^2} \tag{5.13}$$

where k and Δt are computed as before using Eq. 5.10 and Eq. 5.8. For more information and details of derivation refer to [43].

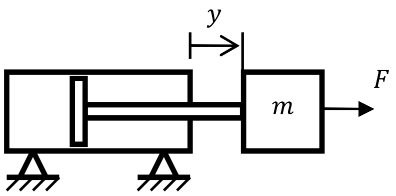


Figure 5-3: Hydraulic actuator model.

Table 5.1: Actuator model parameters.

	Parameter	Value	
A_1	Piston area	201.1	mm^2
A_2	Annulus area	122.5	mm^2
C	Leakage coefficient	4×10^{-14}	$\mathrm{kg}^{-1}\mathrm{m}^4\mathrm{s}$
c_f	Friction coefficient	400	$\mathrm{N\,s\,m}^{-1}$
K_1	Spool gain	0.1	$\mathrm{m\,V}^{-1}$
K_v	Valve coefficient	2.23×10^{-8}	$\mathrm{kg}^{-\frac{1}{2}}\mathrm{m}^{\frac{7}{2}}$
L	Actuator stroke	0.08	m
m	Load mass	50	kg
P_S	Supply pressure	16	MPa
P_R	Return pressure	0	MPa
W_1	Excess volume piston side	80	ml
W_2	Excess volume rod side	80	ml
β	Bulk modulus	1.56	GPa
ζ	Spool damping ratio	0.75	
ω_n	Spool natural frequency	50	Hz

5.3 CLSS control of hydraulic actuator in simulation

5.3.1 Simulation model

Figure 5-3 shows a single-acting hydraulic cylinder with an inertial load. This model will be used to compare the effect that adding a signal shaper to a P controller has on position control performance. Specifically, the control loop shown in Fig. 5-6 will be compared with the one in Fig. 5-7.

The hydraulic actuator is modelled using a set of non-linear differential equations. This is a model that captures the behaviour of hydraulic actuators suffi-

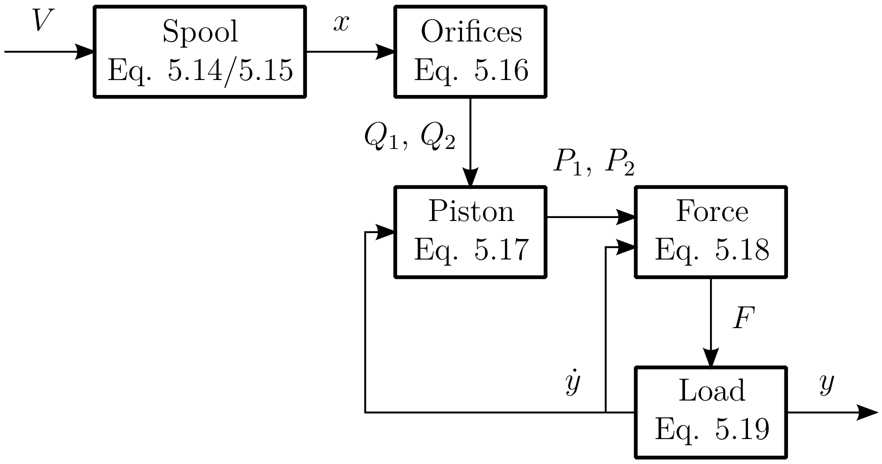


Figure 5-4: Block diagram of hydraulic cylinder model.

ciently well to demonstrate the effects of CLSS control.

This section details the hydraulic actuator model used to compare control techniques. The hydraulic system to be modelled (Fig. 5-3) consists of a set of equations which will be addressed in 5 parts:

- Valve spool
- Valve orifices
- Piston
- Extension force
- Load

The overall block diagram illustrating how these equations link together is shown in Fig. 5-4.

Valve spool

The valve spool moves to change the opening of a hydraulic servo valve x. A signal from the controller V commands the desired opening of the valve. The

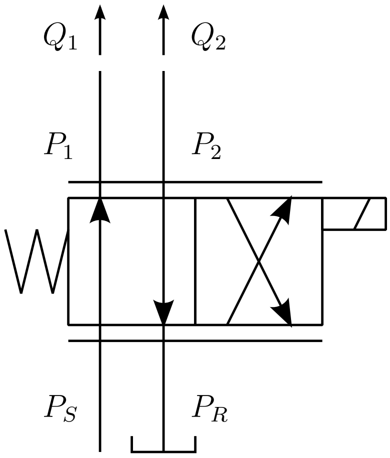


Figure 5-5: Symbolic representation of 4-way valve.

signal voltage V is sent to a current amplifier which then moves the valve spool via a hydraulic pilot stage. The dynamics of this sub-system are modelled using the following second order equation:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + x = K_1 V \omega_n^2 \tag{5.14}$$

The meanings of terms are given in the Table 5.1.

The valve is fully closed when $x = 0$ and fully open in one direction or the other when $x = \pm 1$. Before being passed out to the next set of modelling functions, the value of x is limited so that $|x| <= 1$:

$$\begin{aligned} x > +1 \;&\Rightarrow\; x \leftarrow +1 \\ x < -1 \;&\Rightarrow\; x \leftarrow -1 \end{aligned} \tag{5.15}$$

Valve orifices

The valve being modelled is a 4-way proportional hydraulic control valve. The valve is symbolically represented in Fig. 5-5. The valve opening x allows flow Q_1 and Q_2 through the two channels. Flow is induced by pressure differences across the valve and is modelled by the following functions:

$$\begin{aligned}
&\text{For } x \geq 0: \\
&\quad Q_1 = K_v x \,\mathrm{sign}(P_S - P_1)\sqrt{|P_S - P_1|} \\
&\quad Q_2 = -K_v x \,\mathrm{sign}(P_2 - P_R)\sqrt{|P_2 - P_R|} \\
&\text{For } x < 0: \\
&\quad Q_1 = K_v x \,\mathrm{sign}(P_1 - P_R)\sqrt{|P_1 - P_R|} \\
&\quad Q_2 = -K_v x \,\mathrm{sign}(P_S - P_2)\sqrt{|P_S - P_2|}
\end{aligned} \tag{5.16}$$

where P_S and P_R are the supply and reservoir pressures and P_1 and P_2 are the two load pressures. The valve coefficient is K_v.

Piston model

The pressures in the piston chambers, P_1 and P_2, that result from fluid flow, Q_1 and Q_2, into the piston are modelled by the following set of differential equations:

$$\begin{aligned}
Q_1 &= A_1\dot{y} + \left(\frac{A_1 y + W_1}{\beta}\right)\dot{P}_1 + C(P_1 - P_2) \\
Q_2 &= -A_1\dot{y} + \left(\frac{A_2(L - y) + W_2}{\beta}\right)\dot{P}_2 + C(P_2 - P_1)
\end{aligned} \tag{5.17}$$

where the variable y is the piston position. The meanings of constant parameters are listed in Table 5.1.

Extension force

The force on the piston rod is modelled as a function of the pressures on either side P_1 and P_2, and the piston velocity $\dot{y}$ due to viscous friction:

$$F = A_1 P_1 - A_2 P_2 - c_f \dot{y} \tag{5.18}$$

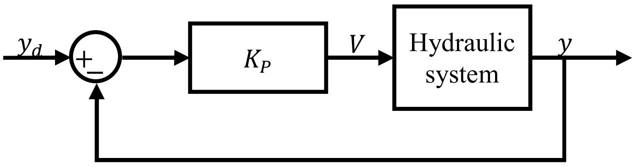


Figure 5-6: Proportional control.

The constant parameters are listed in Table 5.1.

Load

The load in the model is a simple mass with the piston acting horizontally. This is simply modelled by:

$$m\ddot{y} = F \tag{5.19}$$

where m is the combined piston and load mass.

5.3.2 Simulation results

A simulation of the hydraulic actuator model developed in this section was run (Fig. 5-3). A list of the constant simulation parameters is given in Table 5.1. The parameters have been selected to match the hydraulic equipment used in the experimental part of this project. The leakage coefficient C, viscous friction c_f and bulk modulus β are rough estimates. The list of state variables and their initial values are in Table 5.2. All simulations are run with the same model parameters and initial values. Only the control loop is varied.

Additionally, a zero-order hold with a sample rate of 200 Hz is placed before the input control voltage V and after the output position y (Fig. 5-6. This simulates a digital controller operating with a 200 Hz sample rate.

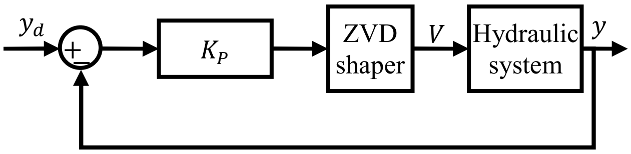


Figure 5-7: Proportional control with signal shaping filter in closed loop.

Table 5.2: Simulation initial conditions

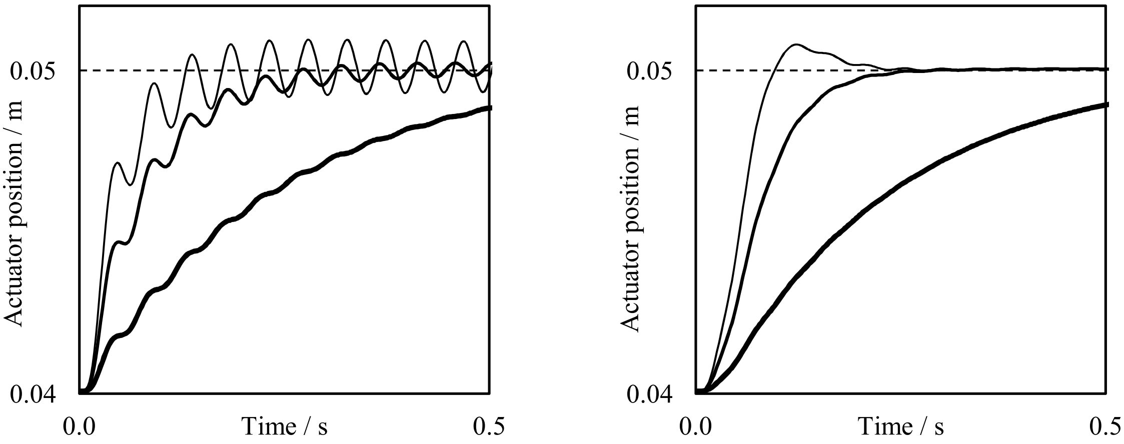
Variable	Initial value	
x	0	m
$\dot{x}$	0	$\mathrm{m\,s^{-1}}$
y	0.04	m
$\dot{y}$	0	$\mathrm{m\,s^{-1}}$
P_1	2.9	MPa
P_2	4.8	MPa

Method

The model constructed in section 5.3.1 can be represented by a block with: one input, the spool position control voltage V; and one output, the actuator position y. The simplest method of controlling the actuator position is using a proportional control loop, Fig. 5-6. The performance of this is to be compared against a modified control loop in which the input voltage to the valves is shaped as shown in Fig. 5-7. By placing the signal shaper inside the closed loop where it acts on the error signal as shown, it will act to remove oscillations caused by disturbances as well as by changes in demand.

Implementing the ZV shaper or its variants requires the natural frequency and damping ratio of the system as parameters. These can be obtained by giving a step input in voltage in an open loop to the hydraulic system and analysing the oscillations in the velocity response $\dot{y}$ to get a value for natural frequency and damping ratio. Before giving the open loop step input in voltage, closed loop proportional control is used to bring the actuator to a starting position at which it is held for a few seconds in order to allow transient motion to die down. This was done in simulation giving oscillations with an approximate natural frequency of 22 Hz and damping ratio of 0.1.

Results



(a) Proportional control.

(b) Proportional + closed loop ZVD shaping.

Figure 5-8: Simulation results: Response to step change in demand position of 0.01 m for three different proportional gains $K_P = \{100 \text{ (thick)}, 300, 500 \text{ (thin)}\}$ without (a) and with (b) signal shaping in a closed loop.

The ZV filter delays a part of the input by a delay amount Δt. The ZVD filter uses two delays, one of Δt and one of $2\Delta t$. The simulation results of implementing proportional control with and without a ZVD filter in the loop are shown in Fig. 5-8. It can be clearly seen that the signal shaping filter reduces oscillations allowing for higher gains. The reduced response speed due to use of the filter can be compensated by increasing proportional gain K_P.

5.4 Validating CLSS with experiment

Figure 5-9 shows the system used to experimentally validate the effectiveness of closed loop signal shaping control. The 2-link, 2-DoF articulated leg was lifted and fixed in place. This allowed the foot to move freely in the air without coming into contact with the ground.

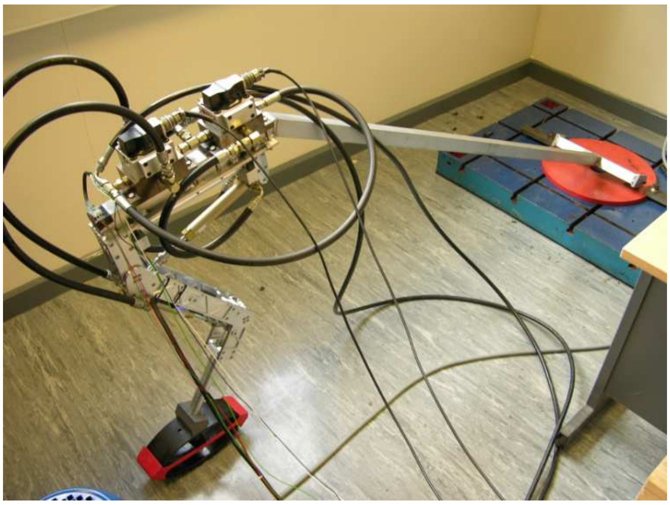


Figure 5-9: Experimental rig: 2-link hydraulic leg constrained to hop vertically by pivoted beam.

The cylinders used in experiment had the same dimensions (stroke and piston areas) as those in simulation. In experiments, the initial displacements of the upper and lower cylinders was $y_1 = 0.023$ m and $y_2 = 0.057$ m respectively. This starting position was reached and held using a low gain P controller with manually tuned offset voltages. This low gain P controller was switched off just before beginning an experiment at the same time as starting data-logging. The initial values of y_1 and y_2 (seen in Fig. 5-9) were selected so that the foot was vertically below the hip joint and the leg was not near full extension or full flexion. This was done so the foot would not be near the edge of its workspace.

Method

Before any of the ZV signal shapers can be used values for their parameters, frequency ω_n and damping ζ, must be determined. This was done in the following way. From the starting position, just as the initial P controller was switched off, the signal voltage to the upper actuator was stepped up by 4 V whilst the lower actuator signal voltage remained unchanged. The velocity of the upper actuator was recorded and analysed to obtain the natural frequency and damping ratio

Table 5.3: Velocity response results to 4 V step input from starting position. ZVD filters are tuned to remove oscillations with these parameters in experiments.

	Natural frequency ω_n	Damping ratio ζ
Actuator 1 (upper)	7.6 Hz	0.17
Actuator 2 (lower)	20 Hz	0.11

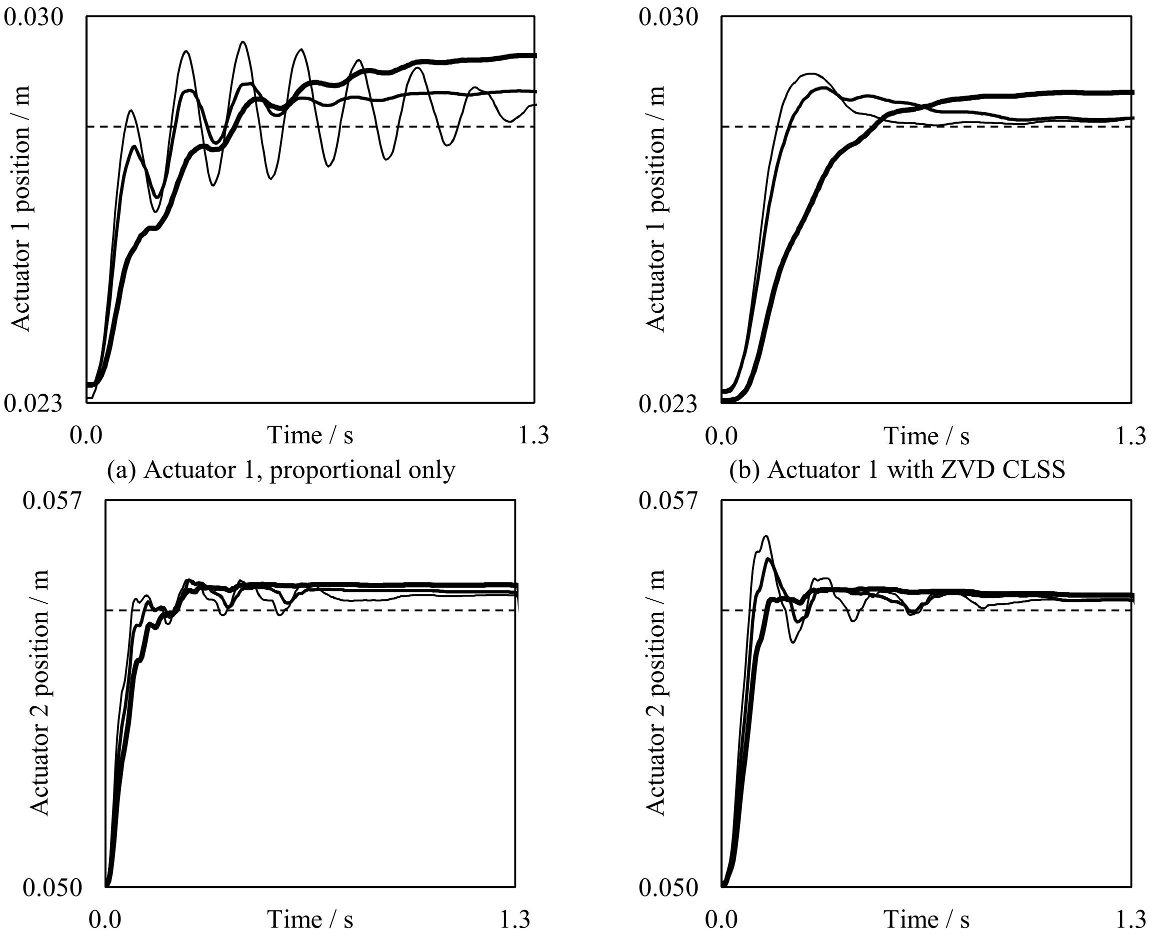
of the oscillations. The same process was then repeated for the lower actuator, with the upper actuator signal kept steady, to obtain a second set of natural frequency and damping ratio parameters. The resulting shaper parameters are listed in Table 5.3.

Results

Closed-loop P controllers without and with a ZVD shaper in the loop were implemented on both actuators (see Fig. 5-6 and Fig. 5-7). The position response of the actuators to step changes in demand position of 0.005 m, 6.25% of stroke is shown in Fig. 5-10. A step was given to the upper actuator demand position, y_1, while keeping the demand to the lower one, y_2, steady. This was then repeated with the roles reversed so y_2 is stepped and y_1 is fixed. This is done for each of three values for the proportional gain K_P. This gives the 12 time series results plotted in Fig. 5-10.

For the upper actuator with a simple P controller, (a), oscillations at approximately 6.3 Hz occurred for each of the three proportional gain values K_P tried. The oscillations became worse with higher K_P. It can be seen that adding the ZVD filter, tuned to 7.6 Hz, into the control loop, as in (b), removed the oscillations. These results are comparable to those from simulation shown in Fig. 5-8.

Inertial loading at the lower link actuator was lower than for the upper hip actuator. This lower inertia led to sufficiently responsive control even with just P control, (c). The lower actuator with P control (c) exhibited oscillations at two separate frequencies at least. Oscillations at 27 Hz and 6.3 Hz are present in (c).



(a) Actuator 1, proportional only

(b) Actuator 1 with ZVD CLSS

(c) Actuator 2, proportional only

(d) Actuator 2 with ZVD CLSS

Figure 5-10: Experimental results: Response of actuators to step change in demand with and without ZVD in closed loop.

The 6.3 Hz oscillations were likely due to the coupling in the dynamics of the two links of the leg. Results of applying the closed-loop ZVD to the lower actuator are shown in (d). In (c) and (d) it can be seen that the overshoot for higher gains is actually made worse by closed-loop ZVD and oscillations around 6.3 Hz are unaffected. A potential solution might be to implement a signal shaping filter tuned to remove multiple frequencies. This was not investigated because performance was adequate for the foot placement experiments carried out in this work with just the single frequency closed-loop ZVD.

5.5 Conclusion

In this chapter the problem of quickly repositioning the foot during the flight of a hop without inducing vibrations was addressed. The nature of hydraulic actuators meant that sufficient control performance could not be achieved using a conventional PD controller. Instead it was found by simulation and experiment that closed-loop signal shaping (CLSS) added to a proportional-only controller could remove vibrations giving sufficiently responsive position control performance. This application of CLSS to hydraulics is thought to be novel.

The relatively simple addition of a ZVD signal shaping filter into a proportional controller gives a number of benefits summarised below:

- Destabilizing oscillations can be removed allowing for increased servo-hydraulic performance.

- The filter is in a closed-loop controller so oscillations due to disturbances are also rejected.

- Implementing the filter only required the frequency and damping response of the system. These can be obtained using a simple experiment. Detailed modelling and analysis of the system is not required making the controller simple and easy to implement.

- The controller runs at a relatively low rate of 200 Hz and only requires position feedback. Alternative methods to improving hydraulic position control would require a higher sample rate and/or acceleration or force feedback.

- The CLSS filter can be switched off and on.

CLSS offers a relatively easy method for improving the performance of hydraulic actuators in some circumstances without requiring additional sensors. The ZV filters work by delaying a portion of the signal to create destructive interference. There is also however the potential for constructive interference. This may mean that the control loop as it stands at the moment is limited to applications where the innate oscillation frequencies do not vary too much. Further work might be carried out to:

- More rigorously investigate the stability and robustness of CLSS control of hydraulic systems experimentally.

- Mathematically analyse the stability of comb filters placed in closed loop controllers with a hydraulic actuator as plant represented by a third order transfer function.

- Compare the performance of CLSS against common techniques such as the introduction of notch filters and first order lags.

Chapter 6

Instantaneous control of hopping period

This chapter develops the controller shown in Fig. 6-1. The aim is to control the period of hops so that they match the demanded hop periods T_d. Ideally the controller should meet the demanded hop period on the next hop rather than converge to it over several hops. This level of control over the trajectory of a hop would allow hopping from one safe foot placement spot to the next. A video of an experiment can be seen here:

`https://youtu.be/pcIOQIeuZAs`

The controller outputs:

- Control signals to actuators. Here these are signal voltages (V_1, V_2) to control hydraulic valves.

The controller takes as inputs:

- Joint positions (proprioception). Here these are joint angles (θ_1, θ_2).
- A means to detect contact with the ground. Here the force measured by a

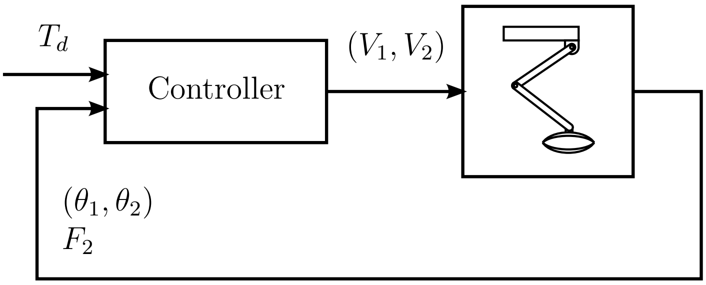


Figure 6-1: Controller developed in this chapter. Outputs are voltages (V_1, V_2) to signal hydraulic servovalves. Inputs are: joint positions (θ_1, θ_2); and load cell force at the knee F_2.

load cell at the knee joint F_2 will be used. The hop period is computed as the time between ground contact events.

Using the experimental rig described in chapter 4, hopping was first carried out on stationary ground. The controller was then extended to run on a treadmill over a range of different speeds.

The results presented validate a novel, simple, easily-implemented approach that can be used to achieve fine control over the flight and stance times of hops. Adaptation to different ground properties and treadmill speeds is also shown. This lends credence to the generalisability of the approach.

6.1 Low level control

The hopping period controller developed in this chapter calls upon and switches between two lower level controllers:

- Foot velocity controller. This is an open-loop controller used mainly while the foot is in contact with the ground to push downwards and sweep backward so there is no relative velocity between the foot and the ground when running at speed.

- Foot position controller. This is a closed-loop controller used during the flight phase of a hop to reposition the foot in preparation for touch-down.

6.1.1 Actuator asymmetry correction

The actuators used in our experimental setup are single ended hydraulic cylinders with rod and piston diameters of 0.010 m and 0.016 m respectively. The same flow rate results in different velocities and forces depending on whether the cylinder is extending or retracting due to asymmetric volumes on the two sides of the piston. To reduce this non-linearity control signals output to the servovalves are attenuated in one direction:

$$V_i \leftarrow \begin{cases} V_i & : V_i \leq 0 \\ 0.6V_i & : V_i > 0 \end{cases} \quad \text{for } i = \{1, 2\} \tag{6.1}$$

6.1.2 Open loop foot velocity control

The kinematics of the leg have, for a given position, a linear relationship between end effector velocity and actuator velocities. Additionally if transient behaviour is neglected then there is also a linear relationship between actuator velocities and control signal voltages. Overall a linear relationship between the end effector velocity and control signal voltages would be a reasonable hypothesis.

By carrying out a set of experiments and fitting lines to the results, the relationship between control signal and end effector velocity was determined. This was then used to achieve open-loop control of end effector velocity.

Rationale

To a first approximation hydraulic actuator velocity is proportional to the signal voltages output by the controller:

$$\dot{\mathbf{d}} \propto \mathbf{V} \tag{6.2}$$

This is the case because the signal voltages drive a current amplifier which outputs a proportional current. The current proportionally sets the position of the valve spool. The opening of the valve results in a proportional flow rate through the valve which finally results in a proportional cylinder velocity if the cylinder load is light and once transients have died out.

In section 4.2.2 the 2 × 2 matrix $\mathbf{F} = \mathbf{F}(\boldsymbol{\theta})$ was derived. This transforms reference Cartesian foot velocities $\dot{\mathbf{p}}$ to actuator velocities:

$$\dot{\mathbf{d}} = \mathbf{F} \cdot \dot{\mathbf{p}} \tag{6.3}$$

Combining Eq. 6.2 and Eq. 6.3 implies that, for a given foot position, foot velocity will be proportional to the valve signal voltages:

$$\mathbf{V} \propto \mathbf{F}(\boldsymbol{\theta}) \cdot \dot{\mathbf{p}} \tag{6.4}$$

Expanded:

$$\underbrace{\begin{pmatrix} V_1 \\ V_2 \end{pmatrix}}_{\mathbf{V}} = k \underbrace{\begin{pmatrix} F_{11} \\ F_{21} \end{pmatrix}}_{\mathbf{F}_x} \dot{p}_x + k \underbrace{\begin{pmatrix} F_{21} \\ F_{22} \end{pmatrix}}_{\mathbf{F}_z} \dot{p}_z \tag{6.5}$$

Method

In order to determine the relationship between $\mathbf{V}$ and $\dot{\mathbf{p}}$ a particular leg position is selected: $\mathbf{p}_h = (0, -0.56)$ m which corresponds to particular joint angles $\mathbf{p}_h \Rightarrow \boldsymbol{\theta}_h$

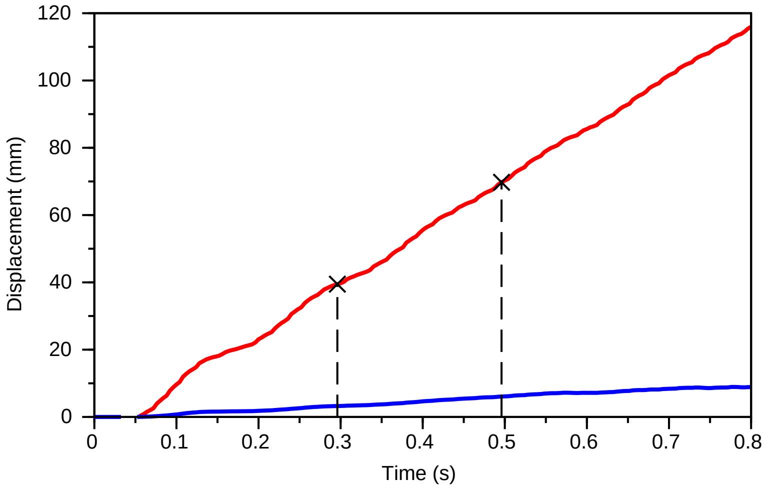


Figure 6-2: Foot displacement response for $(\alpha, \beta) = (4, 0)$. Red and blue lines are displacements from home position in x and z axes respectively. Between times $t = 0.296$ s and $t = 0.496$ s, the average value of $\dot{p}_x = 0.151$ ms^{-1}.

(see section 4.2.3 on inverse kinematics). Computing $\mathbf{F}$ at this position gives:

$$\mathbf{F}(\boldsymbol{\theta}_h) = \begin{bmatrix} \mathbf{F}_x(\boldsymbol{\theta}_h) & \mathbf{F}_z(\boldsymbol{\theta}_h) \end{bmatrix} = \begin{bmatrix} -0.0758 & 0.10798 \\ -0.0001 & 0.22585 \end{bmatrix} \tag{6.6}$$

From Eq. 6.5 it can be seen that:

$$\mathbf{V} = \alpha \mathbf{V}_x + \beta \mathbf{V}_z = k\dot{p}_x \mathbf{F}_x + k\dot{p}_z \mathbf{F}_z \tag{6.7}$$

if $\mathbf{V}_x$ and $\mathbf{V}_z$ are normalised versions of $\mathbf{F}_x$ and $\mathbf{F}_z$ so that $\mathbf{V}_x = (-1.00, 0.00)$ and $\mathbf{V}_z = (0.48, 1.00)$. A linear relationship is expected between α, β and $\dot{p}_x, \dot{p}_z$.

A set of experiments were carried out to map the relationship between α, β and $\dot{p}_x, \dot{p}_z$. With the leg held off the ground, the foot was brought to the home position $\boldsymbol{\theta}_h$ using PI joint control and all transients allowed to decay. The PI controller is then switched off and either:

- With $\beta = 0$, α was set to a non-zero value or

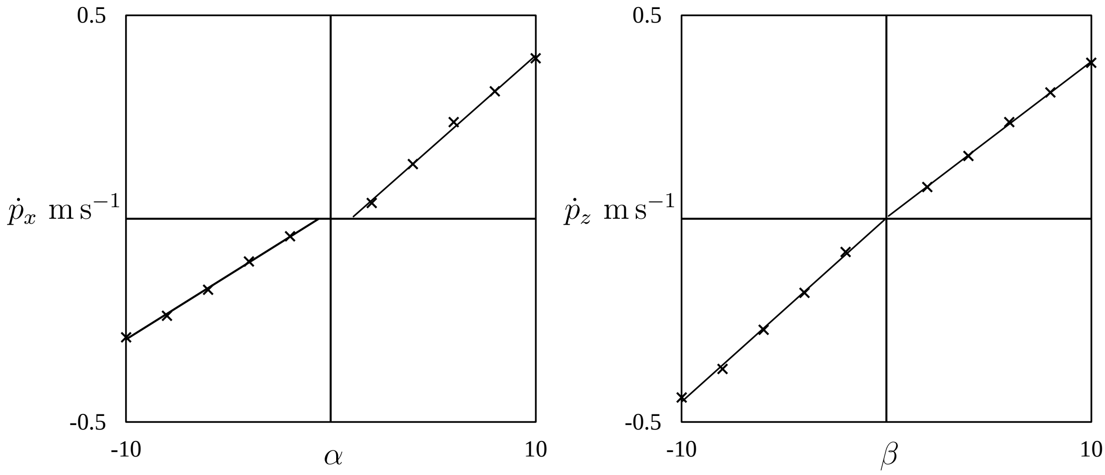


Figure 6-3: Steady-state foot velocity response with: variable α with $\beta = 0$ (left) and; variable β with $\alpha = 0$ (right). Lines have been fitted to these experimental results.

- With $\alpha = 0$, β was set to a non-zero value

This was carried out for each α from -10 to +10 in increments of 2 excluding $\alpha = 0$. The same range of values was applied to β. This gives 20 experiments in total with the end effector position response recorded in each case. A typical displacement response is shown in Fig. 6-2 for $(\alpha, \beta) = (4, 0)$. From the Eq. 6.7 model it should be expecteded that $\beta = 0 \Rightarrow \dot{p}_z = 0$. It can be seen that the velocity in the z axis is an order of magnitude lower than that in x so $\dot{p}_z \approx 0$ as expected.

In reality, the velocity response $\dot{p}_x$ to α is not instantaneous so has to be measured after allowing transients to decay. The velocity $\dot{p}_x$ must also however be measured close to the home position. Based on these two opposing requirements, it was decided the average velocity between the two times indicated in Fig. 6-2 would be measured as $\dot{p}_x$. In each experiment, the same time values are used.

Results

The results of all 20 experiments are plotted in Fig. 6-3. The results show linearity as expected but are discontinuous around $\dot{p}_x = 0$ and $\dot{p}_z = 0$. This may be due to a combination of factors including friction and the asymmetry of the single cylinders used. To accommodate these discontinuities piecewise lines were fitted giving the following:

$$\alpha(\dot{p}_x) = \begin{cases} 22.4\dot{p}_x + 0.98 & : \dot{p}_x > 0 \\ 31.7\dot{p}_x - 0.59 & : \dot{p}_x < 0 \end{cases} \tag{6.8}$$

$$\beta(\dot{p}_z) = \begin{cases} 26.0\dot{p}_z - 0.07 & : \dot{p}_z > 0 \\ 22.1\dot{p}_z - 0.03 & : \dot{p}_z < 0 \end{cases} \tag{6.9}$$

These lookup functions can now be used to achieve open-loop velocity control. Given a demand foot velocity $(\dot{p}_x, \dot{p}_z) = \dot{\mathbf{p}}$, Eq. 6.8 and Eq. 6.9 can be used to compute (α, β). The signal voltages are then computed using:

$$\begin{aligned} \mathbf{V} &= \alpha\mathbf{V}_x + \beta\mathbf{V}_z \\ \Rightarrow \mathbf{V} &= \mathbf{F}(\boldsymbol{\theta}_h) \begin{bmatrix} 0.0758 & \\ & 0.22585 \end{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \end{aligned} \tag{6.10}$$

If $\boldsymbol{\theta}$ is substituted into the above in place of $\boldsymbol{\theta}_h$ then the applicability of the above function is extended beyond one fixed home position giving:

$$\mathbf{V} = \mathbf{F}(\boldsymbol{\theta}) \begin{bmatrix} 0.0758 & \\ & 0.22585 \end{bmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{6.11}$$

Strictly speaking, using Eq. 6.11 may be called closed loop as feedback of joint positions is used in such a velocity controller. Finally, all experiments were carried out with the asymmetry correction given in Eq. 6.1 so these have to be applied to **V**.

6.1.3 Closed loop foot position control

During the flight phase of a hop the foot needs to be repositioned ready for landing. Different control methods may be used to achieve this. In this work, the control loop shown in Fig. 6-4 was used. It is a proportional controller implementing closed-loop ZVD signal shaping as developed in chapter 5. Here however the controller is formulated around the Cartesian end effector position rather than at the joint or actuator level. This requires computing the Jacobian matrix **F** (see section on kinematics 4.2.2).

In this controller, three parameters require tuning. These are:

- The ZVD shaper frequency
- The ZVD shaper damping ratio
- The proportional gain K

The ZVD parameters are tuned as in chapter 5. The gain K is manually tuned as follows. The demand position is cycled through 4 positions at 1 Hz drawing a

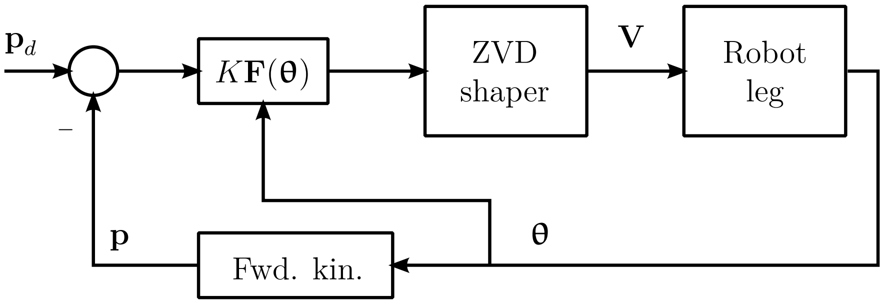


Figure 6-4: Block diagram for position controller. The signal **V** is corrected using Eq. 6.1 before being output to the robot. Forward kinematics are computed using Eq. 4.6.

0.05 m square around the home position $\mathbf{p}_h = (0, -0.56)$ m:

$$\mathbf{p}_d = \mathbf{p}_h + 0.025 \left\{ \begin{pmatrix} +1 \\ +1 \end{pmatrix}, \begin{pmatrix} +1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ +1 \end{pmatrix} \right\} \tag{6.12}$$

Live plots of $\mathbf{p} \cdot \mathbf{x}$ and $\mathbf{p} \cdot \mathbf{z}$ were looked at while tuning the value of K.

6.1.4 Switching between control modes

In the work which follows the two low-level control modes outlined above are used: position (section 6.1.3) and velocity (section 6.1.2). The two modes are used during different phases of the gait cycle. When switching between these two modes only the ZVD shaper presents an issue. This is because the shaper works by delaying a portion of the input which requires the use of an array to buffer the input. When entering position control mode, this array must be reinitialised. This is done by setting all values to the initial input to the ZVD shaper.

6.2 Model of stationary hopping

6.2.1 Theory

The trajectory of hopping can be split into two phases: flight and stance. These trajectories are illustrated in Fig. 6-5. Here the convention will be used that the nth hop begins at touch-down. The state of the robot at lift-off will be a function of the initial state at touch-down and the control action taken:

$$\mathrm{S}_n^{\mathrm{LO}} = f\left(\mathrm{S}_n^{\mathrm{TD}}, a_n^{\mathrm{S}}\right) \tag{6.13}$$

where a_n^{S} is the action taken during the stance phase of the nth hop from a set of possible stance control actions, $a_n^{\mathrm{S}} \in \mathrm{A}^{\mathrm{S}}$.

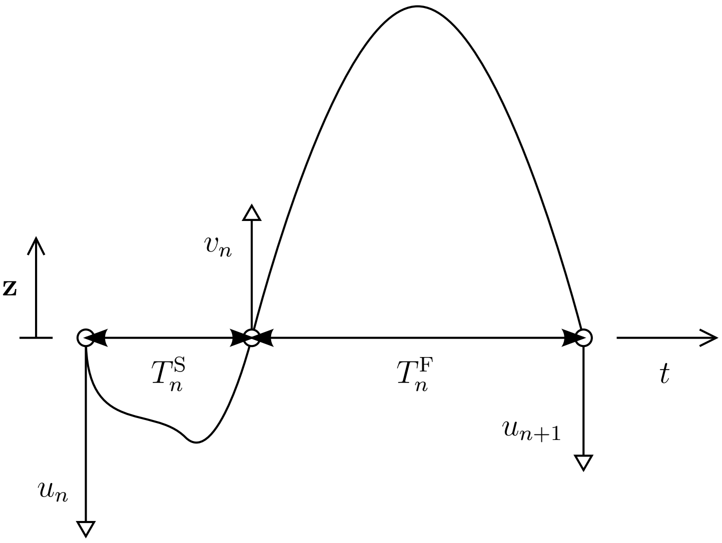


Figure 6-5: Trajectory of the nth hop. Touch-down occurs with downward velocity u_n and lift-off with upward velocity v_n. The stance and flight phases last for time T_n^{S} and T_n^{F} respectively. During flight, the trajectory is parabolic.

Assuming that only the vertical velocity varies significantly for different actions implies:

$$v_n = f_1\left(u_n, a_n^{\mathrm{S}}\right) \tag{6.14}$$

Assuming hopping on terrain of a fixed height, it is clear from Fig. 6-5 that $u_{n+1} = v_n$. Therefore:

$$u_{n+1} = f_1\left(u_n, a_n^{\mathrm{S}}\right) \tag{6.15}$$

The equations of motion under constant acceleration can be used to get the flight period giving:

$$T_n^{\mathrm{F}} = \frac{2u_{n+1}}{g} \tag{6.16}$$

$$\begin{aligned} \Rightarrow \quad u_{n+1} &= \frac{g}{2}T_n^{\mathrm{F}} \\ \Rightarrow \quad u_n &= \frac{g}{2}T_{n-1}^{\mathrm{F}} \end{aligned} \tag{6.17}$$

Substituting Eq. 6.17 into Eq. 6.15 and using $T_n = T_n^{\mathrm{S}} + T_n^{\mathrm{F}}$ gives:

$$\frac{g}{2}\left(T_n - T_n^{\mathrm{S}}\right) = f_1\left(\frac{g}{2}\left(T_{n-1} - T_{n-1}^{\mathrm{S}}\right), a_n^{\mathrm{S}}\right) \tag{6.18}$$

Assuming now that the stance time is constant $T_n^{\mathrm{S}} = T^{\mathrm{S}}$, Eq. 6.18 can be rearranged to get a function for T_n:

$$T_n = f_2\left(T_{n-1}, a_n^{\mathrm{S}}\right) \tag{6.19}$$

What Eq. 6.19 states is that the period of a hop T_n for a given control action a_n^{S} will depend on the previous hop period T_{n-1} subject to the following assumptions:

- The height of the ground does not change.
- The properties and behaviour of the ground are unchanging.
- The vertical velocity is the only significant change in robot state between touch-downs.
- The stance period is constant.

In order to control the hopping period a function of the following form is required:

$$a_n^{\mathrm{S}} = f_c\left(T_n, T_{n-1}\right) \tag{6.20}$$

If T_n is substituted by the desired hopping period T_d then the control action required can be determined:

$$a_n^{\mathrm{S}} = f_c\left(T_n \leftarrow T_d, T_{n-1}\right) \tag{6.21}$$

6.2.2 Application

In order to apply the step period control theory stated above the hopping leg experimental rig will be used (see chapter 4 and Fig. 4-1). A stance control action also has to be chosen. The action will be to apply a downward push with the foot using the open loop velocity controller. This controller uses the Jacobian so that control variables α and β can be used to create a horizontal or vertical motion/force at the foot.

A new variable V_c can be created so that:

$$\beta = -V_c \tag{6.22}$$

then positive values of V_c correspond to extending the leg and pushing downward with the foot (when vertical). The control action is then going to be to set and hold the variable V_c to a constant value throughout stance. It should be noted that a constant value of V_c does not accurately correspond to a downward velocity. This is because the open loop velocity controller is based on steady-state motion with no loading. It does still however correspond to some downwards force.

Substituting V_c into Eq. 6.19 gives:

$$T_{(n)} = f_2\left(T_{(n-1)}, V_{c(n)}\right) \tag{6.23}$$

The goal now is developing a controller to compute V_c so that the coming hopping period $T_{(n)}$ will match a desired value knowing that V_c will also be a function of the previous hopping period:

$$V_{c(n)} = f_c\left(T_{(n)}, T_{(n-1)}\right) \tag{6.24}$$

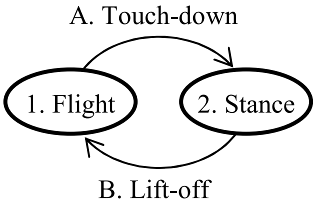


Figure 6-6: Controller states

6.3 Hopping while stationary

Consider the case when the step period converges to a steady-state value $T_{n-1} = T_n = T_{\text{ss}}$ for a constant control action V_c:

$$\begin{aligned} V_c &= f_c\left(T_{(n)} \leftarrow T_{\text{ss}}, T_{(n-1)} \leftarrow T_{\text{ss}}\right) \\ &= f_{\text{ss}}\left(T_{\text{ss}}\right) \end{aligned} \tag{6.25}$$

The function f_{ss} can be determined experimentally by setting different values of the control variable V_c and recording the corresponding steady-state hopping period T_{ss}.

In order to hop, the leg was programmed with a two-state controller as shown in Fig. 6-6:

- During the flight state, position control is used to demand the foot return to a home position $\mathbf{p}_h = (0, -0.56)$ m.
- During the stance state, the open loop velocity controller is set to push downwards with $(\alpha, \beta) = (0, -V_c)$.

Switching between states is triggered as follows:

- From flight to stance a threshold crossing on the actuator 2 (knee) load cell F_2.
- From stance to flight automatically when the stance state duration had exceeded 0.14 s.

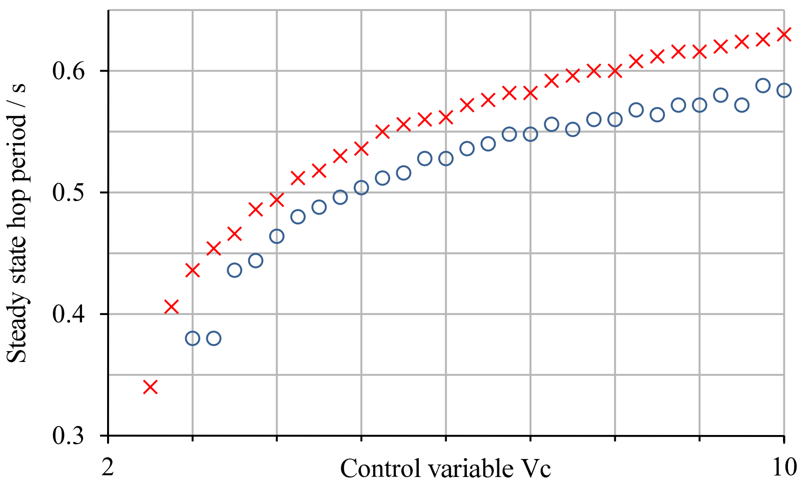


Figure 6-7: Relationship between control variable V_c and steady state hopping period time on hard (red crosses) and soft (blue circles) ground.

The time-out value of the stance state was selected so it would be longer than the true stance period (approximately 0.13 s). An alternative might be to add a means to detect when the foot lifts off the ground. This is not necessary however because the exact moment when lift-off occurs is not important. As long as the stance state lasts longer than actual stance, the forces applied during stance will be identical.

The hop period can be computed at each touch-down from the current time and the time at the previous touch-down. A plot of steady state hopping period T_{ss} against V_c is shown in Fig. 6-7. The relationship between steady-state hopping period and the control input changes with different ground properties and robot masses. The same plot also shows another set of data obtained by placing soft cloth matting on the floor (circles). Fitting a cubic equation to the hard ground data (crosses) gives a lookup function giving the control action V_c as a function of a desired steady state hopping period T_{ss}:

$$f_{\mathrm{ss}}(T) = 437.12T^3 - 517.4T^2 + 210.08T - 26.36 \tag{6.26}$$

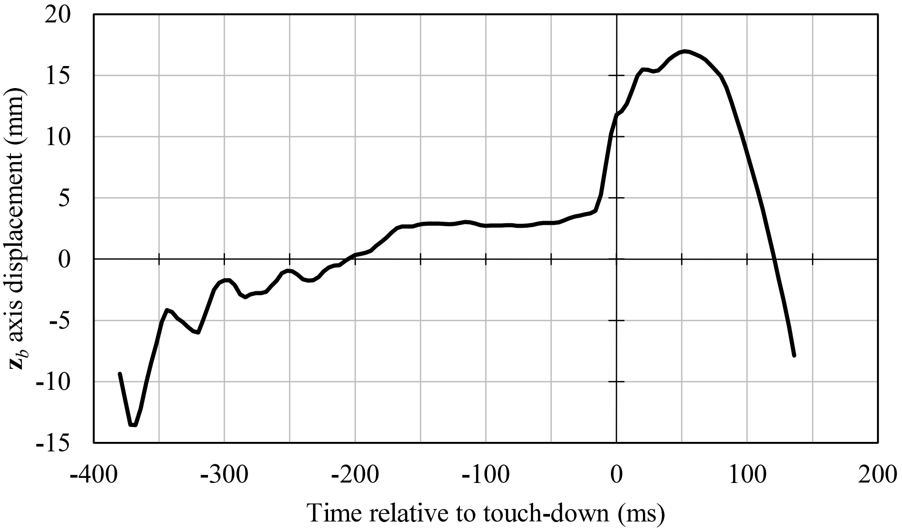


Figure 6-8: Displacement of foot relative to body in $\mathbf{z}_b$ axis, see Fig. 4-1, during one hop for steady state hopping with $V_c = 5$. The controller enters stance state at $t = 0$.

As mentioned earlier, open loop velocity control mode is used to generate a downward push during stance. The typical foot motion this results in is shown in Fig. 6-8. During the flight state, from $t = -0.38$ s to 0, position control mode attempts to return the foot to the home position. It can be seen that transients are successfully removed by the controller. An offset error of about 4 mm can be seen. Actual impact with the ground occurs shortly before $t = 0$ and the leg begins to flex. After entering stance control, velocity control mode is used to apply a downward force which eventually results in the leg extending. The stance control state ends at $t = 0.14$ s.

6.3.1 Feed forward control

The lookup function f_{ss}, Eq. 6.26, can be used in a feed forward hopping period controller. The results of doing this on a stationary treadmill (crosses) and soft ground (circles) are shown in Fig. 6-9. It can be seen that step changes in

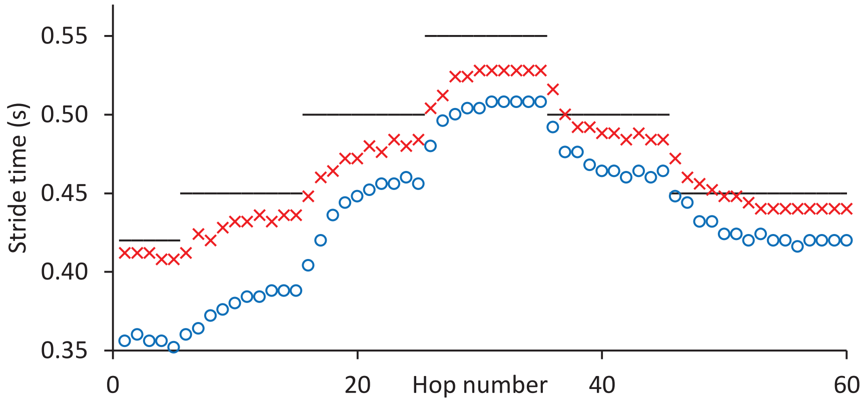


Figure 6-9: Feed forward only hopping control on different grounds: stationary treadmill (red crosses); soft mat (blue circles). Solid lines show demanded stride time. Steady state error is smaller on the treadmill because the controller references a function fitted to hard ground data.

demand take a few steps to track. Furthermore, there are steady state errors if ground properties do not match those of the feed forward lookup function. There is a steady state error on soft ground. Hopping on the treadmill also results in noticeable steady state errors. Steady state errors may also occur due to changes in hydraulic fluid properties between experiments, for example due to changes in oil temperature.

6.3.2 Feed forward + PI control

Feed forward control is a simple and stable way to control the hopping period/height but it takes several hops to converge to the demanded value and there are steady state errors. Both of these issues can be improved upon by forming a closed-loop controller using the error between the demanded stride time $T_{d(n)}$ and the most recent completed stride time $T_{(n-1)}$. An integral action on the error can remove steady-state errors and a proportional gain can improve dynamic performance. Adding these actions results in a feed forward + PI controller as illustrated in Fig. 6-10. Because the job of the integral gain is to remove steady state errors, the integral gain is switched to $K_I = 0$ when the demand is not steady

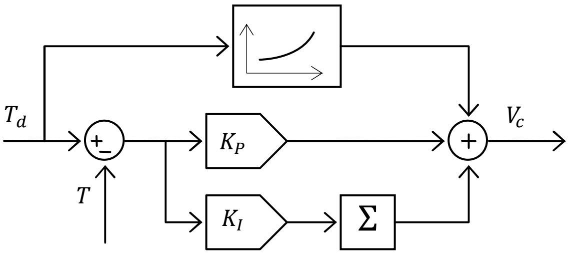


Figure 6-10: Block diagram of feed forward + PI hopping period controller. Feed forward function is given in Eq. 6.26.

$T_{d(n)} \neq T_{d(n-1)}$.

Tuning gains

The closed loop controller in Fig. 6-10 requires the tuning of two controller gains:

- The integral gain K_I was manually tuned with $K_P = 0$ to remove steady state errors without causing overshooting. A high K_I value can lead to instability however stable oscillations while overshooting occur before this making it easy to avoid an unstable K_I if the value is tuned up starting from zero. In the case here the goal was to remove steady-state error slowly over several, 10 or so, hops meaning a finely tuned K_I value was not required. Indeed, tuning K_I to remove steady-state errors much faster is not desirable because it would then start to affect the independent behaviour of the proportional action.
- The proportional gain K_P can be tuned manually but a more systematic and convenient method of self-tuning was adopted to allow faster tuning for different ground properties and robot parameters. The desired value of K_P is correction of error with no over- or under-shooting in a single hop. Slightly higher or lower values will result in over- or under-shooting but will be stable. They will however take several hops to converge to a steady state.

In order to investigate whether there is a difference between increasing the hopping period and decreasing it, the controller gain K_P was split into two:

$$K_P \leftarrow \begin{cases} K_P^+ & : T_{d(n)} - T_{(n-1)} > 0 \quad \text{(increasing hop period)} \\ K_P^- & : T_{d(n)} - T_{(n-1)} < 0 \quad \text{(decreasing hop period)} \end{cases} \tag{6.27}$$

To determine these values, the demanded hopping period was set to alternate every 3 hops between 0.43 s and 0.48 s. The values for K_P were then updated based on the error between the demand and actual hop period of the hop just completed. When stepping up, K_P^+ was automatically tuned:

$$K_{P\,(n+1)}^+ = K_{P\,(n)}^+ + \delta(T_{d(n-1)} - T_{(n-1)}) \tag{6.28}$$

When stepping down:

$$K_{P\,(n+1)}^- = K_{P\,(n)}^- + \delta(T_{(n-1)} - T_{d(n-1)}) \tag{6.29}$$

The value of δ can be used to change the rate at which the tuned gains converge on a final value. The value of δ does not require fine tuning as drifts in the proportional gain K_P are slow to occur due to their causes being slow phenomena such as changes in fluid temperature for example. Convergence over 100s of hops is acceptable although much faster tuning is possible. This convergence over 100 hops can be seen in Fig. 6-11. Starting from $K_P = 30$, it can be seen that tuning converges to a similar value regardless of whether the ground is hard (a) or soft (b) or whether attempting to increase or decrease hopping period.

Results

After tuning, the results for feed forward + PI control with step changes in demand are plotted in Fig. 6-12. The step demand results show the improvement over open-loop control (Fig. 6-9). A more challenging demand is shown in Fig.

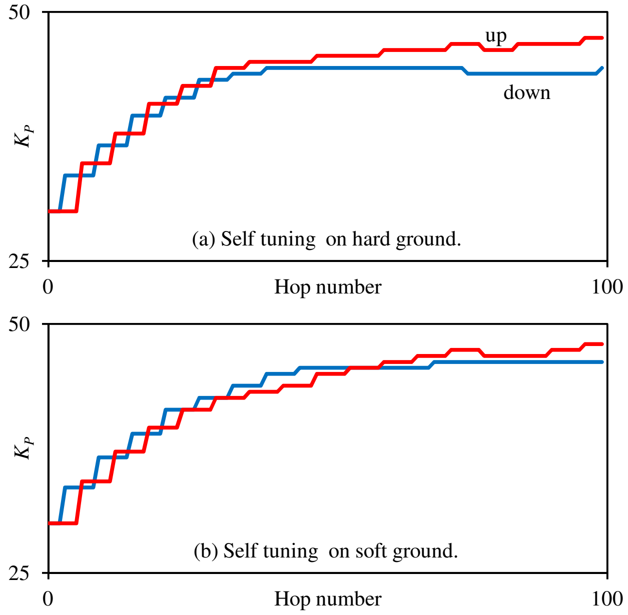


Figure 6-11: Convergence of controller gain $K_P^{\pm}$ by automatic tuning.

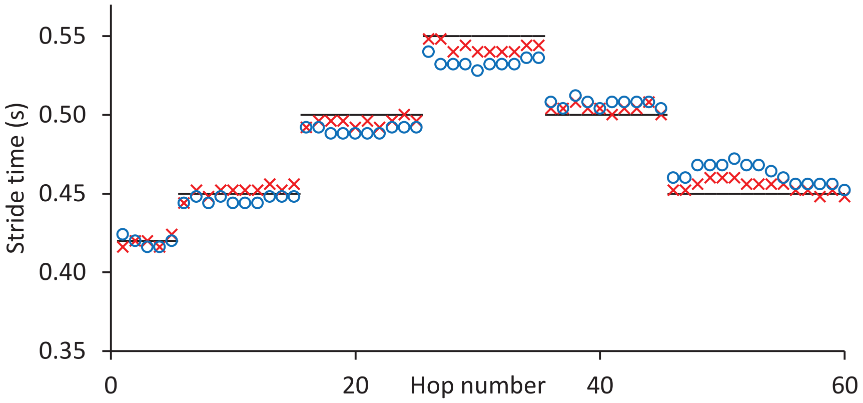


Figure 6-12: ‘Closed-loop’ hopping control on different grounds. Hard ground (crosses); soft ground (circles).

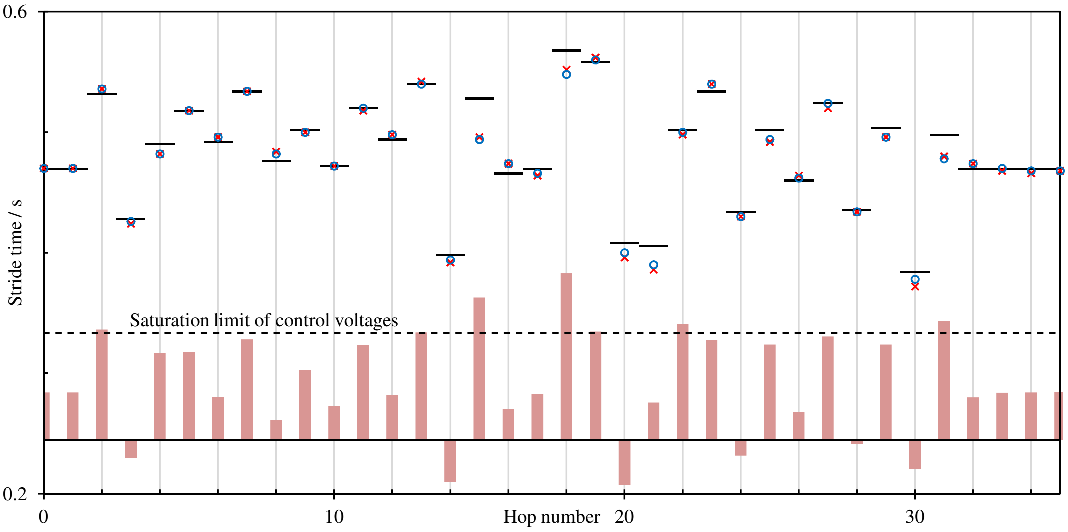


Figure 6-13: Results for PI+feed-forward controller with random hopping demand. Stride times range from 0.38 s to 0.58 s which corresponds to hopping heights of 0.08 m and 0.24 m respectively. Controller was auto tuned before beginning random demand input (horizontal lines). The same experiment was performed first on hard (crosses) ground then on soft ground (circles). The control variable V_c has also been plotted for the case of hard ground.

6-13. Here random hopping periods are demanded in the range 0.38 s to 0.57 s. This corresponds to hopping heights from 0.077 m to 0.237 m. It can be seen that the large shortfalls on hops 15, 18 and 31 occur because the control signal had reached saturation. This may be avoided by limiting demanded hopping periods to within the performance envelope of the robot. Additionally, it should be noted that some hops require a negative value for the control variable V_c. This means that the leg has to actively flex to absorb more energy than passive damping alone would accomplish.

6.4 Hopping while running

To simulate the effects of running on hopping control while neglecting considerations of balance, experiments were carried out on a treadmill in motion (Fig. 4-1).

6.4.1 Foot sweeping

When running, it is desirable to begin sweeping the foot backwards before touch-down with the ground. This reduces the severity of the impact with the ground because the foot's relative horizontal motion to the ground is removed. In order to begin sweeping the foot before touch-down the next touch-down time $t_{\mathrm{td}(n+1)}$ has to be anticipated. Additionally, the foot needs to be positioned slightly ahead of the desired foot position on touch-down so that as it sweeps backwards in the air it reaches the desired foot position upon impact.

Assuming the robot executes the current hop so that it lasts for approximately the demanded amount of time $T_{d(n)}$ then touch-down will occur at $t_{\mathrm{td}(n+1)} \approx t_{\mathrm{td}(n)} + T_{d(n)}$. Foot motion has to begin a little earlier at say:

$$t = t_{td(n)} + 0.8T_{d(n)} \tag{6.30}$$

If the foot is swept backwards at the robot's running speed u_{gr}, then the starting foot position before sweep x_{F} has to be ahead of the desired foot position at touch-down x_{td}. Giving:

$$x_{\mathrm{F}} = x_{\mathrm{td}} + 0.2T_{d(n)}u_{\mathrm{gr}} \tag{6.31}$$

The duration of stance is estimated to be $T_s = 0.13$ s. The foot will sweep a distance of $\Delta x \approx u_{\mathrm{gr}}T_s$. A reasonable value for the desired foot position upon touch-down would be $x_{\mathrm{td}} = \Delta x/2$. Combining these with Eq. 6.31 gives:

$$x_{\mathrm{F}} = \left(0.5T_s + 0.2T_{d(n)}\right) u_{\mathrm{gr}} \tag{6.32}$$

The foot will be positioned to this value using position control mode during flight and at the time computed by Eq. 6.30 velocity control mode will be used to demand velocity $\dot{p}_x = -u_{\mathrm{gr}}$ to match treadmill speed.

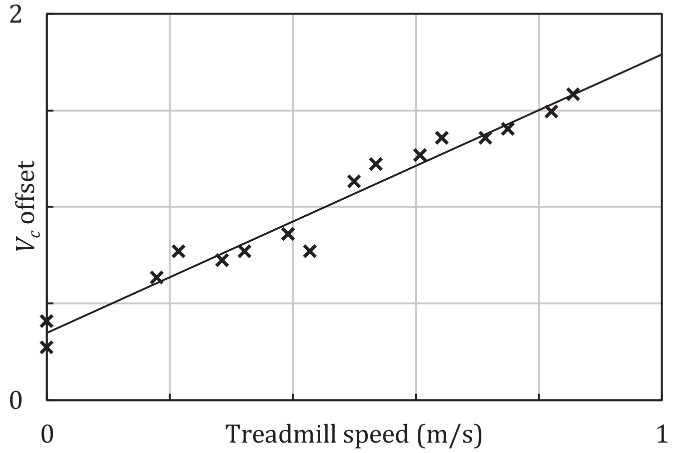


Figure 6-14: Corrective offset required to V_c, the nominal voltage output to extend the leg, at different running speeds to achieve 0.472 s steady state hopping period.

6.4.2 Running at different speeds

With foot sweeping added to the hopping controller, a set of hopping experiments were carried out with the treadmill in motion. A set of randomly varying hop period demand values was used in each experiment. The results of two such experiments are plotted in Fig. 6-15. Hopping on a stationary treadmill is compared against $0.37\,\mathrm{m\,s^{-1}}$. It can be seen that treadmill motion results in hop periods consistently lower than desired. A possible explanation for this is that while stationary, leg extension forces are directed vertically whereas running requires energy to be expended in swinging the leg, accelerating and decelerating the foot horizontally on each step. The same V_c therefore gives smaller hops when running.

In the case of the FF+PI control the steady-state error introduced by going from stationary hopping to running at speed, or by changing speed, is eventually corrected by the integration part of the control loop. The value of V_c required to achieve steady state hopping at different running speeds for a hop period demand

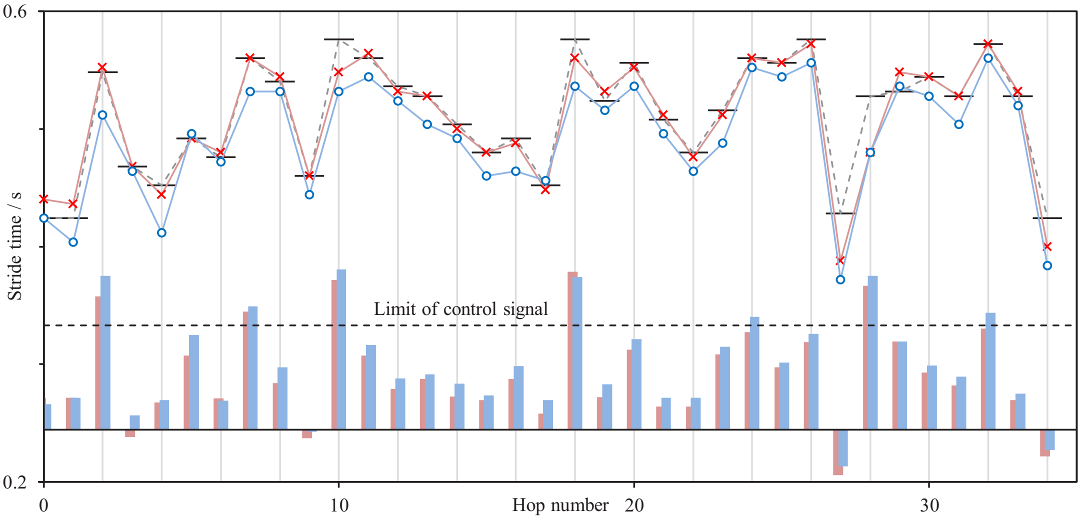


Figure 6-15: FF+PI control on treadmill at speeds 0 (red crosses) and 0.37 $\mathrm{m\,s^{-1}}$ (blue circles).

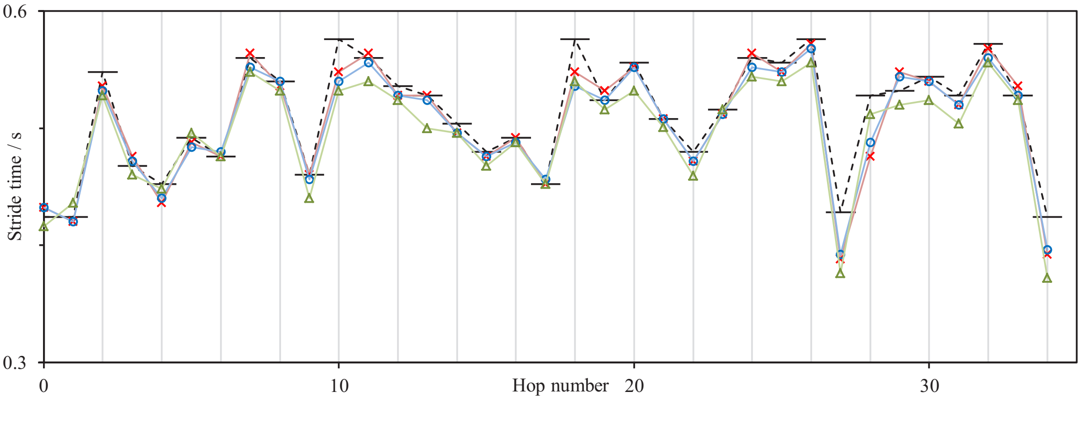


Figure 6-16: Extended FF+PI control at different running speeds: red crosses, blue circles, green triangles are 0, 0.37, 0.71 $\mathrm{ms^{-1}}$ respectively.

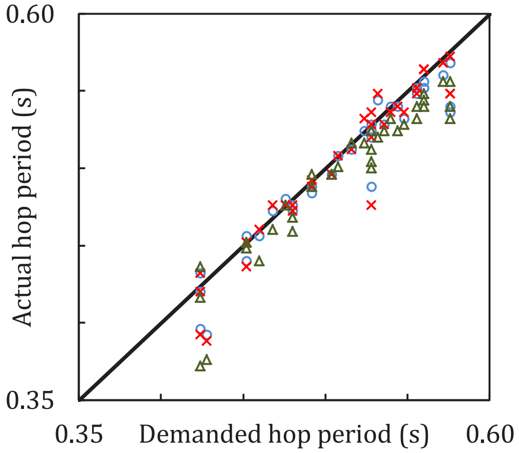


Figure 6-17: Demand vs actual plot.

$T_d = 0.472$ s has been plotted in Fig. 6-14. Fitting a line to this gives:

$$V_c = \underbrace{1.44}_{k_u} u_{\text{gr}} + 0.35 \tag{6.33}$$

The gradient k_u, which has units $\mathrm{m^{-1}\,s}$, can be used to calculate the addition needed to V_c to compensate for running at speed (assuming k_u does not vary with T_d):

$$\Delta V_c(u_{\text{gr}}) = k_u u_{\text{gr}} \tag{6.34}$$

Extending the V_c output of the FF+PI controller with this term allows the removal of steady state errors which occur due to changes in speed.

Experimental results of adding this speed based compensation are plotted in Fig. 6-16. Error is reduced. Large changes in the demand result in greater error. Hop 10 undershoots because of actuator saturation. Saturation occurs more frequently at high speed. For the same results, the hop period is plotted against the demand in Fig. 6-17.

The results for no speed compensation and no integral action are shown in Fig. 6-15 for $0\,\mathrm{m\,s^{-1}}$ and $0.37\,\mathrm{m\,s^{-1}}$. For $0.37\,\mathrm{m\,s^{-1}}$, a steady-state error with hops consistently smaller than demanded can be seen. Results with speed compensation (and still no integral action) are plotted in Fig. 6-16 with an additional set of data at $0.71\,\mathrm{m\,s^{-1}}$. With speed compensation, steady-state errors are greatly reduced. These results are also plotted in Fig. 6-17. At the highest running speed, $0.71\,\mathrm{m\,s^{-1}}$, the control signal is regularly saturated so performance is relatively poor. Nevertheless there is an appreciable improvement in steady-state error compared to Fig. 6-15.

It should be noted that for all of these results, the same feed-forward function and PI gains are used.

6.5 Conclusion

The control laws developed here are relatively simple. An alternative approach might be to employ actuators and sensors allowing for high speed force control. A model-based control loop, for instance, could then be developed to impart the required impulse to the ground during the stance phase. The work here provides an example of how, given limited sensing and computation, it is still possible to achieve performance which may be good enough for foot placement control while running on a variety of different terrain. This can be done, given favourable passive dynamics, by stacking simple laws to excite, maintain and perturb those dynamics.

The overall approach to control hopping taken here has been:

1. Use a machine with a passive hopping motion. Here this is due to a springy foot.

2. Formulate a variable to impart a vertical impulse, V_c, to be controlled discretely once per hop. This can then be used to form a discrete hop control loop executed once per hop.

3. Generate a look-up table/function for open loop, steady state control of

hop periods.

4. Improve steady state and dynamic performance by closing the loop with a simple proportional and integral action.

This approach could be applied to machines with different mechanical designs. For example:

- Pneumatic or electrical actuation might be used instead of hydraulics.
- It is not necessary that the leg is articulated. It could equally well be telescopic or some other design.
- Impact with the ground was detected as a spike in the force sensor at the knee but different sensors placed elsewhere would serve equally well.
- A passive hopping motion is required but this does not have to be provided by a springy foot. Indeed, elasticity might be emulated by the actuators. With real elasticity however, energy is stored and released from one hop to the next. This means that for steady state locomotion, actuators only need to make up energy losses between hops. And to change hop size, actuators need to make up (or dissipate) the energy difference. Actuators typically will not store energy so emulating elasticity would be inefficient. It would also require much more powerful actuators capable of responding to impact forces.

Balance was not a consideration in this chapter because the machine's body orientation was constrained but, as shown by Raibert et al in 1986, height control can be considered decoupled from body orientation and horizontal velocity. Nevertheless, especially at higher running speeds, this assumption cannot always be made. Further work can address this by building a more powerful and/or lighter machine and removing constraints on the body orientation and position.

By allowing movement in 2D or 3D this approach to controlling hops can be fully tested as one part of a 3-part control loop.

Chapter 7

Conclusion

7.1 Hypothesis support

An effort was made in this work to contribute to the development of robots able to traverse rough terrain. Specifically, the hypothesis stated in the Introduction chapter was:

> The foot placement of a robot designed for steady-state controlled passive dynamic running can be controlled by appropriately moving actuators during the ground contact phase.

To support this hypothesis:

- A control strategy was first developed in chapter 3. This was done by extending the SLIP model and analysing it mathematically. The strategy adopted was to modulate the extension rate of the actuator on each hop to change the lift-off velocity. A planar simulation of the extended SLIP model was used to show how modifying the flight trajectory in this way could be used to control foot placement while running.

- An experimental rig to validate this strategy was built as described in chapter 4. The rig consisted of a 2-link hydraulic leg with a spring-loaded foot. The body was constrained to move vertically only with a treadmill underneath. Using only joint position sensors and a load cell to detect ground contact, the strategy of leg extension rate modulation was validated experimentally in chapter 6. This is a novel demonstration of how a simple control law can be used to achieve sudden changes in hopping trajectory. Further, it was demonstrated how the control strategy might be extended to adapt to different ground properties and running speeds.

As an aside, it was found that PID control was inadequate for foot repositioning during the flight phase without inducing vibration. This problem was solved with a novel implementation of so-called 'zero-vibration' (ZV) shaping in a closed-loop in chapter 5. The application of closed-loop signal shaping (CLSS) to the control of hydraulics is novel. This allowed the use of a lower control loop sample rate, 200 Hz, and only position feedback.

Support for the hypothesis was provided by the effort made here. A comprehensive demonstration by building of a robot able to run and control its foot placement over rough terrain remains to be done. This requires further work on a number of threads initiated here as well as the integration of a number of different technologies. These are good topics for further research.

7.2 Limitations and further work

The experimental work carried out here constrained the robot body to move vertically, just one degree of freedom, removing considerations of balance. In doing so the assumption was made that the dynamics of height control are decoupled from body orientation and horizontal velocity. Whilst this assumption, Raibert's 3-part controller, is a successful one the dynamics are in fact coupled. The coupling is stronger for radical manoeuvres and high running speeds. It also depends on the mechanical design of the robot. In reality the body of a robot would be free to move in all 6 degrees of freedom. The general approach of modulating the foot motion during stance developed here should be tested alongside orientation control in simulation and experiment. The approach can then be developed further.

7.3 Final comment

Legged robots systems currently lack the performance required to address the applications for which they have been conceived. One of the potential places where legs might be more useful than wheels or tracks is on rough terrain or in environments designed for humans which have stairs for instance. This is because legs allow foot placement to be controlled so that discontinuous surfaces with isolated spots suitable for load bearing can be traversed. So the topic of foot placement in legged robotics is a practical one. It is also a stimulating and open topic requiring mechanical, electrical and software engineering whilst touching on biology. Nevertheless few researchers have addressed foot placement. This may be because legged machines are a multi-degree-of-freedom system and therefore the problem is presumed to be a complicated or low priority, one best left until steady-state locomotion is solved. The control problem however isn't necessarily a complicated one. Controlled passive dynamic walkers, Raibert's 3-part controller based running machines, and Jerry Pratt's Virtual Model Control based Spring Flamingo robot were the first to demonstrate this. The work done here additionally implies that simple control laws can be used to achieve agile locomotion: sudden, accurate changes in speed and direction. This is a necessary component of running on rough terrain.

References

[1] Y. G. Chen and M. H. Hsu, “Searching Ancient Inventions from Modern Techniques - A Research of Walking Horses with 8-Link Type Leg Mechanisms,” in *Proceedings of Twelfth World Congress in Mechanism and Machine Science*, 2007.

[2] “Walking Machines inc. Steam Man Timeline.” [Online]. Available: http://cyberneticzoo.com/

[3] M. Silva and J. Machado, “A Historical Perspective of Legged Robots,” *Journal of Vibration and Control*, vol. 13, no. 9-10, pp. 1447–1486, 2007.

[4] “Blog Archive » 1968 – Phoney Poney – Frank, McGhee (American).” [Online]. Available: http://cyberneticzoo.com/

[5] “Where Jules Verne Meets Star Wars: GE Walking Truck of the 1960s | GE Reports.” [Online]. Available: http://www.gereports.com/

[6] K. Waldron and R. B. McGhee, “The adaptive suspension vehicle,” *Control Systems Magazine, IEEE*, vol. 6, no. 6, pp. 7–12, 1986.

[7] “Not so long ago, in an OSU engineering lab nearby... | From Woody’s Couch.” [Online]. Available: http://library.osu.edu/

[8] E. Garcia, M. Jimenez, P. De Santos, and M. Armada, “The evolution of robotics research,” *IEEE Robotics Automation Magazine*, vol. 14, no. 1, pp. 90–103, 2007.

[9] M. H. Raibert, *Legged robots that balance.* MIT press Cambridge, MA, 1986.

[10] M. Buehler, R. Playter, and M. Raibert, “Robots step outside,” in *International Symposium on Adaptive Motion of Animals and Machines*, 2005, pp. 1–4.

[11] “Boston dynamics: Dedicated to the science and art of how things move.” [Online]. Available: http://www.bostondynamics.com/

[12] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, “Autonomous navigation for BigDog,” in *2010 Int. Conf. on Robotics and Automation*, Anchorage, AK, USA, May 2010, pp. 4736–4741.

[13] “BigDog Overview (Updated March 2010),” Apr. 2010. [Online]. Available: http://www.youtub.be/cNZPRsrwumQ

[14] M. Raibert, K. Blankespoor, G. Nelson, R. Playter *et al.*, “BigDog, the Rough-Terrain Quadruped Robot,” in *Proceedings of the 17th World Congress*, 2008, pp. 10 823–10 825.

[15] “Boston Dynamics Youtube channel.” [Online]. Available: http://www.youtube.com/user/BostonDynamics

[16] G. Pratt, “Legged robots at MIT: what’s new since Raibert?” *IEEE Robotics Automation Magazine*, vol. 7, no. 3, pp. 15–19, 2000.

[17] J. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, “Virtual model control: An intuitive approach for bipedal locomotion,” *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 129–143, Jan. 2001.

[18] E. Garcia and P. Gonzalez de Santos, “Controlling dynamic stability and active compliance to improve quadrupedal walking,” *Climbing and Walking Robots*, p. 205–212, 2006.

[19] T. Bretl and S. Lall, “Testing static equilibrium for legged robots,” *Robotics, IEEE Transactions on*, vol. 24, no. 4, p. 794–807, 2008.

[20] M. Hirose and K. Ogawa, “Honda humanoid robots development,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 365, no. 1850, pp. 11–19, Jan. 2007.

[21] P. Sardain and G. Bessonnet, “Forces Acting on a Biped Robot. Center of Pressure–Zero Moment Point,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 34, no. 5, pp. 630–637, 2004.

[22] M. Vukobratovic and B. Borovac, “Zero-moment point–thirty five years of its life,” *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[23] “Honda Worldwide | ASIMO | History.” [Online]. Available: http://world.honda.com/ASIMO/history/

[24] J. E. Pratt, “Exploiting inherent robustness and natural dynamics in the control of bipedal walking robots,” DTIC Document, Tech. Rep., 2000.

[25] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, “Efficient bipedal robots based on passive-dynamic walkers,” *Science*, vol. 307, no. 5712, pp. pp. 1082–1085, 2005. [Online]. Available: http://www.jstor.org/stable/3840156

[26] T. McGeer, “Passive dynamic walking,” *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990.

[27] S. Lipfert, M. Günther, D. Renjewski, and A. Seyfarth, “Impulsive ankle push-off powers leg swing in human walking,” *The Journal of Experimental Biology*, 2013.

[28] “MIT Leg Laboratory,” Nov. 2010. [Online]. Available: http://www.ai.mit.edu/projects/leglab/

[29] M. Raibert, M. Chepponis, and H. Brown, “Running on four legs as though they were one,” *IEEE Journal of Robotics and Automation*, vol. 2, no. 2, pp. 70–82, Jun. 1986.

[30] S. Cotton, I. Olaru, M. Bellman, T. van der Ven, J. Godowski, and J. Pratt, “Fastrunner: A fast, efficient and robust bipedal robot. Concept and planar simulation,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 2358–2364.

[31] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim, “Design principles for highly efficient quadrupeds and implementation on the MIT

Cheetah robot," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 3307–3312.

[32] J. Hodgins and M. Raibert, "Adjusting step length for rough terrain locomotion," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 289–298, Jun. 1991.

[33] J. Prosser and M. Kam, "Control of hopping height for a one-legged hopping machine," *Mobile Robots VII*, vol. 1831, p. 604–612, 1992.

[34] A. Lebaudy, J. Prosser, and M. Kam, "Control algorithms for a vertically-constrained one-legged hopping machine," in *IEEE Conference on Decision and Control*, vol. 3, 1993, pp. 2688–2693.

[35] M. Ahmadi and M. Buehler, "Controlled passive dynamic running experiments with the ARL-monopod II," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 974–986, Oct. 2006.

[36] S. Claudio, "HyQ – Design and Development of a Hydraulically Actuated Quadruped Robot," PhD, University of Genoa, Italian Institute of Technology, Apr. 2010.

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

[38] "NI CompactRIO." [Online]. Available: http://www.ni.com/compactrio/

[39] J. R. Huey, K. L. Sorensen, and W. E. Singhose, "Useful applications of closed-loop signal shaping controllers," *Control Engineering Practice*, vol. 16, no. 7, pp. 836 – 846, 2008.

[40] J. Bhatti, A. R. Plummer, P. Iravani, and M. N. Sahinkaya, "Implementation of closed loop signal shaping in a hydraulic system," in *The 13th Mechatronics Forum International Conference*, 2012.

[41] A. R. Plummer, “Control techniques for structural testing: a review,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 2, 2007.

[42] K. A. Edge, “The control of fluid power systems-responding to the challenges,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 211, no. 2, 1997.

[43] W. Singhose, “Command shaping for flexible systems: A review of the first 50 years,” *International journal of precision engineering and manufacturing*, vol. 10, no. 4, pp. 153–168, 2009.

[44] Z. Mohamed and M. O. Tokhi, “Command shaping techniques for vibration control of a flexible robot manipulator,” *Mechatronics*, vol. 14, no. 1, 2004.

[45] D. Lewis, G. Parker, B. Driessen, and R. Robinett, “Command shaping control of an operator-in-the-loop boom crane,” in *American Control Conference, 1998. Proceedings of the 1998*, vol. 5, Jun. 1998, pp. 2643 –2647 vol.5.

[46] Z. Mohamed and M. O. Tokhi, “Vibration control of a single-link flexible manipulator using command shaping techniques,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 216, no. 2, 2002.