
THE UNIVERSITY *of* EDINBURGH

# LEGGED ROBOTIC LOCOMOTION WITH VARIABLE IMPEDANCE JOINTS

ALEXANDER M ENOCH


Doctor of Philosophy

Institute of Perception, Action and Behaviour
School of Informatics
University of Edinburgh

2016

# LAY SUMMARY

Humans have a complex arrangement of bones, muscles and tendons that allows them to perform a wide variety of tasks and movements. As well as simply adjusting the position of our joints, we can also modify the dynamic properties of the joints - for example increasing the stiffness of a joint by tensing certain sets of muscles. This is in part responsible for the amazing flexibility and efficiency of human movement.

Variable impedance - the ability to change dynamic properties such as stiffness and damping - has become a large field in robotics, and tailoring the impedance of a robot to a particular task can improve efficiency, stability, and potentially safety. A two legged robot walking is a perfect example of a task for which variable impedance may provide such advantages.

This thesis details the creation of two novel bipedal robots with variable impedance joints. These robots aim to achieve some of the benefits of compliance (springiness), while retaining the ability to do many different tasks and be truly versatile machines. Many designs for variable impedance joints have been published, and the field is explored and evaluated before the design of the robots is presented. Rapid prototyping techniques such as 3D printing were used in the creation of both robots, and both robots are based around a custom, high performance electronics and communication system.

The human walking cycle was analysed and a simple parametrised representation developed. A framework was built to produce human-like walking movements which would function on the robot, and these were tested using dynamic simulations before being tried on the robot BLUE. With the robot capable of locomotion, the effect of varying stiffness on efficiency was explored, and we found that changing the stiffness of the robot can have an effect on energy efficiency. We introduce a system for controlling the robot by detecting walking movements of an operator and attempting to copy the goals of the movement, rather than copy the joint angles directly. In this way, the robot can produce the same overall locomotion as the human, but with joint trajectories and stiffness levels are more suited for its dynamics.

# ABSTRACT

Humans have a complex musculoskeletal arrangement which gives them great behavioural flexibility. As well as simply moving their legs, they can modulate the impedance of them. Variable impedance has become a large field in robotics, and tailoring the impedance of a robot to a particular task can improve efficiency, stability, and potentially safety. Locomotion of a bipedal robot is a perfect example of a task for which variable impedance may provide such advantages, since it is a dynamic movement which involves periodic ground impacts.

This thesis explores the creation of two novel bipedal robots with variable impedance joints. These robots aim to achieve some of the benefits of compliance, while retaining the behavioural flexibility to be truly versatile machines. The field of variable impedance actuators is explored and evaluated, before the design of the robots is presented. Of the two robots, BLUE (Bipedal Locomotion at the University of Edinburgh) has a 700mm hip rotation height, and is a saggital plane biped. miniBLUE has a hip rotation height of 465mm, and includes additional joints to allow hip adduction and abduction. Rapid prototyping techniques were utilised in the creation of both robots, and both robots are based around a custom, high performance electronics and communication architecture.

The human walking cycle is analysed and a simple, parameterised representation developed. Walking trajectories gathered from human motion capture data, and generated from high level gait determinants are evaluated in dynamic simulation, and then on BLUE. With the robot being capable of locomotion, we explore the effect of varying stiffness on efficiency, and find that changing the stiffness can have an effect on the energy efficiency of the movement. Finally, we introduce a system for goal-based teleoperation of the robots, in which parameters are extracted from a user in a motion capture suit and replicated by the robot. In this way, the robot produces the same overall locomotion as the human, but with joint trajectories and stiffnesses that are more suited for its dynamics.

# ACKNOWLEDGMENTS

I would like to say a very big thank you to Andrius Sutas, for his help with the electronics, and a boat load of moral support.

Thanks to Dougie, Gilbert, Hugh and Colin in the workshop, for helping with the fabrication of the robots.

And also a gigantic thank you to Alex, for keeping me somewhat sane through the course of this PhD, and for the awesome cooking.

# DECLARATION

I declare that this thesis has been composed by myself and that the work has not been submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where work which has formed part of a jointly-authored publication has been included. My contribution and those of the other authors to this work have been explicitly indicated below. I confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

A summary of the work presented in chapter 2, and parts of the work presented in chapter 3 and section 5.3 regarding the robot BLUE were previously published in the *Proceedings of the 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2012* as *BLUE: A bipedal robot with variable stiffness and damping* by A. Enoch (Student), A. Sutas, S. Nakaoka, and S. Vijayakumar (Supervisor). Andrew Sutas, working as a lab intern, produced the layouts for the circuit boards detailed in sections 4.4 and 4.5, wrote the code for the low level Ethernet communications detailed in 4.6, the daemon script introduced in section 4.7.1, and assisted with parts of the firmware. Shin'ichiro Nakaoka produced the Choreonoid software used in section 3.4.1. I conducted all other work under the supervision of Sethu Vijayakumar.

The work presented in chapters 2, 3, and parts of 5 was previously published in the *ASME Journal of Mechanisms and Robotics* as *Rapid Manufacture of Variable Impedance Robots* by A. Enoch (Student) and S. Vijayakumar (Supervisor). Except as previously acknowledged, I conducted all work under the supervision of Sethu Vijayakumar.

The work presented in Chapter 7 was previously published in *Proceedings 7th IEEE International Conference on Robotics, Automation and Mechanisms (RAM) 2015* as *A Three Degree of Freedom Leg Design and Implementation in a Bipedal Robot* by A. Enoch (Student). This was a sole author paper.

Alexander M Enoch,

*Edinburgh, 2016*

# PUBLICATIONS

The following publications have been composed during the course of this doctorate:

A. Enoch, A. Sutas, S. Nakaoka and S. Vijayakumar. BLUE: A bipedal robot with variable stiffness and damping. *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pp 487–494, 2012.

A. Enoch, and S. Vijayakumar. Rapid Manufacture of Variable Impedance Robots. Accepted to *ASME Journal of Mechanisms and Robotics*.

A. Enoch. A Three Degree of Freedom Leg Design, and Implementation in a Bipedal Robot. Accepted to *7th IEEE International Conference on Robotics, Automation and Mechatronics (RAM) 2015*.

# CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

# Part I

# INTRODUCTION & BACKGROUND

# 1

# INTRODUCTION & THE BIOMECHANICS OF HUMAN LOCOMOTION

## 1.1 INTRODUCTION

Human walking is a complex dynamic process, involving moving through positions which are not themselves statically stable, to create an overall periodic movement that is stable. Impacts with the ground must be dealt with, and the whole motion should be as efficient as possible - the amount of energy lost at each impact should be reduced, and the natural dynamics of the body should be utilised wherever possible.[91][28] The arrangement of our joints, muscles and tendons give us the ability to do more than just control the position of our limbs: we can store energy, absorb shock forces, and adjust the stiffness and damping of our joints. [71]

Importantly, through the interplay of these mechanisms and neural control we are able to produce walking behaviour which is remarkably efficient, taking advantage of the natural dynamics of our bodies. At the same time, human walking is incredibly adaptive to variations in the terrain, usually without requiring conscious effort from the person walking. Mechanical compliance allows the body to react to disturbances, while various levels of reflexive loops and top down control interact to adjust the walking behaviour. [27][42] From the stiff but elastic heels of our feet, [40] to landing each step with a slightly bent knee [103] to allow flex, our bodies also work to absorb shock loading from ground reaction forces.

Utilising compliance can also reduce torque loading on actuators, and improve dynamic task performance. [7] In robotics, compliance is often cited as being important for decoupling link inertias and reducing damage to people/objects in the event of collisions. [5]

It is very difficult to deconstruct human walking behaviour to discover, for example, how the stiffness of our joints changes as we walk, or as we change the speed at which we are walking. Similarly, it is difficult to explore the effect that varying stiffness has

on walking performance, on efficiency, joint torques, ground reaction forces, and the like, since you cannot instruct a human to simply assume a particular stiffness during motion.

In the field of robotics bipedal locomotion has been a hot topic for some time. Typically bipedal robots used only rigid joints, and were largely incapable of taking advantage of their dynamics. (e.g. [81]) There was a movement to create passive dynamic walkers, which were mechanical designed with compliance and no (or very little) actuation to produce a walking gait. However, these robots can only walk at one speed, and are usually incapable of even standing still. [13] The introduction of fixed compliance into a fully powered walker gives more options for energy storage and shock absorption, but is not adaptable to take advantage of the plant dynamics during different movements. Furthermore, for such robots to become rigid they must employ active stiffness control - opposing induced changes in joint angles through high bandwidth control through the compliant element.

Only recently have researchers begun to look at bipedal robots which are capable of varying their joint impedance, and very little hardware has been constructed. When this project began, there were few bipedal robots capable of varying their mechanical impedance. [97][93][39]

## 1.2 THIS THESIS

This project aimed to create robust bipedal robotic platforms which allow the control of joint impedance. We created two bipedal robots, BLUE and miniBLUE, both of which have this ability, and are designed to be able to independently vary stiffness and damping. The creation of variable impedance bipedal robots allows behaviourally flexible locomotion, with the ability to increase the rigidity of the joints, for example when discrete movements are required.

The robots are designed as platforms to investigate the role of stiffness and damping in bipedal locomotion. Whilst the dynamics of the robot are quite different from those of a human, with appropriate cost functions the constructed platforms could also serve as useful tools for investigating underlying principles of bipedal locomotion, especially with regard to modulating stiffness.

The primary goal of this project is to create robots which can walk with an efficient human-like heel-toe gait, and which have the ability to vary the dynamics of their joints. It follows that morphologically, these platforms should follow a humanoid design, however it is not feasible to attempt to closely copy human musculature, as will be illustrated in this chapter. There are many ways of achieving variable impedance, and since it is this ability we wish to copy, rather than the specific method for doing so, we will instead consider the most suitable mechanisms for this task.

This thesis considers the field of variable impedance mechanisms, which has seen an explosion in recent years. A review of these mechanisms is presented, and the suitability of each for legged locomotion evaluated.

The bipedal robots BLUE, which is purely planar, and miniBLUE, which has two degrees of freedom in each hip, will be introduced. The construction methods are presented, and include rapid prototyping technologies, 3D printing for compliance, and a technique for combining 3D printing with waterjet or laser cutting. We have also developed a modular high performance electronics infrastructure and communications protocol, which is used on both robots.

Our methods for transferring walking motions from humans to the robots will be introduced, along with a method for generating trajectories from gait transition events. Control strategies for exploiting stiffness were investigated, for example utilising cost functions based on reducing torque and increasing efficiency.

Preliminary results from varying the impedance of the joints are presented for both simulation and on the actual hardware. The simulations show that at very slow speeds, there is no real predicted energy benefit from decreasing stiffness, although maximum torques may decrease. At faster walking speeds, the simulations predicted that decreasing stiffness would generally lead to a decrease in energy usage a well as decreased peak torques. On the real robot, decreasing stiffness for walking at very slow speeds was associated with an increase in energy consumption. For faster, more human-like walking speeds, the energy consumption was largely consistent across stiffness levels, while peak torques decreased with stiffness level.

The preliminary results showed that the robot was capable of producing locomotion at a range of fixed stiffness levels, and that changing stiffness can effect the energy usage and torque magnitude. The results indicated that at fast, dynamic walking speeds, even if a predicted decrease in energy usage did not occur, the walk was

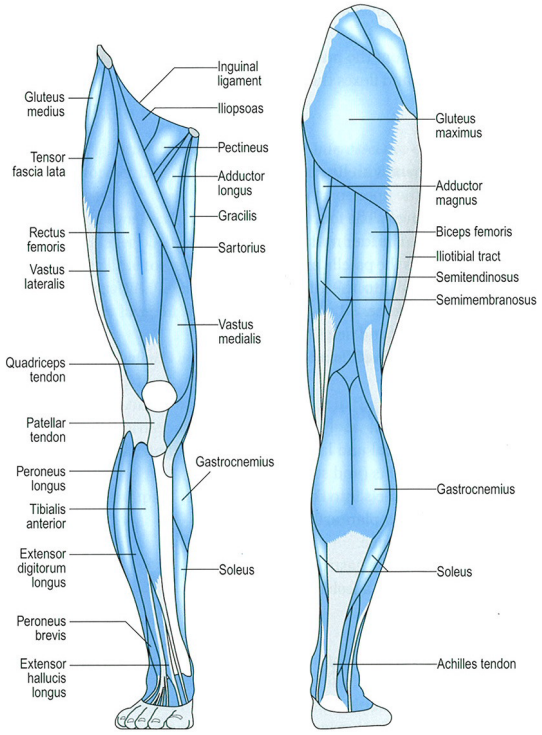


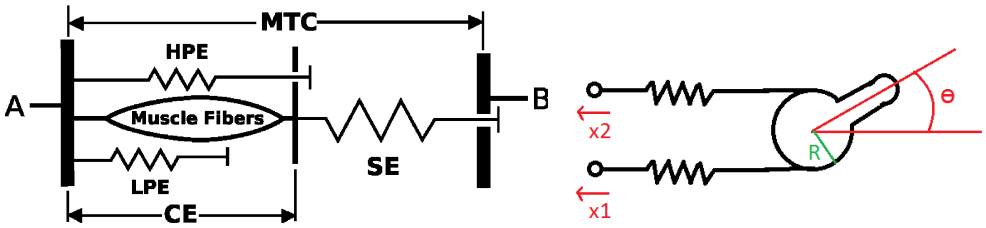

Figure 1: Superficial muscles of the right leg[103]

at least as efficient with lower stiffness settings, while subjecting the robot's joints to lower torques, and being more protected from shocks and disturbances.

Furthermore, we present teleoperation based on copying parameters from a user's walk, with the joint trajectories and stiffness of the robot being set according to pre-optimised values which reproduce the salient parameters.

In addition to the work on variable impedance locomotion, a design is also presented for a bipedal robot with three degree of freedom legs. Each leg utilises two four bar linkages, and only one strong motor is required per leg. Despite the minimal requirements for actuators, the robot is capable of walking, turning, kicking, and performing other motions.

Firstly though, to explain the complexity of the problem at hand, it is necessary to consider the complexity of the human locomotive system.




(a) Muscle Tendon Complex[18] (b) Basic Antagonistic Layout

Figure 2: Model of a MTC based around a Hill muscle model, and a schematic of a simplistic antagonistically actuated joint

## 1.3 HUMAN JOINT ACTUATION AND DYNAMICS

The musculature of the human leg is highly complex, as shown in figure 1, with many muscles and tendons stretching across one or two joints. The muscles provide pulling force, whilst the tendons are the principle elastic elements in the legs[55][71]. The tendons themselves are elastic with a non-linear spring characteristic, which actually hardens as they stretch[61]. Each Muscle Tendon Unit (MTU) or Complex (MTC) can be modelled as a combination of elastic elements and a Contractile Element (CE) - the muscle fibres, as shown in figure 2a. The CE has a damping component, and so the whole assembly exhibits a non-linear force response[18].

We can consider the principle of biological actuation of joints at its most basic by looking at a simple antagonistic arrangement as shown in figure 2b. In this figure two contractile elements are connected, one to x2 and one to x1. As each contracts they exert a pulling force and thus a turning moment on the joint. A full analysis of this system, and the need for non-linear springs, will be given in the next chapter, but the principles can be easily understood by observation. If one CE contracts it will pull the joint towards it, if both CEs contract the joint will encountered increased resistance as it attempts to move, and thus the stiffness of the joint is increased by *co-contraction*. Damping is also involved in the muscles, and will also change as the CEs are activated, although with a complex coupled relationship to stiffness. The same principles as apply to this basic model apply to the joints of the leg, and thus we can see that the stiffness and damping of human joints can be modulated.

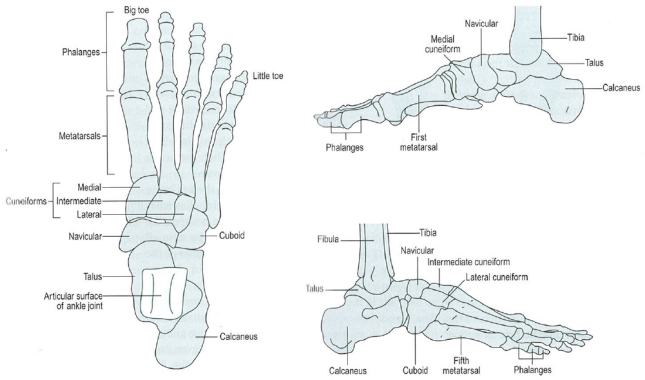



Figure 3: The bones of the foot[103]

### 1.3.1 *The Foot*

The human foot is composed of a great number of small bones which make it capable of moulding to the contours of the ground and flexing to absorb shocks and store energy in ways which a rigid foot could not[53]. In order to reproduce a human-like walking pattern, it is important to mimic the foot as closely as possible, since it is specialised for handling the heel-toe walking gait. The bones of the foot are shown in figure 3.

The toes are important for walking since they spread the weight bearing area which is available during the latter stages of the stance phase of walking, before toe-off[53]. In this way the large forces during the second peak of vertical loading are not just borne by the heads of the metarsals. The toes also play a part in the "*windlass mechanism*", which will be discussed shortly.

The longitudinal arch of the foot (sometimes subdivided into a major medial arch and a less-flexible lateral arch) is an important structure for absorbing shocks and providing elasticity in the foot.[47] Height changes in the longitudinal arch primarily occur at the transverse tarsal (or midtarsal) joint, which is comprised from articulation of the talonavicular joint on the medial side of the foot, and the calcaneocuboid joint on the lateral side [53]. Ker et al [50] conducted experiments to determine the elasticity and energy storage capability of the "spring in the arch of the human foot". They determined that 'the arch of the foot stores enough strain energy to make running more energy efficient", and it can be observed that since the arch deforms during walking it will be storing energy then as well.[26][15]. Since the arch is not a simple

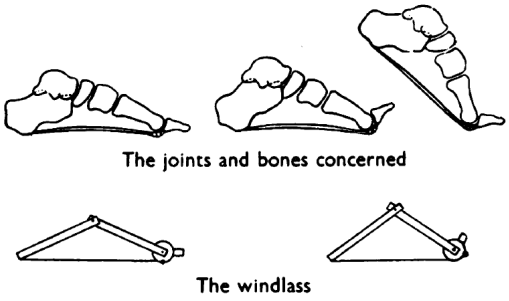



Figure 4: The windlass mechanism[37]

joint its elastic behaviour can change as the foot moves and there is evidence that the arch is softer during loading and stiffer towards toe-off, due purely to the morphology of the foot:

> [T]he foot is as flexible as possible to be an effective shock absorber during the loading response at midstance in the gait cycle... [T]here is rigidity at the midtarsal joint at toe-off when the foot is supinated so that it can act as an effective lever for propulsion[53]

The *windlass mechanism* is a passive mechanism which relates the extension (dorsiflexion) of the toes to a raising of the longitudinal arch.[37] The mechanism can be explained basically as a cable which attaches at the back of the arch, wraps around the metatarsal heads and attaches to the toes. When the toes extend the cable is wrapped around the 'drum' of the metatarsal heads and the arch is thus pulled taught. This principle is illustrated in figure 4, which shows the bones of the foot and a mechanical approximation below. The windlass mechanism is important since it helps to resist some of the loading during the latter half of the stance phase of walking, and provides energy back during toe off[37][47]. It should be noted, however, that there has been some research to suggest that the windlass mechanism may not be as passive as previously thought, with one paper finding that in some subjects the mechanism is subject to a delay.[48]

## 1.4 WALKING

Walking can be considered as a periodic movement which alternates between phases of double support, where both feet are on the ground, and single support, where

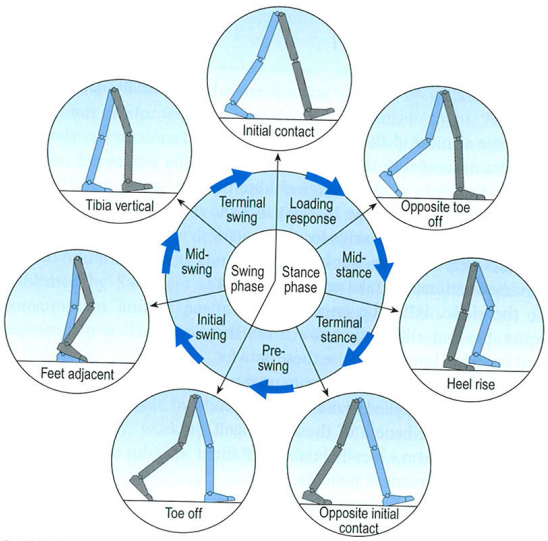



Figure 5: The stages of walking[103]

only one foot is on the ground. Considering one leg at a time, a walking cycle can be broken down as shown in figure 5 - with the stance phase when the foot of that leg is on the ground, and the swing phase when the foot is in the air.

While it could be tempting to view each step of the walking cycle as a vault over the leg on the ground, in fact walking is a bouncing gate[28], with significant amounts of kinetic energy from the swing leg being stored in elastic elements and preserved during the stance phase.[71] Alexander [2] notes three primary uses for springs in legged locomotion: energy storage in a bouncing gait, smooth return of oscillating joints (e.g. the hip), and shock absorption (e.g. the pad of the foot). The effect of walking at different speeds should also be noted; it seems that our self selected natural walking speed is a compromise between the most efficient frequency mechanically, and the speed at which our metabolic conversion of energy from chemical to kinetic is most efficient.[91] The salient features of human walking will be discussed later in this work, while specifying the design criteria for the robot, and in order to generate human-like walking trajectories specialised for the robot.

## 1.5 MEASURING STIFFNESS DURING WALKING

While it is evident that joint stiffness can be modulated in the legs, it is not a trivial task to measure it. Kinematic and dynamic analysis could be used to provide

many possible stiffness profiles for a given movement, but by itself cannot determine which is actually used. Ideally we would like to pause a person mid-walk, administer a small perturbation to a joint and measure the resulting force to get an idea of stiffness. Obviously, due to the nature of walking as a dynamic process, this is not possible. Small random perturbations have been applied to arm movements during reaching tasks in order to try to calculate stiffness[3], with limited success. Trying to measure stiffness with perturbations faces the additional problem that we may encounter reflexive effects which will resist these disturbances, and appear as increased stiffness[59].

ElectroMyoGraphic (EMG) recordings can be used to try to determine muscle activations during movement. Typically these recordings have been very noisy, and it has been difficult to isolate individual muscle signals from those of nearby muscles. Recent techniques utilising probabilistic modelling of the EMG signal are starting to yield results in predicting individual muscle activation[55], and it is suggested that by combining these with kinematic models of muscle activation the EMG signals for deeper muscles could be discerned. A combination of force plate recordings and x-ray recordings was used to try and determine the stiffness of the plantar fascia in the foot during walking[26]. This technique is not widely applicable to measuring muscle/tendon lengths throughout the leg, but the use of ultrasound instead of x-rays may provide the best yet estimates of muscle activation and joint stiffness during locomotion.

Recently, detailed muscle models, tuned to match empirical behaviour, have been used to estimate joint torque and stiffness in the knee from EMG sensors on three pairs of antagonistic muscles. [49] The authors built an assistive compliant knee exoskeleton, which utilises a variable stiffness actuator.

It remains a difficult problem to extract detailed joint stiffness information, especially during a task like walking. Furthermore, it is even more difficult to get a subject to produce a specific joint stiffness profile. A robotic platform which can vary the impedance profile of its joints could be a valuable tool for investigating the effect of varying stiffness on locomotion.

The robot built and used for this project has quite a different mechanism for varying impedance, as will be illustrated later. Unlike a human, it does not cost the robot any energy to hold its joints at a high stiffness. In order to study strategies suitable

for human walking, it would be possible to utilise a cost function which did penalise high stiffnesses. For producing efficient locomotion on the robot, it does not make sense to try to copy human stiffness levels directly, for example because of the differences in compliant mechanism energy usage and leg mass distribution. For this reason, in this work we explore ways of copying the behaviour of a human to the robot (e.g. locomotive or discrete movements), while the robot assumes a stiffness level appropriate to that motion.

## 1.6 ROBOTIC APPROACHES TO WALKING

Walking robots can be divided into several types: rigid drive, (actuated) passive dynamic, series elastic, and variable impedance. Rigid drive robots are the most common, with famous robots like Honda's Asimo[38] and the HRP series of robots[46] falling under this category. With no real compliance, the position of these robots joints is rigidly controlled, and typically the efficiency of movements such as walking is very low compared to humans and passive dynamic walkers. An exception to the inefficient rigid-drive robot is that developed by Braun[9], where back-driving of the motors is encouraged to allow the robot joints to swing. Whilst this produces a more efficient walking gait, which does not require the joint trajectory to be completely prescribed and tracked, it cannot store and release energy in compliant elements.

If the joints in a rigid-drive robot are torque-controlled, rather than position controlled, compliance can be simulated[41][72]. This involves a fast feedback loop which can detect the torque the motor is applying, and follow a trajectory in torque space. With this control strategy *active compliance* is achieved, for example if the robot encounters an obstacle, it will not exert excessive force trying to power through the obstacle, but will exert a force on the obstacle based on the torque commands of the joints.

The introduction of a compliant element between the motors and the joints is known as Series Elastic Actuation, and was first introduced by Pratt.[76] With this type of setup energy can be stored and shocks absorbed, however the mechanical compliance is fixed, and hence so are the dynamics. The effective compliance can be varied in a similar way to torque control, by using the motor to modulate forces applied through the spring. The robot Baxter utilises SEA actuators in order to improve

safety and prevent the robot from damage caused by disturbances in its environment. [31] The compliant version of the iCub, the cCub, uses a type of SEA with fairly rigid links,[88] as does the bipedal robot COMAN [89]

On the other end of the spectrum from rigid-drive robots are passive dynamic walkers. These are specifically designed to take advantage of the natural dynamics of the humanoid form to provide perambulation effectively as a result of a controlled fall. The most basic of these robots are purely mechanical and require downhill slopes and gravity to provide power. More advanced actuated passive dynamic walkers utilise the principle of passive-dynamic walkers with one or two powered joints in order to power motion.[13] Whilst these robots are very efficient, they are typically only capable of locomotion at one speed, cannot stand still, and outside of locomotive behaviour have very little control over their joints.

Bridging the gap between passive-dynamic walkers and rigid-drive walkers are robots with variable impedance. These robots have compliant elements in their drive train, but they can be tuned to have a desired dynamic behaviour. Thus the ideal variable impedance biped would be able to have joints that could be rigid enough to at least absorb loading due to gravity while static, but could also have their impedance tuned to a suitable level to achieve efficient walking when required to move.

Tuned compliance may allow the joints of the robot to be moved according to their natural dynamics and gravity, potentially reducing the energy required to drive them, as in passive dynamic walkers. Compliance may also allow less energy to be lost during initial contact, especially when the robot is walking with a human like walk which will involve a heel strike rather than a carefully lowered foot. In addition to just from heel strike, compliance may allow a degree of energy storage, like in the plantar fascia of the foot, but also as seen at least in running humans.

During the stance phase of the walk, as the robot "vaults" over the leg on the ground, compliance may allow the joints to flex, allowing the vertical position of the pelvis to move passively and storing energy to put towards the final part of the stance phase. Compliance may also allow for disturbances in the terrain, or inaccuracies in the model of the robot or changes in the actual robot, to be adapted to more passively.

For a bipedal robot which will be carry out a number of tasks, adjusting the compliance allows the dynamics of the robot to be tailored to that tasks. For walking, the stiffness of the joints may need to be adapted based on the speed of the walk.

Efficient walking at different speeds should be possible by adjusting the impedances, and it may be the case that it is advantageous to vary impedances during the cycle of the walk. This is a relatively new field, but some robots aiming to produce variable impedance have been created.

One of the earliest attempts to build a robot with variable impedance used an antagonistic pair of non-linear springs on each joint.[105] However, the final robot was bulky and slow, and could only walk with a step speed of more than seven seconds per step, and a step length of 0.1m. A similar approach was employed in the robot Lucy[97], which used pleated pneumatic artificial muscles in an antagonistic arrangement. Lucy was only capable of walking with a speed of 0.15m/s, and a step length of 0.18m. Both of these robots failed to exhibit any real dynamic properties due to their extremely slow movement. A more recent bipedal robot which uses an anthropomorphic arrangement of pneumatic artificial muscles has achieved better results, and is able to both walk and jump.[39] However, the control of compliance in this robot is very rough, as is the overall control system. Another robot which has been constructed with antagonistic joints is MABEL, which is capable of walking at 1.5m/s and hopping.[30] The robot is very bulky and mechanically complex, and does not have any feet, merely two link legs. Nevertheless it does move at a reasonable speed with compliance.

Moving away from antagonistically actuated robots, two robots have been built which utilise the MACCEPA actuator, which is able to vary stiffness independently of equilibrium position.[93][63] The biped built by Uemura and Kawamura uses a mechanism which is very similar to the MACCEPA in its operation.[90] These mechanisms are very easy to construct from hobbyist parts, which no doubt explains why they were chosen, but can only realise very low stiffness settings, and cannot realistically transfer the kind of torque which is required for a proper dynamic walk. Furthermore the adjustment of stiffness with this device relied on changing the pretension of a spring, and is not energy efficient.

In recent years there has been an explosion in the number of designs for variable impedance joints, and so by careful selection of the best mechanism for the task it should be possible to create bipedal robots with physical variable impedance, suitable for the task of recreating human-like heel-toe walking.

### 1.6.1 *Methods for producing walking*

Many different methods have been devised for producing walking behaviour on bipedal robots.

Probably the most widely utilised technique is to utilise the Zero Moment Point (ZMP). "ZMP is defined as that point on the ground at which the net moment of the inertial forces and the gravity forces has no component along the horizontal axes."[101] In other words, it is the point in the robots support polygon (the area bounding the foot or feet currently on the ground) where the net forces from movement and from gravity produce no moment. For a robot that is dynamically balanced (i.e. not in the process of falling over), the ZMP coincides with the centre of pressure. If the ZMP is in the support polygon, the robot is dynamically balanced.

Setting a target ZMP trajectory and calculating the robot poses and joint trajectories required to produce it - all the while ensuring that the robot remains dynamically balanced - is a commonly used way to produce functional walking trajectories - e.g. [106][45][73]. These methods typically require an accurate model of the robot's dynamics, and produce walks which keep the feet of the robot flat on the floor.

Recently, optimization methods for generating trajectories based on special Centre Of Mass (CoM) trajectories have been used on the Atlas robot during the Darpa Robotics Challenge. [23]

Optimal control methods have also been used to optimise trajectories for locomotion in simulation, e.g. [21]

More reactive methods have been used, for example by Raibert [80][79], where feet are placed and ground reaction forces generated that generally act towards the centre of the robot, trying to push the robot towards a stable behaviour.

Hybrid Zero Dynamics control, which utilises models of the dynamics of the robot in each phase of motion, was used with high gain controllers to produce locomotion of the underactuated planar biped MABEL. [85]

The above methods all usually involve detailed models of the robots dynamics.

Neurologically inspired methods have also been used to generate walking trajectories, for example using reflexive control [27] and by attempting to combine reflexes with Central Pattern Generators.[52] See also [42] for a good review of CPGs in animals, and applied to robots.

## 1.7 GOALS

In this project, it was intended to study the state of the art in variable impedance - both stiffness and damping control - and create novel platforms to explore the efficacy of physical variable impedance in bipedal robots.

To best approach the amazing efficiency of human walking, the robots were to be made as outwardly anthropomorphic as possible, with regard to aiming to recreate the heel-toe walking cycle. However, this does not necessarily mean copying an antagonistic actuator style; it may be possible to improve upon this, for example allowing stiffness to be changed, or high stiffness levels held, without expending a great deal of energy.

To make the problem more tractable, we first concentrated on the sagittal plane, since the sagittal plane joints constitute the vast majority of the mechanical power in walking. We then started to introduce 3D motion through hip adduction/abduction, as a stepping stone on the way to full 3D movements.

With a robot designed morphologically for human-like walking, it was necessary to understand the human walking cycle in some detail in order to generate trajectories to produce a similar motion on a robot constrained in the sagittal plane. We developed a method for parameterising and representing joint angles during walking, based on gait transition events and a specified step length and clearance height. These motions were then empirically evaluated, in simulation and on the hardware, to ascertain which trajectories produced motion at which speeds. The stiffness was also varied, to attempt to determine the best stiffness at which to perform a given motion, at a given speed. In a similar way to how humans recruit from a bank of movement generation circuits, we learn a set of base motions for the robot, with associated stiffness values.

From this, we have begun to build a library of learned locomotive movements, which are described by step length, speed, clearance height, energy usage, and the like. A system for teleoperating the robot was developed, wherein motion tracking data from the users legs is analysed to determine whether they are making a discrete movement or producing locomotion. If the user is producing locomotion, the system attempts to reproduce the movement, without just copying joint angles, by recruiting a known locomotive trajectory from the movement database. In this way, features

like the step length and speed of the user's walk can be copied, but with joint angle trajectories and stiffness values that are more suited to the robot.

# Part II

# ROBOT DESIGN & MANUFACTURE

# 2

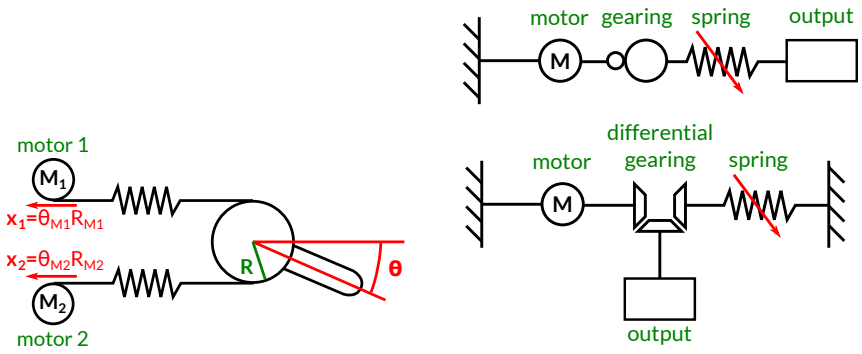
# VARIABLE IMPEDANCE TECHNOLOGY

The dynamic behaviour of a single actuated joint can be described in general form by the equation $T = I\ddot{\theta} + c\dot{\theta} + k\theta$, where $T$ is the external torque on the joint, $I$ is the second moment of inertia, $c$ is the damping coefficient, $k$ is the stiffness, and $\theta$ is the joint angle. The dynamics of the joint can therefore be altered by varying $k$, $c$ and/or $I$. Varying any or all of these parameters to change the dynamics of the joint is the focus of the field of variable impedance.

To build robots capable of varying their impedance, it is important to review the work already done in this field. This involves evaluating the many published designs for variable impedance mechanisms, and an overview of many of these designs is given in this chapter, before evaluating which designs are most suited to our application of legged robotic locomotion.

The field of variable impedance can trace its origins back to the late 1980s, although one of the first truly structured attempts to capitalise on the advantages brought by variable impedance is the work by Laurin-Kovitz around 1991[60]. Laurin-Kovitz and Colgate utilise the most common variable stiffness setup: antagonistic actuation with compliant linkages. The authors realise that in order to be able to vary stiffness in such an antagonistic design the springs used must have a non-linear characteristic, and in fact a quadratic force/displacement profile is advantageous. This fact has been elaborated in many subsequent works (e.g. [19]), and explains why many papers working with antagonistic variable impedance deal specifically with ways to make quadratic springs. Laurin-Kovitz and Colgate develop prototype designs for a 'programmable' damper and a non-linear spring.

Whilst not dealing with variable impedance, Pratt and Williamson's work on *Series Elastic Actuators* (*SEA*s) in the mid 1990s[75] is also an important milestone in development of intrinsically compliant actuators. The SEA class of actuators refers to a simple fixed stiffness compliant element in series with a traditional stiff actuator, but introduces the idea of using compliance to improve efficiency, shock tolerance and




(a) Antagonistic (b) Series

Figure 6: The two basic implementations of variable stiffness

provide energy storage. Replacing the fixed spring of a traditional SEA with a variable stiffness spring produces a variable impedance actuator. In certain configurations it is possible to adjust the stiffness of a series elastic joint by varying the pretension in the compliant element(s) which provides the restoration force to push the joint towards equilibrium[93][104]. A higher pretension generally means that when the joint moves out of equilibrium it is subject to greater forces, and thus a higher effective stiffness is observed.

There are two basic variable stiffness layouts - *antagonistic* or *series*. Series-type mechanisms require only one compliant element (although may have more for extra force/stability etc.) and can utilise either a tunable spring or a pretension mechanism in order to vary stiffness. It is most common in antagonistic layouts to use fixed non-linear springs with a pretension mechanism, however it is also possible to use tunable springs to vary the stiffness - possibly also adding a pretension mechanism.

Variable stiffness mechanisms can therefore be further categorised based on whether or not they are *pretension based*, i.e. whether they change their stiffness by changing the tension in a compliant element.

Stiffness is usually varied in tunable springs by altering mechanical variables such as the distance of springs from the rotational axis, the length of a compliant element, or otherwise changing the physical layout of the joint. A much less common method is based around using magnetic attraction/repulsion as the compliant element [11]. The strength of the magnetic fields can be varied if electromagnets are used; alterna-

tively it is possible to physically move the magnets to change the area of interaction - in this case the actuator would also fall into the mechanically-adjustable category.

So far only ways to vary the stiffness of a joint have been discussed. As previously mentioned, for any dynamic system the important parameters (apart from moment of inertia) are the stiffness and the damping - thus it is important to also consider ways of tuning the damping factor of a joint. Whilst there is considerably less literature on the topic of variable damping, there have nevertheless emerged several solutions for creating variable damping. These can be based on *Magneto-Rheological* fluid (MR-fluid) modulated with electromagnetic fields[36], variable frictional damping[68][57], changing the channel size in a hydraulic damper[60], or using motor braking as done at the University of Edinburgh [78].

In addition to varying the damping and/or stiffness of a joint, it is also possible to vary the moment of inertia of the links themselves in order to change the joint dynamics. In passive-dynamic walkers it is not uncommon to tune the mass distribution to achieve the desired dynamics[35]. We could vary this by attaching an additional concentrated mass on a linear slider and varying its distance from the axis of rotation, however this would involve the addition of mass to a system which we are trying to keep as lightweight as possible, and thus will not be discussed further in this section.

Over the years many different variable stiffness or variable damping designs have emerged[94] [98], and this chapter will attempt to provide a structured and detailed overview of the state of the art. Looking first at variable stiffness, basic design principles are introduced before reviewing the many published designs, and evaluating their suitability for legged robots. The MAwAS variable stiffness joint, used in both of our robots, is introduced. Potential mechanisms for variable damping are also reviewed and evaluated. Finally, we present the general variable impedance joint architecture which is used in our robots.

## 2.1 VARIABLE STIFFNESS PRINCIPLES

Variable stiffness mechanisms can be grouped into two orthogonal categorisations: pretension or tuned-spring; antagonistic or series.

ANTAGONISTIC Two or more compliant actuators working in opposition

SERIES A single compliant element in series with the output link

PRETENSION Stiffness is varied through adding tension to a compliant element

TUNED SPRING Stiffness is varied through a means different to pretension

It is of course possible to create an antagonistic design from the parallel connection of two series mechanisms, but generally this would produce a joint too bulky and complex to be useful outside of specific circumstances.

### 2.1.1 *Antagonistic Design*

From the diagram of a general antagonistic joint (Fig. 6a) it can be seen that the overall torque on the joint is given by:

$$T = F_k(x_1 + R\theta)R - F_k(x_2 - R\theta)R \tag{1}$$

where $F_k(x)$ is the force exerted by a spring, $\theta$ is the output position, and $R$ is the radius at which the moment from the springs acts. If standard linear springs are used for the joint, i.e. $F_k(x) = kx$, the torque equation becomes

$$T = kR(x_1 - x_2 + 2R\theta) \tag{2}$$

Giving the overall joint stiffness:

$$K = \frac{dT}{d\theta} = 2R^2k \tag{3}$$

It is therefore clear that if standard linear springs are used in an antagonistic arrangement the overall joint stiffness is fixed, regardless of the position of the inputs $x_1$ and $x_2$. If quadratic springs are used instead: $F_k(x) = ax^2 + bx + c$

$$T = R\Big(a(x_1 + R\theta)^2 + b(x_1 + R\theta) + c - a(x_2 - R\theta)^2 - b(x_2 - r\theta) - c\Big) \tag{4}$$

$$= R\Big(a\big(x_1^2 - x_2^2 + 2R\theta(x_1 + x_2)\big) + b(x_1 - x_2 + 2R\theta)\Big) \tag{5}$$

$$K = \frac{dT}{d\theta} = 2R^2\Big(a(x_1 + x_2) + b\Big) \tag{6}$$

Therefore the stiffness is now controllable and is proportional to the sum of the two inputs. It is independent of $\theta$ and therefore the joint presents itself as a standard

linear torsion spring. Note that if we can eliminate the $b$ coefficient from the spring equation the stiffness simply becomes

$$K = 2aR^2(x_1 + x_2) \tag{7}$$

Solving for the equilibrium position, this will occur when the external torque is zero:

$$0 = R\Big(a\big(x_1^2 - x_2^2 + 2R\theta_{eq}(x_1 + x_2)\big) + b(x_1 - x_2 + 2R\theta_{eq})\Big) \tag{8}$$

$$\theta_{eq} = -\frac{a(x_1^2 - x_2^2) + b(x_1 - x_2)}{2R\big(a(x_1 + x_2) + b\big)} \tag{9}$$

$$= -\frac{(x_1 - x_2)\big(a(x_1 + x_2) + b\big)}{2R\big(a(x_1 + x_2) + b\big)} \tag{10}$$

$$= -\frac{x_1 - x_2}{2R} \tag{11}$$

For an ideal antagonistic variable stiffness actuator with ideal quadratic springs, it is therefore possible to control equilibrium position and stiffness independently. By varying $(x_1 + x_2)$ whilst keeping $(x_1 - x_2)$ constant, the stiffness can be adjusted without changing the equilibrium position, and by varying $(x_1 + x_2)$ while keeping $(x_1 - x_2)$ constant the equilibrium position can be changed without affecting the stiffness setting. This is why it is desirable to achieve a quadratic spring function when dealing with antagonistic joints; any non-linear spring will produce an actuator which can vary its stiffness, but the control decoupling will not be as neat.

#### 2.1.1.1 *Energy Cost of Changing Stiffness*

As antagonistic mechanisms typically rely on increasing the pretension of springs in order to vary stiffness, energy must be expended in changing the stiffness. For a rudimentary calculation of this energy, we assume that a non-backdrivable mechanism such as a lead screw is used to adjust spring position, otherwise the drive motors will need to continually supply force to counter the pretension. When stiffness is increased the springs are pretensioned, and energy must be put into the spring and stored elastically. In theory this energy could be recovered when relaxing the joint, but in practice there is no simple way to implement such a mechanism. In fact, when relaxing a pretensioned spring the continual force being applied by the spring will actually increase friction on the leadscrew mechanism and cost energy.

For this analysis we examine the energy required to be supplied to the antagonistic springs in order to pretension them and increase stiffness - this gives a lower bound on the energy cost of the stiffness adjustment. This does not include any efficiency losses associated with the gearing and motor systems. Elastic potential energy is given by:

$$U_e = -\int F(x)dx \tag{12}$$

Where $F(x)$ is the force/displacement function of the spring. As previously discussed, traditional linear springs do not work, whilst quadratic springs give good stiffness range and controllability. For a quadratic spring with force equation $F(x) = -kx^2$:

$$U_e = \int kx^2 dx \tag{13}$$

$$= \frac{1}{3}kx^3 \tag{14}$$

As explained above, to vary stiffness it is necessary to adjust $(x_1 + x_2)$ whilst keeping $(x_1 - x_2)$ constant. Setting the equilibrium position to 0 and adjusting stiffness with $x_1 = x_2$; there are two springs, so the total energy consumption for an adjustment from $x_{s1}$ to $x_{s2}$ will be given by:

$$U_e = \frac{2}{3}k\left[x_{s2} - x_{s1}\right]^3 \tag{15}$$

From eqn (7):

$$k = \frac{K_2}{4R^2 x_{s2}} \tag{16}$$

Where $K_2$ is the desired stiffness in Nm/rad with $x_1 = x_2 = x_{s2}$.

For a simple example, let us posit that we are adjusting from a totally compliant state where $x_{s2} \approx 0$, and that in the final stiff state $x_{s2} = R$, this is a reasonable assumption for a high stiffness state in antagonistic mechanism:

$$U_e = \frac{2K_2}{12R^2 x_{s2}}\left[x_{s2}\right]^3 \tag{17}$$

$$\approx \frac{K_2}{6} \tag{18}$$

So, to move from a compliant state to a stiffness of 10Nm/deg (≈573 Nm/rad) in this simple example the minimum bound on energy usage would be $U_e = \frac{573}{6} = 95.5$J. Human walking at preferred speed has a net metabolic cost of around 3W/kg[91][22], meaning that this amount of energy is equivalent to a 62kg average weight human walking for half a second. Therefore, to use this amount of energy to change the stiffness of a single joint seems excessive. For a lightweight passive dynamic robot like the Cornell Ranger with a power consumption of 11.5W[4], this amount of energy would produce over 8 seconds of walking.

### 2.1.2 *Series Design*

The situation with series-type variable stiffness actuators is considerably simpler, with the stiffness and equilibrium position being almosts decoupled. It should be noted that with all series designs there are two possible ways to construct the drive-train: simple series connection or differential connection (see figure 6b). A differential connection, whilst requiring a mechanically more complex linkage, means that the tunable spring does not have to be moved to adjust equilibrium position, and lends itself naturally to implementation using a harmonic drive. Series connection is more popular due to its simplicity and the fact that harmonic drives are very costly, but some recent work has used harmonic drives as differentials [104] [44]

With the exception of pretension adjustment devices, there is no necessary energy input for stiffness adjustment save for the frictional losses and the cost of physically moving the required hardware.

## 2.2 VARIABLE STIFFNESS DESIGNS

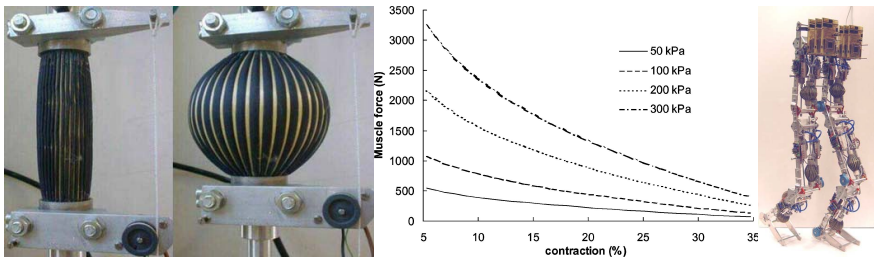
There are a great many methods for implementing variable stiffness, and these will be summarised over the following pages. Firstly methods for making non-linear springs for use in antagonistic layouts will be presented, before series mechanism.

### 2.2.1 *Methods for making non-linear springs*

Whilst traditional linear springs are readily available, simple mechanical devices, non-linear springs are more complex. Several methods have therefore arisen, with many aiming to create the quadratic force profile discussed earlier.

#### 2.2.1.1 *Pneumatic Artificial Muscles*

Pneumatic devices are often avoided in modern robotics because of both the added complexity of requiring a pressurised air source and the inherent hysteresis effects which make control difficult. A benefit of using such devices is that since the power is generated by the compressor off the robot, the actuators themselves can be more lightweight (although not necessarily more compact). Pneumatic artificial muscles are air filled bladders which contract when pressurised. Verrelst et al [99] use *Pleated Pneumatic Artificial Muscles* (*PPAM*)[14] which reduce the hysteresis and pressure threshold effects of normal air muscles to create a biped robot "*Lucy*" (fig 7c) which has three rotational joints per leg based on antagonistic arrangement of PPAMs. These air muscles display non-linear force/displacement characteristics which can be varied based on the pressure level inside the muscle as shown in figure 7b.



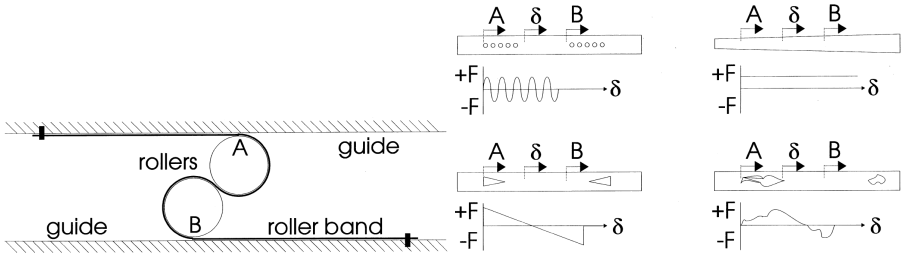

(a) PPAM (b) Force contraction curve for PPAM at various pressures (c) Biped Lucy

Figure 7: Pneumatic Pleated Artificial Muscles (PPAMs).[99]

The force/contraction curve is approximated by:[96]

$$F = pl_0^2 f\left(\epsilon, \frac{l_0}{R}\right) \tag{19}$$

$$f \approx pl_0^2(f_4\epsilon^3 + f_3\epsilon^2 + f_2\epsilon + f_1 + f_0\epsilon^{-1}) \tag{20}$$




(a) Basic Rolamite Layout

(b) Rolamite band shapes and corresponding force profiles

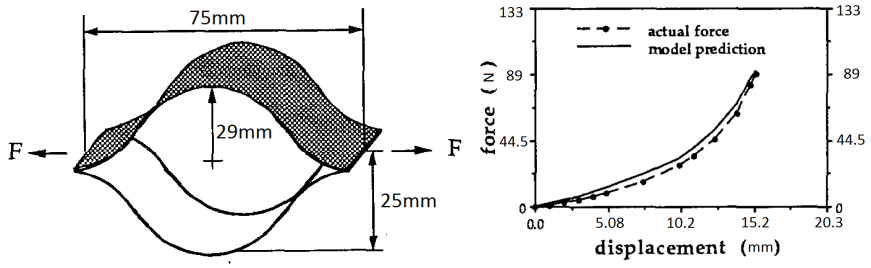
Figure 8: A Rolamite mechanism can be used to create a custom stiffness profile[20]

Where p is pressure, $l_0$ is the original length of the air muscle, and f is a dimensionless 'force function' which depends on contraction $\epsilon$ and muscle geometry ($l_0$ and unloaded radius R). f can be approximated by the function shown in eq (20), where $f_{0...4}$ are co-efficients.

As can be seen, using PPAMs creates the interesting situation where pretensing the non-linear springs actually means they contract in length, although this does not change the basic premise of antagonistic actuation. As shown in fig 7b, a single PPAM can generate a force of up to 3000N. The authors do not state the stiffness range achievable using the PPAMs, but are not able to reach their target stiffness of 42Nm/rad for walking as it is too low.[97] It is also telling that the maximum walking speed of Lucy is 0.15m/s, limited because "the exhaust valves cannot follow the imposed values above [this] speed". Also of vital importance for a walking robot is that the none of the joints in Lucy has a range of more than 60deg, which is another issue with using PPAMs in this setup. Work on Lucy was stopped in 2008.[97]

#### 2.2.1.2 *Rolamite Springs*

Never realised in practice, English and Russell proposed the use of a 'Rolamite' mechanism for creating a quadratic force/displacement profile.[20] The basic Rolamite design is shown in 8a and allows the two rollers to move along the traction with only rolling friction, giving the assembly a very low frictional coefficient. A roller band comprised of a springy material is connected to both the top and bottom tracks and is bent around the two rollers. As the band attempts to straighten out the result is axial forces at A and B. For a constant cross-section band these two forces are equal




(a) Non-linear spring (b) Spring force/displacement curve

Figure 9: The non-linear spring developed by Laurin-Kovitz and Colgate[60]

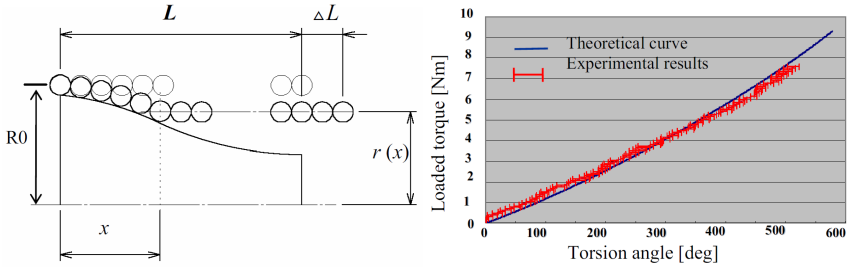
and opposite, so there is no net force; however if the width of the band is different at points A and B an overall force will result:

$$F \approx \frac{Et^3(w_B - w_A)}{6d^2} \tag{21}$$

Where $E$ is the Young's modulus of the band material, $t$ is the band thickness, $w_B$ and $w_A$ are the band width at B and A respectively, and $d$ is the roller diameter. By changing the width of the band (e.g. by creating cutouts) we can therefore exert a movement force on the rollers and create an arbitrary spring profile. Figure 8b shows a selection of Rolamite bands and their corresponding force profiles. English and Chad[20] show some possible band designs, but do not construct any. Designing to a set of geometric limitations so that a complete unit could (just) fit within a prosthetic forearm, the authors determine that a generous maximum output torque from an antagonistic joint based on these springs would be 2.3Nm - falling far short of the 16Nm they hoped to be able to achieve. Stiffness range in their design was 7-80 Nm/rad (0.122 - 1.40 Nm/deg).

#### 2.2.1.3 *Clamped Ellipsoid Spring*

In one of the earliest variable stiffness designs, Laurin-Kovitz and Colgate[60] realise that "material strength is higher in tension than in bending, and any shape which flattens out as it deflects should provide a hardening characteristic". The hardening characteristic they are referring to is the desired nonlinearity, where the stiffness increases with deflection.




(a) Schematic of guide shaft (b) Torque/angle curve

Figure 10: ANLES system developed by Koganezawa.[54]

Figure 9 shows a drawing of the prototype non-linear spring developed, as well as its characteristic force/deflection profile[1]. It should be noted that the spring is non-quadratic; the authors use a lumped parameter model to generate the theoretical force response shown in 9b.

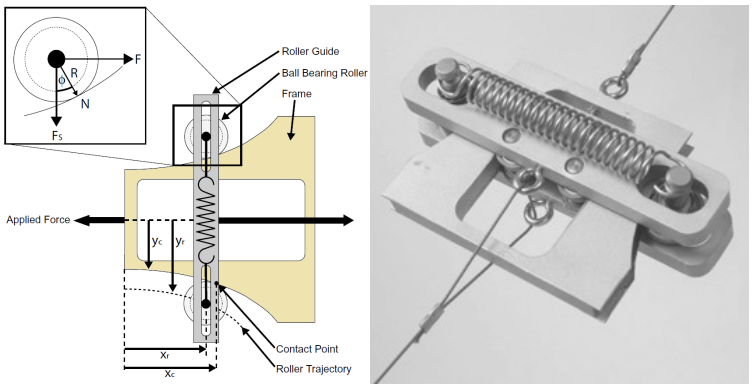
#### 2.2.1.4 *ANLES - Guided torsion spring*

As a torsion spring is wound up the number of coils increases, but as the wire length is constant this has the effect that the radius of the spring decreases. Koganezawa, Inaba and Nakazawa develop the *Actuator with Non-Linear Elastic System* which creates non-linear springs based around restricting the minimum spring radius. Figure 10a shows a profile of the guide shaft inside the torsion spring, and illustrates how as the torsion spring is wound it gradually becomes more restricted, with less of the spring being active. As the amount of active spring length decreases, the stiffness increases (in the same way that a short cantilever beam is stiffer than a long beam). Figure 10b shows the relationship between rotational displacement and torque for an ANLES unit, and we can see the desired hardening characteristic emerging. The authors construct an antagonistic joint based around two ANLES units, this bulky actuator is then shown to be capable of varying its effective stiffness between 0.09 - 0.14 Nm/deg.

#### 2.2.1.5 *Migliore's Quadratic Springs*

Migliore, Brown and DeWeerth developed a mechanism for creating quadratic springs based around guiding normal linear springs around a cam. The basic idea behind

1 Dimensions have been converted from imperial to metric




(a) Principle of spring operation (b) Realised device

Figure 11: Migliore's Quadratic Spring

the mechanism is quite straightforward, and is shown in 11b. As force is applied the rollers are pushed down the frame, and the linear springs are pushed apart by the profile of the frame. The authors design the profile of the frame to produce a quadratic output force, and produce two prototype units which "demonstrated strong quadratic behaviour ($r^2 = 0.9997$ and $r^2 = 0.9998$)". The force equations of the two constructed prototype units were experimentally determined to be:

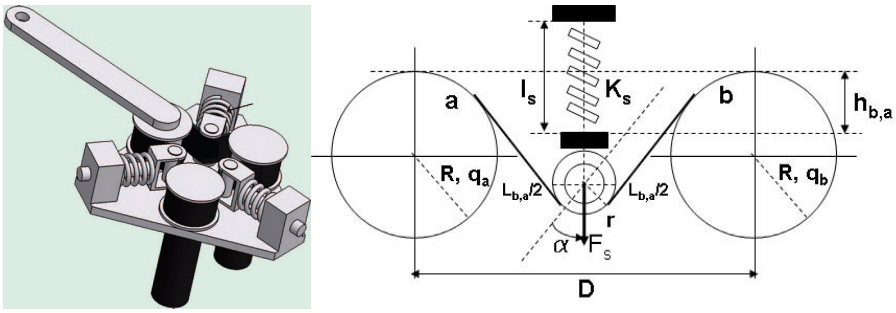
$$F_1(x) = 109700x^2 - 134x + 1.73N \tag{22}$$

$$F_2(x) = 108400x^2 - 282x + 1.94N \tag{23}$$

The differences in the two equations represent anomalies caused by manufacture, although the authors state that "effect on system performance is minimal". After creating an antagonistic joint based around their springs Migliore, Brown and DeWeerth are able to vary the effective stiffness between about $11 \times 10^{-6}$ - $807 \times 10^{-6}$ Nm/deg. This stiffness is quite low due to the fact that the linear springs used in the construction have low stiffness (550N/m); there is nothing to stop a more robust version of the joint being built with stiffer springs.

#### 2.2.1.6 *VSA-I - Belt tensioning spring*

Although more looking more complex in its implementation, the *Variable Stiffness Actuator* VSA-I developed by Tonietti, Schiavi and Bicchi[86] is essentially a traditional "pull-only" antagonistic design. The major difference in this case is the addition of a




(a) Layout of VSA-I (b) Non-linear spring

Figure 12: Tonietti's VSA-I[86]

third spring which counteracts the other two and helps to reduce energy usage. The layout of the VSA-I is shown in figure 12a, and can be seen to consist of a timing belt stretched around three pulleys, two of which are driven by motors and one of which is attached to the output joint. In the space between two pulleys the timing belt is tensioned by a springs; a schematic of one of these linkages is shown in figure 12b. If we imagine that the pulley on the right is locked and cannot rotate, then as the other pulley rotates anticlockwise it will pull the timing belt tighter and decrease the spring length $l_s$. This arrangement creates a non-linear relationship between the relative displacement of the two pulleys and the exerted force, calculated by the authors to be:

$$F = 2K_s \frac{\bar{h} - h}{h} L \tag{24}$$

$$L = \hat{L} + R(q_b - q_a) \tag{25}$$

$$h = \frac{\sqrt{L^2 - D^2}}{2} \tag{26}$$

Where $\bar{h}$ is the length of the spring at rest, $h$ is the length of the spring, $L$ is the length of the timing belt between the two pulleys, $q_a$ and $q_b$ are the rotations of the two pulleys and the other variables are dimensions as shown in figure 12b. The force function described by equation 24 is non-linear and monotonically increasing with $(q_b - q_a)$, tending to infinity as $L \to D$. It is shown in figure 13, and can be approximated well by a fourth order polynomial (shown as the thin yellow line). The authors construct a prototype VSA-I actuator, but have since abandoned the design

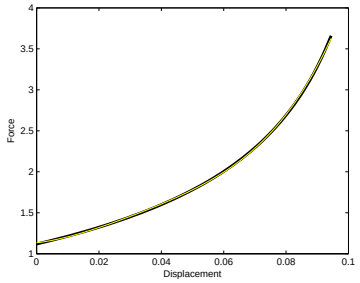
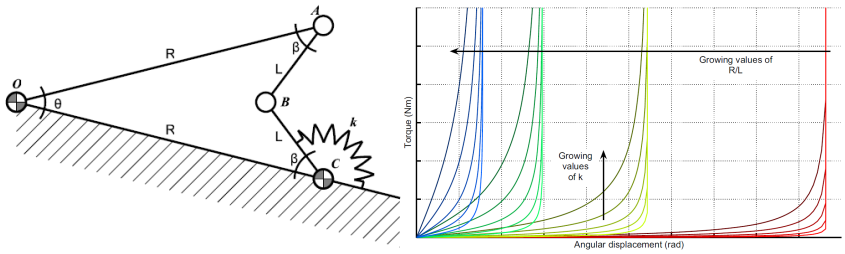



Figure 13: Characteristic of the non-linear spring used in the VSA-I




(a) Four bar linkage (b) Torque vs. displacement

Figure 14: VSA-II four bar linkage mechanism[82]

since "it finds limitation in torque capacity and in practical implementation in a robotic arm"[82].

#### 2.2.1.7 *VSA-II - Four bar linkage*

The successor to the VSA-I was developed by Schiavi, Grioli, Sen and Bicchi in 2008 [82] and uses a non-linear spring which is capable of pushing and pulling. The spring itself is constructed as a torsion spring formed from a four bar linkage as shown in 14a. OA is the output link, and a torsion spring is connected to link BC. The relationship of torque to displacement is complex, and the full equation is presented by the authors in [82] - a graph of this relationship is shown in figure 14b for various stiffness presets and values of the design variable R/L. It can be seen that the behaviour is again non-linear, monotonically increasing and has a hardening characteristic.

The VSA-II actuator constructed by the authors uses relatively weak springs with k=0.5Nm/rad, and this results in a final output stiffness range of $0 \rightarrow 0.05$ Nm/° ( $0 \rightarrow 3$Nm/rad). It should be noted though that stiffness depends heavily on displacement from equilibrium, and as the deflection increases the stiffness also increases

rapidly. Many variable stiffness actuators exhibit a hardening or softening characteristic, but in the VSA-II has an extremely pronounced hardening stiffness profile. Whilst a hardening characteristic can help to prevent the actuator from exhausting its elastic range, for the actuator to be most useful the majority of the compliant range should usually be at a useful (i.e. not overly soft or hard for the application) level.

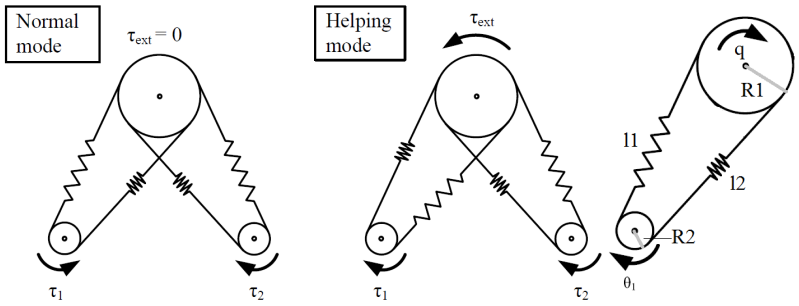
### 2.2.2 *Other antagonistic designs*

Several other designs exist which utilise opposing elastic elements, but not in the traditional antagonistic layout.

#### 2.2.2.1 *Bidirectional Antagonistic*

Petit et al[74] noted that in many implementations of the traditional antagonistic setup the implemented springs are only capable of pulling, and so can only provide force to the output joint in one direction. Such "pull-only" antagonistic actuators suffer from the major drawback that the maximum torque that can be applied to the output link is the stall torque of one of the motors - i.e. only half of the possible motor torque can be delivered to the output. Note that where the non-linear springs can push and pull (e.g. Rolamite springs[20], VSA-II[82]) this problem is not encountered since one motor can push and the other can pull. To overcome the problem when using pull-only springs the authors propose the configuration shown in figure 15a, where each motor is coupled to the output link by a pair of springs and can thus pull the output pulley in either direction.

Considering one motor-spring unit, as shown in 15b we can calculate the torque exerted on the output pulley by that unit.




(a) Joint configuration (b) Motor-spring unit

Figure 15: The bi-directional antagonistic setup proposed by Petit et al. In Normal Mode the joint operates almost as a traditional pull-only antagonistic actuator, with pretension increasing the output joint stiffness. In Helping Mode it is possible to use the torque of both motors to drive the output joint.

$$l_1 = l_{eq} + R_1 q - R_2 \theta_1 \tag{27}$$

$$l_{eq} = l_f + l_p \tag{28}$$

$$F_1 = F_k(l_1 - l_f) \tag{29}$$

$$= F_k(l_p + R_1 q - R_2 \theta_1) \tag{30}$$

$$T = R_1(F_1 - F_2) \tag{31}$$

$$= R_1\left(F_k(l_p + R_1 q - R_2 \theta_1) - F_k(l_p - R_1 q + R_2 \theta_1)\right) \tag{32}$$

Since the stiffness of the motor unit is the derivative of $T$ with respect to $q$, and we want to be able to vary the stiffness by changing $\theta_1$, it is necessary to obtain a term which is a product of both $q$ and $\theta_1$ in the torque equation. Thus $F_k(l_p + R_1 q - R_2 \theta_1) - F_k(l_p - R_1 q + R_2 \theta_1)$ must contain a $q\theta_1$ term. If we are dealing with springs described by a polynomial we can explore what happens as we increase the order of the polynomial:

$$(l_p + R_1 q - R_2\theta_1)^1 - (l_p - R_1 q + R_2\theta_1)^1 = 2(R_1 q - R_2\theta_1) \tag{33}$$

$$(l_p + R_1 q - R_2\theta_1)^2 - (l_p - R_1 q + R_2\theta_1)^2 = 4l_p(R_1 q - R_2\theta_1) \tag{34}$$

$$(l_p + R_1 q - R_2\theta_1)^3 - (l_p - R_1 q + R_2\theta_1)^3 = \ldots 6(R_1 q R_2^2\theta_1^2 - R_1^2 q^2 R_2\theta_1)\ldots \tag{35}$$

$$(l_p + R_1 q - R_2\theta_1)^4 - (l_p - R_1 q + R_2\theta_1)^4 = \ldots 24l_p(R_1 q R_2^2\theta_1^2 - R_1^2 q^2 R_2\theta)\ldots \tag{36}$$

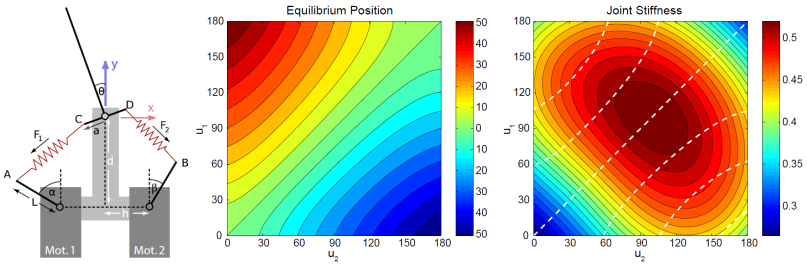
Therefore to gain any control over the stiffness of the joint, if we are describing the springs as polynomials, they must be of order 3 or greater. The torques from each motor unit can be summed to find the resultant output torque, and since $K = dT/dq$, the individual motor unit stiffnesses may also be summed to calculate the overall joint stiffness. In their prototype implementation, Petit et al use a "belt tensioning spring" method similar to that used in the VSA-I (see section 2.2.1.6). We have already seen that the $F_x$ of this spring can be described by a fourth order polynomial, and therefore the resulting joint which uses four of these will be able to change its stiffness.

The resulting stiffness changes with deflection angle, becoming stiffer as deflection increases. The prototype joint created by the authors is capable of varying its stiffness between 0.57→1.05Nm/°.

#### 2.2.2.2 *Edinburgh SEA - Lever arms with linear springs*

Mitrovic, Klanke, Howard and Vijayakumar produced the Edinburgh SEA[65] which uses normal linear springs attached to lever arms to create the required non-linearity. A schematic of the design is shown in figure 16a, along with graphs showing how the equilibrium position and stiffness change with the positions of the two motors in figure 16b.

As with all antagonistic designs, the pretension can be varied to change the joint stiffness whilst changing the differential angular displacement of the input motors can change the equilibrium position. There is a high amount of coupling between position and stiffness, and the joint stiffness depends on deflection from equilibrium in a non-trivial way. The prototype actuator built by the authors with fairly weak springs (k=424 N/m) is capable of varying the output link stiffness between about 0.3→0.5Nm/° as shown in figure 16b. We note that stiffness is directly proportional to the linear spring constant k.




(a) Schematic (b) Equilibrium position and stiffness profiles

Figure 16: The Edinburgh SEA and its characteristic curves[65]

### 2.2.3 *Series Pretension Adjustment Devices*

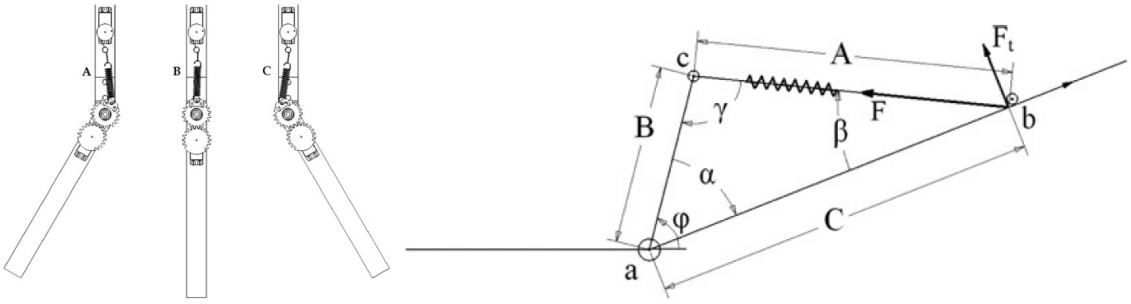
As mentioned earlier, series VSAs can be either pretension or non-pretension based, depending on whether or not they depend on changing the tension of an elastic element in order to change stiffness. In this section, pretension based mechanisms are introduced first, followed by non-pretension mechanisms.

#### 2.2.3.1 *MACCEPA - Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator*

The MACCEPA was developed by van Ham, Vanderborght, Van Dammr, Verrelst and Lefeber[93] and represents perhaps the first series variable stiffness actuator that varies the output link stiffness by spring pretension. The basic premise for the actuator is that an intermediate lever arm sets the equilibrium position, attached to the end of this arm is a spring which connects to the output link at a distance, C, from the axis. The spring will be at its shortest when the output link is aligned with the intermediate link, and this represents the equilibrium position. Any deviation from this will further stretch the spring and change the angle of spring force so it is no longer parallel with the output link, and therefore a turning moment is introduced on the output link. A drawing of a MACCEPA implementation in figure 17a shows the joint at the equilibrium position in B, and out of equilibrium on either side. The schematic in figure 17b can be used to determine the characteristics of the device.

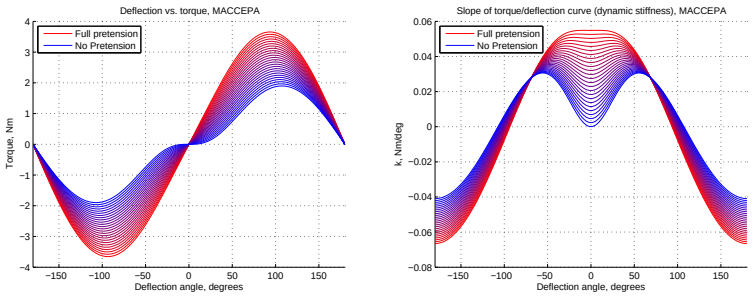
The torque on the output link is given by

$$T = kBC \sin\alpha \left(1 + \frac{P - |C - B|}{\sqrt{B^2 + C^2 - 2BC\cos\alpha}}\right) \tag{37}$$




(a) 3 positions of the MACCEPA (b) schematic of operation

Figure 17: The MACCEPA[93]




(a) Torque vs. Displacement (b) Stiffness vs. Displacement

Figure 18: MACCEPA characteristics

Where k is the stiffness constant of the linear spring, B and C are the distances from the rotation axis to the spring connection points on the intermediate and output links respectively, P is the pretension introduced into the spring, and $\alpha$ is the angle between the intermediate link and output link. The length A is given by $A(\alpha) = \sqrt{B^2 + C^2 - 2BC\cos\alpha}$. We can note that for deflections from equilibrium ($\alpha$) of less than 45° the majority of the spring force results in internal tension rather than producing an output moment. Differentiating the torque equation gives us the equation for stiffness:

$$K = kBC\left(\left(\frac{P - |C - B|}{A(\alpha)} + 1\right)\cos\alpha + \frac{|C - B| - P}{A^3(\alpha)}BC\sin^2\alpha\right) \tag{38}$$

The characteristic curves for the MACCEPA are shown in figure 18, and the highly non-linear relationship between stiffness and deflection from equilibrium can be seen in figure 18b. For low stiffness settings the MACCEPA therefore exhibits a hardening

characteristic up to about 50°, when it begins to soften. At high stiffness settings a softening behaviour is seen, with the joint becoming less stiff as it moves away from equilibrium. Note that the stiffness goes negative at high deflection angles, indicating that the restoration torque is actually decreasing as deflection increases towards 180°. In practice the allowable deflection range may well be limited by the elastic limit of the spring. The stiffness at equilibrium is given by:

$$K_{eq} = \frac{kPBC}{|C - B|} \tag{39}$$

The energy required to move from a compliant state with $P = 0$ to a state with stiffness K and pretension P is given by the normal spring energy equation $E = \frac{1}{2}kP^2$, therefore:

$$E = \frac{K^2(C - B)^2}{2kB^2C^2} \tag{40}$$

For an example system let C=0.1m and B=0.03m. With a relatively weak spring of k=980N/m, free length 28.5mm and maximum length 51.7mm the achievable stiffness range at equilibrium is 0→0.0170Nm/°, with an energy requirement of 0.264J to achieve this stiffness. With a strong extension spring of k=17090N/m, free length 31.75 and maximum length 39.624 the achievable stiffness range at equilibrium is 0→0.101Nm/°, with an energy cost of 0.530J to reach maximum stiffness. The MACCEPA is therefore inherently a low-stiffness joint.

Interestingly, Van Ham has constructed a bipedal robot, Veronica[93], by connecting six MACCEPA actuators together. Whilst the robot could walk slowly, "it should be noticed that this walking motion is not yet stable, and requires human assistance". Work was not continued on the Veronica biped, most likely since the MACCEPA based joints are too soft to provide stable locomotion. The MACCEPA joint forms the basis of work on efficient throwing with variable dynamics using a two link arm at the University of Edinburgh. [6]

#### 2.2.3.2 *MACCEPA 2.0*

Vanderborght, Tsagarakis, Semini, Van Ham and Caldwell propose an update to the basic MACCEPA design which creates a hardening behaviour where stiffness

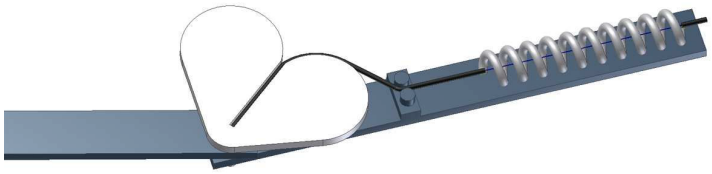
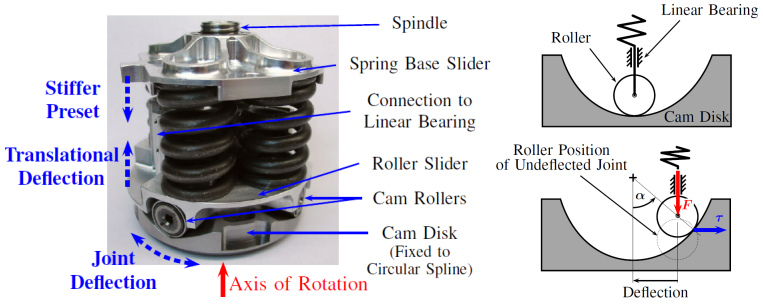


Figure 19: The MACCEPA 2.0 deflected from equilibrium




(a) Implementation (b) Operational principle

Figure 20: The DLR VS-joint

increases with deflection from equilibrium.[95] Figure 19 shows a CAD render of the mechanics behind the MACCEPA 2.0 - as the output link deviates from equilibrium the wire connected to the spring is pulled around a circular profile. This has the effect that the rate of spring deflection increases with deflection, and thus the hardening characteristic is observed. Apart from this the operation is identical to the original MACCEPA, and the stiffness range is not increased with this implementation.

#### 2.2.3.3 *DLR VS-Joint*

A tunable spring which alters its stiffness through a pretension mechanism is the DLR VS-joint, created by Wolf and Hirzinger[104]. Figure 20a shows the implementation of the VS-joint, with three large compression springs arranged symmetrically around and parallel with the rotational axis. A spindle mechanism allows the top plate (the "spring base slider") to be moved and the springs compressed to introduce pre-tension. The other end of each spring is attached to a roller which then sits on a a profiled cam-disk, as illustrated in figure 20b. Any deflection from equilibrium will compress the springs further and introduce a restorative moment forcing the output link back to equilibrium.

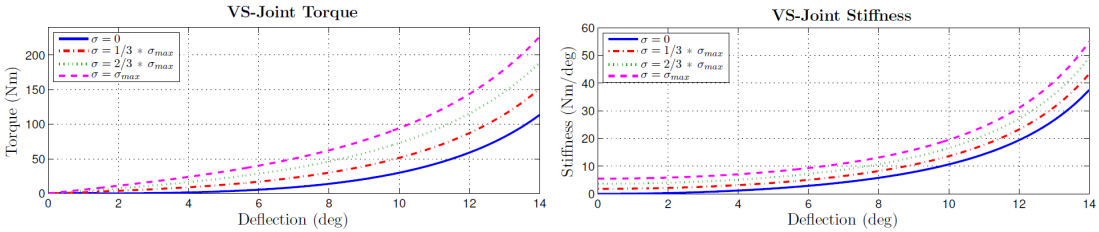



Figure 21: VS-joint characteristics[104]

Though there are three springs, they are in series and so their spring constants can be added and they can be considered as one unit. It is possible to adjust the shape of the torque/deflection profile by altering the shape of the cam disk - and it should be noted that by introducing a vertical face at the end of the slope of the profile the link can be made to become rigid once it has reached maximum deflection.

The torque/deflection and torque/stiffness curves for the VS-joint created by Wolf and Hirzinger are shown in figure 21, where the overall stiffness is $k = 980 \times 10^3 N/m$.

For the constructed VS-joint the stiffness range at equilibrium is $0 \rightarrow 5.93 Nm/°$. Note though that the stiffness depends on the profile of the cam, and the circular cam profile used here has a small slope around the equilibrium position. Compression springs can generally be more forceful than extension springs, and three relatively strong springs are used in this implementation. At the highest pretension setting the spring deflection is $\frac{630}{180} = 3.5mm$ and the corresponding spring force is 3430N. The energy required to reach this pretension is $E = \frac{1}{2}kx^2 = 6J$.

#### 2.2.3.4 *Uemura's design*

Uemura and Kawamura propose a design for a variable stiffness joint in which a high pretension creates a low stiffness, and vice versa. This design has the characteristic that as stiffness increases so does the energy storage potential, unlike the series pretension and antagonistic designs discussed previously. The principle behind the joint is illustrated by the drawing in figure 22, where we can see that the design allows us to change the fixation distance of the spring from the rotational axis (r).

From the geometry we can calculate the equilibrium restoration torque, and therefore the joint stiffness:

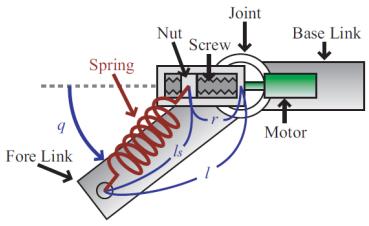
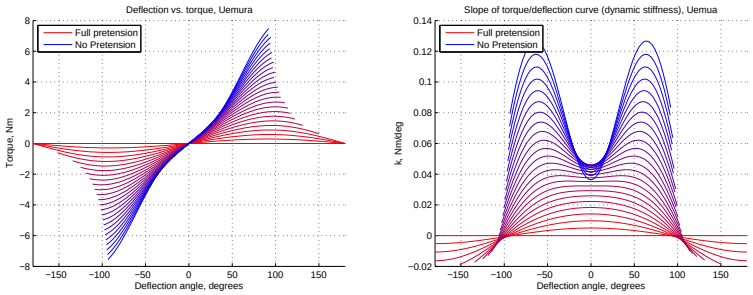



Figure 22: Uemura's variable stiffness design[90]




(a) Torque/deflection (b) Stiffness/deflection

Figure 23: Characteristic curves for Uemura and Kawamura's design

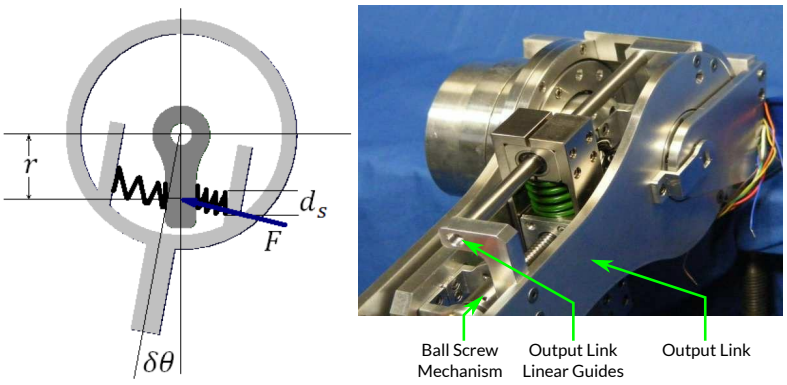
$$l_s = \sqrt{l^2 + r^2 - 2lr\cos q} \tag{41}$$

$$T = lkr\left(1 - \frac{l_f}{l_s}\right)\sin q \tag{42}$$

$$K = \frac{dT}{dq} = lkr\left[\cos q - \frac{l_f \cos q}{l_s} + \frac{l_s lr \sin^2 q}{l_s^3}\right] \tag{43}$$

In the prototype implementation by Uemura and Kawamura a strong extension spring with k=4570N/m is used, with a long link arm of l=0.19m. The characteristic curves for the joint created are shown in figure 23, where we can note from figure 23b that there is not a monotonically increasing relationship between r and joint stiffness at equilibrium. These curves are limited by the maximum permissible deflection of the spring.

The achievable stiffness range at equilibrium, with this strong spring, is $0 \rightarrow 0.046 Nm/^\circ$. In this design energy must be added to the system to move from stiff to compliant, and to travel from $0.046 \rightarrow 0 Nm/^\circ$ the energy cost is $E = \frac{1}{2}kx^2 = 1.32J$.




(a) Schematic (b) Implementation

Figure 24: The *Actuator with Adjustable Stiffness* (AwAS)[44]

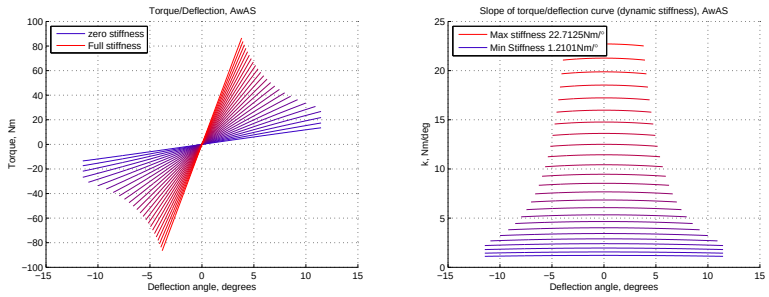
### 2.2.4 *Tunable Springs*

Series mechanisms which are not pretension based are effectively tunable springs, and can remain at a stiffness setting without expending energy.

#### 2.2.4.1 *AwAS - Distance of springs from rotation axis*

The *Actuator with Adjustable Stiffness* (*AwAS*) was originally developed by Jafari, Tsagarakis, Vanderborght and Caldwell in 2010[44]. The basic principle of operation is centred on a pair of linear springs attached to lever arms on the rotating joint. By changing the distance of the springs from the rotation axis the effective stiffness can be changed. Figure 24a shows a schematic representation of the principle behind the AwAS, where the distance of the springs from the rotational axis is shown as $r$. The coil diameter of the spring is given by $d_s$ and the deflection from equilibrium by $\delta\theta$.

We can observe that the spring is not compressed equally across its diameter (i.e. the two spring mounting plates are not parallel), and thus we cannot consider the spring to be acting as a point force at the distance $r$, but must instead integrate the force between $r - \frac{d_s}{2} \rightarrow r + \frac{d_s}{2}$. The resultant torque and force equations for the output link are given by:




(a) Torque vs. deflection (b) Stiffness vs. deflection

Figure 25: Characteristic curves for the AwAS

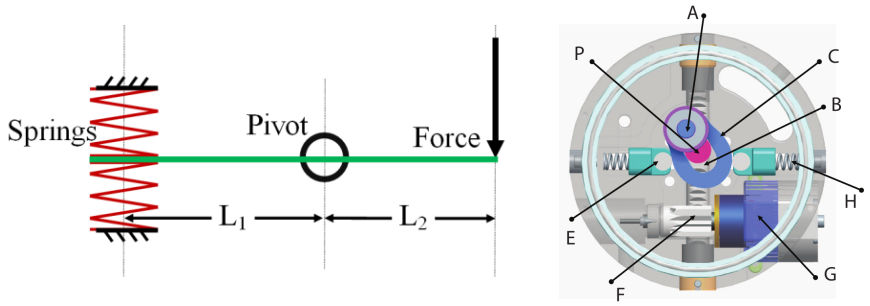
$$T = K_s\left(r^2 + \frac{d_s^2}{12}\right)\sin(2\delta\theta) \tag{44}$$

$$K = 2K_s\left(r^2 + \frac{d_s^2}{12}\right)\cos(2\delta\theta) \tag{45}$$

Maximum deflection in the AwAS is limited at high stiffness ranges by the maximum compression of the springs, but at lower stiffnesses by the intermediate link colliding with a mechanical limit ("A mechanical lock constrains the angular deflection between -0.2 and 0.2 rad"[44]) to stop it hitting the linear guides of the output link. This can be seen in Fig. 24b, which shows how the linear guides of the output link and the ball screw of the intermediate link move in the same plane, limiting the relative angular displacement before the ball screw mechanism contacts the linear guides of the output link. In both cases the joint becomes rigid once the maximum elastic deflection is exceeded (i.e. no damage is done to the springs).

The characteristic displacement/torque and displacement/stiffness curves for the AwAS as implemented by Jafari[44] are shown in figure 25. It is important to note that, at equilibrium, changing the stiffness of the joint does not involve any pretensioning of springs, and the only energy required to change stiffness is that required for the mechanical effort of physically moving the position of the springs. Under load, energy will still be required to increase the stiffness if the output position is to be kept constant.

The Hybrid Dual Actuator Unit (HDAU) by Kim and Song [51] is another mechanism which uses the same principle of moving compression springs orthogonally to




(a) Moving pivot point principle of AwAS-II [43]

(b) Implementation of CompAct-VSA[87]

Figure 26: The AwAS-II and CompAct-VSA operate by moving the pivot point of a lever arm. In (b), the pivot P is moved by the rack and pinion F, changing the amount by which the cam C deflects the springs H. Drive is transferred through joint A at the opposite end of the cam.

the rotation axis. This mechanism uses rack and pinions to move the spring assemblies, rather than leadscrews.

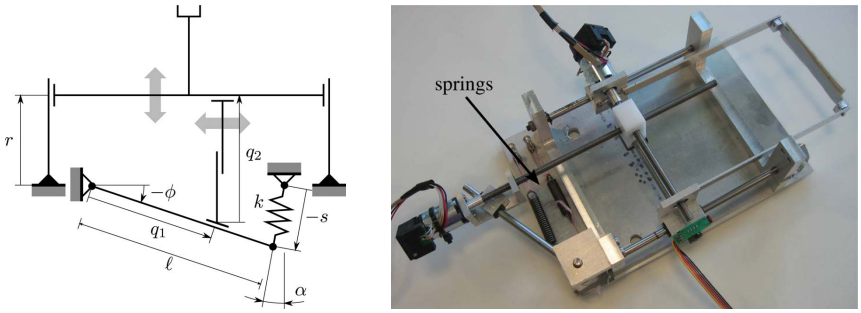
#### 2.2.4.2 *AwAS-II and Compact VSA - Moving pivot point*

Jafari, Tsagarakis and Caldwell take a slightly different approach with the AwAS-II, which moves the pivot point of a level arm in order to achieve a stiffness range which, in theory, can vary from zero to completely rigid. [43]. Fig. 26a shows the operational principle, where the pivot point is moved to change the stiffness. When the pivot is between the springs, the end of the lever can move freely without deflecting the springs. When the pivot point is at the end of the lever, the applied force is directly transferred, i.e. the joint is rigid. Pivot positions between these two extremes produce a spectrum of stiffness values.

A similar principle is used by Tsagarakis, Sardelitti and Caldwell in the CompAct-VSA [87]. The implementation is slightly different, and utilises a cam and compression springs rather than torsion springs, as shown in Fig. 26b.

#### 2.2.4.3 *vsaUT - moving the force along a lever*

The vsaUT[100] from the University of Twente takes yet a third approach to the problem. As shown in the schematic in Fig. 27a, one end of a lever is attached to a




(a) Mechanical Schematic. [100] (b) Prototype implementation [100]

Figure 27: The vsaUT[100] uses the principle of moving the application point of the joint force along a lever from a pivot to a pair of antaogonistic springs

pivot and the other to a spring. The joint force is transferred from the spring through the lever to the arm marked as $q_2$, which connects to the output link (note that the schematic here shows a linear actuator). By changing distance $q_1$, the arm is moved along the lever, and the mechanical advantage between input and output is changed. Therefore, when the arm is close to the pivot point, a small deflection of the output will create a large change in spring length; when the arm is close to the spring, a deflection of the output will create nearly the same deflection of the spring.

In this way the stiffness of the joint can be varied from the minimum value set by the spring, to theoretically infinite stiffness. In the prototype implementation of the joint, which can be seen in Fig. 27b, the authors utilise a pair of extension springs, antagonistically arranged, to provide bidirectional compliance.

#### 2.2.4.4 *MIA - variable length compliant element*

Based on the MIA developed by Morita [67], a simplified MIA joint as shown in figure 28 can be considered to be a cantilever beam in under loading. For relatively small deflections we can use the beam bending equation to relate the applied force to deflection

$$F = \frac{3rEI}{l^3} \tan\theta \tag{46}$$

$$T = \frac{3r^2EI}{l^3} \tan\theta \tag{47}$$

$$K = \frac{dT}{d\theta} = \frac{3r^2EI}{l^3} \sec\theta \tag{48}$$

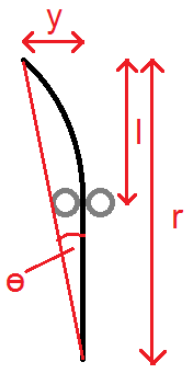



Figure 28: The simplified MIA

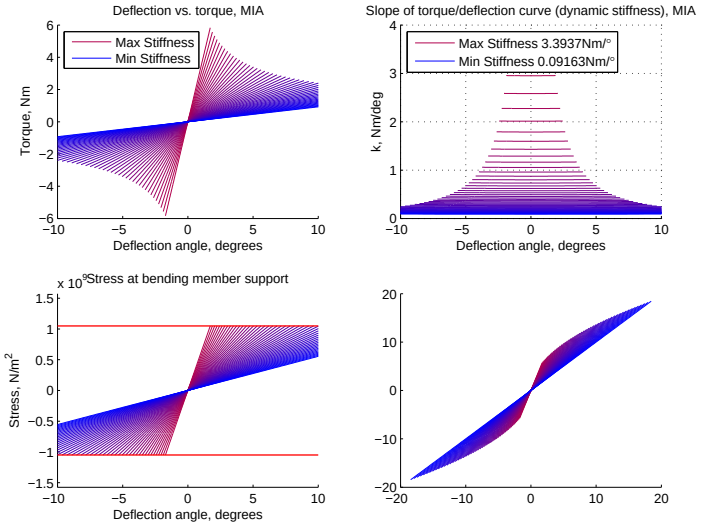



Figure 29: Characteristic curves of the simplified MIA

Where r is the distance from the axis of rotation to the applied force, E is the Young's modulus of the compliant element, I is the second moment of area of the compliant element, θ is the deflection from equilibrium and l is the active length of the compliant element. The maximum deflection angle is limited in this case by the stress, σ, of the bending member at support

$$\sigma = \frac{Mb}{2I} \tag{49}$$

$$M = \frac{T}{r}l \tag{50}$$

$$\sigma = \frac{3rEb}{2l^2}\tan\theta \tag{51}$$

$$\theta_{max} = \tan^{-1}\left(\frac{2\sigma_{yield}l^2}{3rEb}\right) \tag{52}$$

Where $M$ is the bending moment on the beam, calculated as above, $b$ is the thickness of the compliant element and $\sigma_{yield}$ is the yield stress of the compliant element. Note that these formula calculate the stress at the upper/lower surfaces of the beam, which is the maximum stress in the element. The calculated torque and stiffness are shown in figure 29 for one possible compliant element.

Choi et al. [12] use a similar principle in their work, except their mechanism uses four compliant elements sharing the torque transmission.

#### 2.2.4.5 *Other non-pretension based mechanisms*

The mechanism of Choi et al. [11] transfers drive through two concentric rings of alternating magnets. In equilibrium the joint naturally aligns so that the magnetic poles are attracting (i.e. in north-south pairs). As the joint is moved out of equilibrium a restoration torque pushes it back. By moving one of the magnetic rings in the axial direction, the "mesh" between the magnets can be altered, and the strength of the restoration torque varied, thereby changing the joint stiffness.

Umedachi and Ishiguro[92] created a spring mechanism which rotates a heterogeneous outer band around linear springs in order to change the overall stiffness.

Rotating a beam to change its bending properties is another mechanism which can be used to change physical compliance, as demonstrated by Seki et al. [84].

### 2.2.5 *Variable Stiffness Summary Table*

Table 1 shows a summary of a selection of variable stiffness mechanisms. Since most antagonistic designs concentrate on creating non-linear springs, these designs are grouped under *pull only*, *pull-push*, and *bidirectional* antagonistic categories. The *pull*

*only* antagonistic category refers to the traditional antagonistic layout where two opposing compliant elements can act in one direction each only.

In some designs, the spring force does not act purely as a turning moment on the joint, and therefore even if strong springs are used, the overall stiffness of the joint can be low. Whilst the specific stiffness levels achievable will vary based on the design dimensions and spring factors, a rough categorisation of the possible stiffnesses of each design is included in the table. Of note is that the design used by the AwAS II [43] and CompACT VSA [87] is capable of varying from zero stiffness (i.e. free spinning) and infinite stiffness (i.e. rigid joint).

The table also shows how the potential for energy storage changes as stiffness increases. $T_{max}$ vs $k$ shows the how the maximum torque provided by the mechanism changes as stiffness increases; in antagonistic designs, the maximum output torque decreases as stiffness increases, whereas in most other designs the opposite occurs. The energy cost of changing stiffness in antagonistic designs is typically higher than for other designs, as it involves pre-stressing a plurality of springs.

| VSA | Type | Nominal Stiffness | Energy Storage vs stiffness | Max elastic deformation range | Stiffness linearity | $T_{max}$ vs k | Complexity of manufacture | Energy cost to change stiffness |
|---|---|---|---|---|---|---|---|---|
| "pull only" antagonistic | A | L → H | ↓ | ●●● | if quadratic springs | ↓ | simple | ●● |
| "pull-push" antagonistic | A | 0 → H | ↓ | ●●● | if quadratic springs | ↓ | simple | ●● |
| bidirectional antagonistic | A | L → M | ↓ | ●●● | hardening | ↓ | simple | ●● |
| Edinburgh SEA [65] | A | L | ↓ | ●●● | highly non-linear | ↓ | very simple | ● |
| MACCEPA [93] | SP | L | ↓ | ●●● | highly non-linear | ↑ | simple | ● |
| Uemura [90] | SP | L | ↑ | ●●● | highly non-linear | ↑ | simple | ● |
| VS-joint [104] | SP | 0 → H | ↓ | ● | customisable | ↑ | complex | ● |
| AwAS [44] | S | L → H | - | ● | near linear | ↑ | average | - |
| AwAS II [43], CompACT [87] | S | 0 → ∞ | - | ●● | near linear | ↑ | complex | - |
| MIA [66] | S | L → H | ↓ | ● | near linear | ↑ | simple | - |
| Choi2011 [12] | S | M → H | ↓ | ● | near linear | ↑ | complex | - |

Table 1: Variable Stiffness Mechanisms.

A = Antagonistic, S = Series, P = Pretension, L = Low, M = Medium, H = High

## 2.3 VARIABLE DAMPING

Being able to adjust the damping of a joint allows unwanted oscillations to be mitigated, and allows energy to be dissipated in a fashion other than through the drive motor.

### *Magnetorheological damping*

Often used in prosthetic knees[36], this method uses MagnetoRheological (MR) fluid to fill the gaps between many thin disks, which are alternately connected to the input and output links. An electromagnet is used to set up a magnetic field which passes through the discs and the fluid. As the strength of this field increases the MR fluid begins to resist the relative movement of the discs, generating damping torques. This method has been used to alongside SEA in a walking robot [25].

### *Frictional Damping*

One easy way to dissipate energy from a joint is to apply braking through frictional contact. In order to achieve true damping it is necessary to make the applied force proportional to rotational velocity, and thus if traditional brakes are being used it is necessary to use active control to modulate the braking force. This can be with torque control of a disk brake[66], or PWM control of an electromagnetic brake[67].

Laffranchi et al use piezoelectric stacks to push a brake against the joint in the "CompAct" mechanism[56].

### *Motor Braking*

If a motor is being backdriven by a joint, a component of the resistance torque it provides will be proportional to the velocity. If the terminals of the motor are shorted together this damping torque will be maximised, and if PWM is applied to the shorting of the motor terminals, a simple means to control damping is achieved[78]. The maximum achievable damping is:

$$\tau_b(\dot{\theta}) = -\frac{n^2 \kappa_\tau \kappa_{\dot{\theta}}}{R_e}\dot{\theta}$$

Where $\dot{\theta}$ is the angular velocity of the output link, $n:1$ is the gear ratio, $\kappa_\tau$ and $\kappa_{\dot{\theta}}$ are the motor torque and speed constants, respectively, and $R_e$ is the resistance of the motor coils.

*Hydraulic Damping*

Fluid filled dampers are very common devices, where the size of the channel available for liquid to pass around a moving component is a crucial determinant of the damping force. Laurin-Kovitz, Colgate and Carnes[60] develop a "Binary Damper", a linear damper which has four individual channels whose fluid resistances increase in a $2^n$ fashion. Individual control valves on each of the channels allow them to be opened or closed independently, and thus the overall channel size can be varied with a 4-bit resolution (i.e. 16 possible values).

## 2.4 COMPARING AND SELECTING VARIABLE IMPEDANCE MECHANISMS

Eiberger et al[17] introduce some design criteria for variable stiffness joints in their paper introducing the DLR QA-joint. We will consider those and other criteria and joint characteristics, including some more difficult to define properties such as complexity of manufacture, which are shown in *italics*.

- General actuator design variables
    - Maximum torque
    - Maximum joint speed
    - Maximum/nominal power
    - Inertia
    - Weight
    * *Overall dimensions*
    * *Complexity of manufacture*
- Variable stiffness design variables
    - Stiffness range
    - Elastic deformation range

  - Energy storage potential
  - Energy cost of changing stiffness
  - Rate of change of stiffness
  - Linearity of torque/deflection curve
  - Relation of maximum torque to stiffness setting
  - Complexity of control / can stiffness be varied independently of equilibrium position?
  - Behaviour when elastic range exceeded

- Variable damping design variables
  - Damping range
  - Energy cost of changing/applying damping
  - Rate of change of damping

It can be seen from the list above that in addition to the normal design variables to be considered when designing a robot joint (speed/torque curve, power, physical dimension/mass restrictions), there are many more factors to consider when designing for variable dynamics. Of principal importance are the torque/speed curve, the stiffness range and the damping range - these will always be crucial parameters to ensure the designed system meets its requirements. Depending on the application the other factors will have varied importance, but in general: in robotics light weight and compact-ness are a must, needlessly complicated mechanisms are unlikely to be the best choice. Whilst introducing any variable dynamics will help the system to adapt to different tasks, the rate of change of stiffness/damping will almost certainly be important for any reasonably dynamic tasks.

Grioli et al. [29] consider the design criteria relevant to a user selecting a variable stiffness actuator for a given task. They prescribe a data sheet format for describing VSAs, suitable for when an entire VSA product has been developed, or to give full data on an example implementation. They consider VSA specific characteristics such as the torque/displacement curve and the equations for stiffness, output torque, spring energy, as well as many implementation specific features including output speeds and power, weight, and I/O protocol.

Mechanisms which rely on pretensioning to change stiffness will almost certainly result in a higher cost to change stiffness then those which have only frictional forces to overcome. Such mechanisms must either then be non-backdrivable or will have to continually work against the spring pretension force in order to maintain a given stiffness level.

From a controllability point of view, the dynamics of the joint become more simple if the deflection/torque curve of the joint is linear, and the variable stiffness mechanism can be considered as a standard torsion spring when maintaining a constant stiffness level. It is highly advantageous from a point of both simplifying control strategies and improving physical control if equilibrium position is decoupled from stiffness setting. For example, in the classical implementation of an antagonistic mechanism shown above in figure 6a, both motors must turn in complete synchronisation in order to change equilibrium position without changing the stiffness setting. In contrast, in a series variable stiffness joint one motor controls the equilibrium position whilst the other controls the stiffness completely independently. If an external load has been applied and the joint is not in the equilibrium position then changing the stiffness of the joint will alter the output position, but will still not alter the equilibrium position. Another benefit of this situation is that the drive motor and stiffness motor can be sized according to their respective power requirements - the drive motor, which controls equilibrium position, will generally need to be more powerful than the motor which controls the stiffness setting.

The relationship between maximum output torque and stiffness setting is a more esoteric property which depends on the variable stiffness implementation. For example, in normal antagonistic actuators a lower stiffness setting will result in a higher maximum torque, since more of the spring is available for use; that is to say, the opposing spring is not pre-tensioned so much and is not opposing motion so much. In most series mechanisms the opposite is true, and maximum output torque increases with stiffness setting. The desired characteristic will depend on application, although ideally the full torque of the drive motor(s) should be available at all stiffness settings.

### 2.4.1 *Variable Stiffness for Legged Locomotion*

Walking is a dynamic task that involves contact with the ground. Ground reaction forces must be dealt with without over-stressing the hardware, and ideally without the loss of too much energy. If the natural dynamics of the system can be used, there is potential to greatly reduce the amount of energy used for walking. Furthermore, in any real environment, there will be potential obstacles and irregularities in the ground surface - compliance can provide the opportunity to at least partially adapt around such disturbances.

It is a fundamental design criterion that the joints of the robot are able to produce the mechanical power to produce locomotion, but in addition to this the variable stiffness joints should also be as energy efficient as possible - ideally not wasting power merely to change stiffness. In many designs, the available energy storage or maximum output torque also changes with stiffness. To produce locomotion at high stiffnesses, the robot must be capable of high output torques at high stiffness values, and should provide as much energy storage (and hence compliant range) as possible at these high stiffnesses.

Our high-level criteria when selecting a mechanism for variable stiffness are therefore:

- Deliver the required torque without excessive or exhaustive deflection
  - Exhaustive: Capable of delivering >40Nm in the compliant range
  - Excessive: Can follow a dynamic torque profile - if large amounts of compliant deformation are required to produce the required torques this will require a motor that is fast as well as strong, which is undesirable.
- Produce high torque at high stiffness values
  - To be able to fully investigate the effect of changing stiffness, it is necessary to be able to produce the required torques at various stiffness levels. This precludes using some pretension designs. Reducing the time constant of the joint, it allows energy to be imparted to a link more quickly, and potentially with less overshoot.
- Maximised range of stiffness values

  - This makes the platform more useful for investigating the effect of changing stiffness, but also allows the exploitation of more aspects of variable compliance, including reducing apparent link inertia, absorbing shocks, and passive dynamics.

- Energy efficient when changing/holding stiffness level
  - This is important to reduce the overall energy consumption of the robot. It points towards using a non-pretension design.

- Energy storage independent of stiffness level
  - To fully exploit the compliance of the robot, the compliant elements should be loaded to absorb joint loading, rather than to adjust stiffness. This again points towards using a non-pretension design.

- Maximised elastic deformation range
  - For example, the hip joints of a human swing in a rough sinusoid with an average amplitude of about 20 degrees. To allow passive dynamic strategies to be investigated, the compliant range should be at least $\pm 20^\circ$.

- Simple to manufacture
  - We have access to very limited facilities, and do not have the budget for outsourcing CNC. This means the mechanisms must be implementable without CNC milling, and generally prevents the use of any mechanism with a cam mechanism.

Consider first the principle sagittal plane joints of the robot - the hip, the knee, and the ankle. Looking at human kinetic data (for example from [91]), the torques experienced by each joint during human walking cycle are expressed in terms of leg length and body mass. We can use this to gain a rough figure for the magnitude of the torques which each joint of a robot will experience if it is able to achieve human like walking.

For a robot with a mass of 25kg and a leg length of 700mm, this human data would predict that the maximum torque will be 34.3Nm, experienced in the ankle. We must therefore select a mechanism design which can supply in excess of 40Nm of torque. Further details of specific joint requirements are given in the next section.

Using human data is not ideal for specifying the requirements of the robot, as the mass distribution of the final robot will likely be quite different to that of a human, and this affects the moments of inertia and thus peak torques. However, at the early design stage, this is a way to get a ballpark figure of what the requirements will be. Human torque data is used, rather than data from other robots, since the robot is being designed to walk in a human-like manner, with a heel-toe gait and compliant feet. The actual torque experienced will be different, and so it will be necessary to later confirm these numbers through dynamic simulation of a robot design.

The designed variable stiffness actuators must be capable of running continuously with the periodic loading cycles walking produces. This places constraints, such as on the rated continuous power, on the actuators used to drive the variable stiffness actuators. The variable stiffness mechanism itself must be capable of driving the joint along the required trajectory in position and torque space, and this limits how soft the mechanism can be.

Mechanisms such as the MACCEPA and Uemura VSA are only really suitable for low stiffness joints with relatively low torque requirements, and as such are inherently unsuitable for this application. Of the remaining mechanisms we can construct a decision matrix to evaluate against the aforementioned criteria; for the purpose of this analysis antagonistic mechanisms will be grouped under one umbrella entry.

The decision matrix for choosing a variable stiffness mechanism for the major robot joints is is shown in table 2. This decision matrix utilises the high-level criteria listed above, prioritising the factors which are essential to robot operation, e.g. the ability to provide the required torque for walking, and to be able to provide this torque at high stiffness levels. For the AwAS, a major drawback is the limited elastic deformation range. However, a modified version of this mechanism, which will be presented in the next section, extends this compliant range, and increases the numbers in the table as indicated by the asterisks.

Table 2 shows that for our legged robots, the best variable stiffness mechanisms are either the Modified AwAS, or the AwAS II / CompACT VSA. The major advantage of the AwAS II is that it can vary from zero stiffness to a rigid joint, however it is much more difficult to construct, and passes very large forces through a pivot point. Due the simplicity of its construction, we select the Modified AwAS as the variable stiffness mechanism for the principle joints of our robots.

| | **Provide Required Torque** | **High $T$ at high $k$** | **Stiffness range** | **Efficient stiffness change** | **Elastic deformation range** | **Simple manufacture** | $U_e$ **vs.** $k$ | **Overall** |
|---|---|---|---|---|---|---|---|---|
| **Importance** | ●●●● | ●●●○ | ●●● | ●●● | ●● | ●○ | ● | |
| **Antagonistic** | ●●● | - | ●●● | - | ●●●● | ●●●● | - | ●● |
| **VS-joint** | ●●● | ●●● | ●●● | ● | ● | - | - | ●● |
| **AwAS** | ●●●● | ●●●● | ●●● | ●●●● | ●/●●* | ●●● | ●●●● | ●●●● |
| **AwAS II** | ●●●● | ●●●● | ●●●● | ●●●● | ●● | ● | ●●●● | ●●●● |
| **MIA** | ●● | ●● | ●● | ●●●● | ● | ●●● | - | ●●○ |
| **Choi2011** | ●●●● | ●● | ●● | ●●●● | ● | - | - | ●●○ |

Table 2: Decision matrix - variable stiffness mechanism for main joints

To answer the question of why, since this robot is being design to ape human walking, we do not choose the mechanism that most closely resembles human muscle arrangements. This can be answered most basically by restating the goals of the project: to create a robot which can mimic the human ability to vary the stiffness and damping of joints, and to do this in the most efficient way possible. Even the use of antagonistically actuated joints would be a very rough approximation to the complex muscle arrangements at work in the human leg, and stiffness and damping will almost certainly not be coupled in the way that they are in humans.

The goal of this project is to mimic an ability rather than a morphology, and in fact we want to use a mechanism that decouples stiffness and damping from position as much as possible in order to explore the effect of varying these parameters independently and create any desired trajectory through all three parameters. Making the robot as efficient is a basic design premise, and if it is postulated that in humans the cost of changing stiffness is higher than in the robot, this can easily be factored into the control laws by artificially increasing the cost function, regardless of actual hardware implementation.

### 2.4.2 *Selecting a Variable Damping Mechanism*

For walking, control of damping may not be as important as, for example, in a traditional point-to-point discrete movement task, where accuracy and control of oscillation is vital. However, it is still necessary to ensure that the joints of the robot exhibit an appropriate amount of damping, avoiding unwanted overshoot and oscillations.

There are far fewer available mechanisms for variable damping, and in addition to the criteria listed above it is also necessary to consider the ease with which each mechanism can be manufactured and/or the cost of procuring or manufacturing the mechanism.

Of the available options, *magnetorheological damping* and some *frictional damping* methods require energy to be spent in order to provide damping torque. *Variable hydraulic damping* requires a fluid filled system with a number of valves, greatly increasing the complication of each joint and adding a layer of complexity to the manufacture.

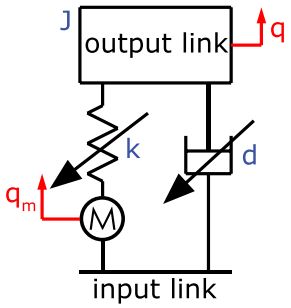



Figure 30: Schematic of the (linear equivalent of the) joint dynamics for each variable impedance joint.

*Motor braking* is the most simple of the mechanisms to manufacture, and is one with which we have previous experience at the University of Edinburgh. The major drawbacks of this method of damping is that it adds weight to each joint, and there is a non-zero level of minimum damping. Since the damping is provided by backdriving a geared down motor, this minimum level of damping may not be insignificant. Depending on the electronics implementation, it is possible to ameliorate the minimum damping issue by driving the damping motor. It would then also be possible to use the damping motor in an "assist" mode, for providing additional torque when required. Whilst usually having significantly less power than that provided by the main drive motor, this additional torque would be provided rigidly rather than through a compliant element, increasing the available actuation options.

Due to the potential advantages of a motor braking variable damping implementation, coupled with the simplicity of its manufacture and the prior experience of our lab with this technology, the selected mechanism for implementing variable damping on the joints of the robots was *motor braking*.

## 2.5 OUR VARIABLE IMPEDANCE JOINTS

As outlined in section 2.4, for the variable impedance joints of our robots, we utilise a modified form of the AwAS variable stiffness actuator and motor braking variable damping. Fig. 30 shows a block diagram of the actuation arrangement used for a full variable impedance joint. Each such joint consists essentially of a Series Elastic

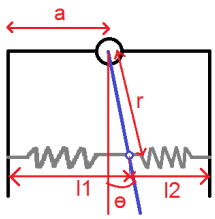



Figure 31: Schematic of MAwAS

Actuator, within which the stiffness of the elastic element can be varied, in parallel with a variable damper.

### 2.5.1 *The Modified AwAS Variable Stiffness Mechanism*

A modified version of the AwAS mechanism is now presented which provides two advantages over the previous implementation:

1. The springs connect to the intermediate arm through a pivoting point which ensures an even, parallel deflection of the springs
2. By moving the intermediate arm and output link into different planes the limit of elastic deformation is now only constrained by maximum spring compression.

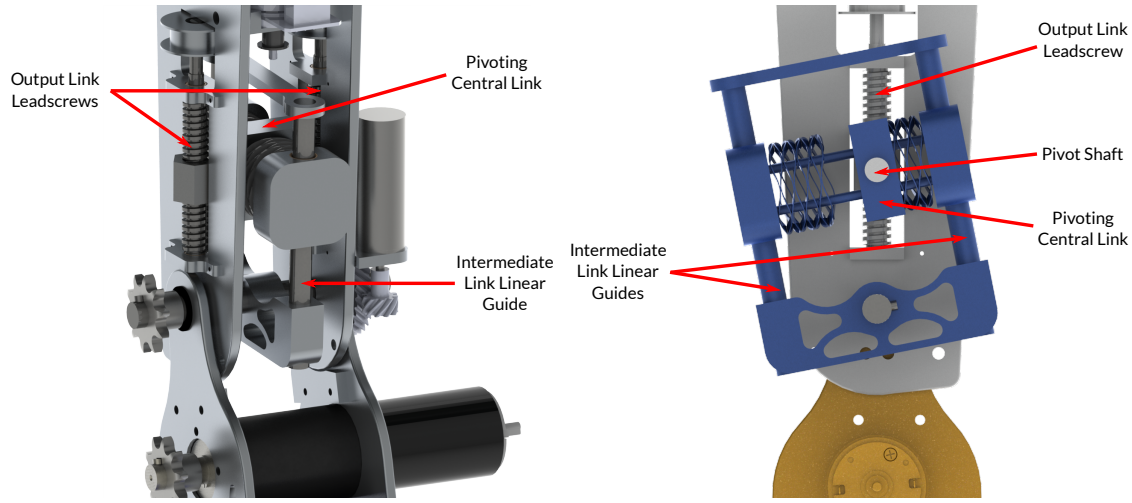
The layout of the design is shown in Fig. 32, where the pivoting centre link is highlighted. The linear guides of the intermediate link can never contact the leadscrew mechanisms of the output link, nor any of the other components of the output link, save for the central pivot itself. This increases the elastic deformation range of the joint.

This mechanism is used in the joints of both robots constructed for this project, and the actual implementation of the design is shown in section 3.2.3.

Considering the geometry (shown in figure 31 we can see that the length of the springs when the joint is out of equilibrium by angle $\theta$ is given by:

$$l_1 = r \sin\theta + a \tag{53}$$

$$l_2 = 2a - l_1 \tag{54}$$




(a) Full MAwAS joint

(b) Mid section of MAwAS joint

Figure 32: The construction of the MAwAS variable stiffness joint

During the set-up of the joint we introduce a pretension $p$ into the springs, to change them from their free length $l_f$. Thus:

$$a = l_f - p \tag{55}$$

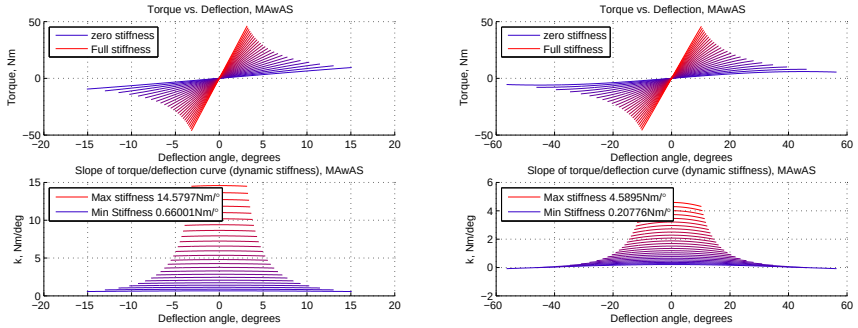
The overall force acting on the middle link at a radius of $\tilde{r}$ is thus given by the difference between the two spring forces at that point:

$$\begin{aligned} F &= K_s(l_1 - l_f) - K_s(l_2 - l_f) && (56)\\ &= K_s(r\sin\theta - p + p + r\sin\theta) && (57)\\ &= 2rK_s\sin\theta && (58) \end{aligned}$$

To calculate the resultant torque we take the component of the spring force which is perpendicular to the middle arm:

$$\begin{aligned} T &= 2r^2K_s\sin\theta\cos\theta && (59)\\ &= r^2K_s\sin(2\theta) && (60) \end{aligned}$$

Thus the stiffness is defined as:




(a) using Lee Spring LW 150 60 0500S (b) Using Lee Spring LW 150 60 1600S

Figure 33: Characteristic curves for theoretical MAwAS actuators using different springs

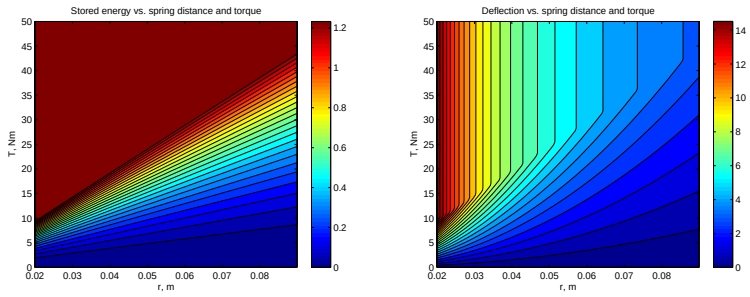
$$K = \frac{dT}{d\theta} = 2r^2 K_s \cos(2\theta) \tag{61}$$

Therefore, as well as being a function of the spring disance, $r$, the stiffness also varies with displacement from equilibrium, being proportional to $\cos(2\theta)$. However, for small deflections $\cos(2\theta) \approx 1$, and therefore for small deflections a constant stiffness is observed. Note that the maximum deflection is limited by the springs being compressed to their minimum length, and as stiffness increases the maximum deflection decreases. If $l_{min}$ is the minimum length of the spring, the maximum deflection from equilibrium position is given by:

$$\theta_{max} = \sin^{-1}\left(\frac{a - l_{min}}{r}\right) \tag{62}$$

Example characteristics for a MAwAS variable stiffness design are shown in figure 33 for two different springs. Figure 33a uses a Lee Spring Wave Spring LW 160 50 0500S which has a spring constant of 47.27N/mm, a free height of 12.7mm and a solid height of 2.285mm. Figure 33b uses another Lee Spring Wave Spring - LW 160 50 1600S, which has a spring constant of 14.88N/mm, a free height of 40.64mm and a solid height of 7.312mm. Wave springs were selected as they allow a more compact implementation than traditional compression springs.

The energy stored in the MAwAS actuator can be calculated based on the deflection of the springs and using the basic spring energy equation $U_e = \frac{1}{2} K_s x^2$




(a) Energy Stored (b) Resultant Deflection

Figure 34: Stored energy in the MAwAS for different stiffness settings and torques

$$U_1 = \frac{1}{2}K_s(l_f - l_1)^2 \tag{63}$$
$$= \frac{1}{2}K_s\left(r^2\sin^2\theta - 2pr\sin\theta + p^2\right) \tag{64}$$
$$\tag{65}$$
$$U_2 = \frac{1}{2}K_s(l_f - l_2)^2 \tag{66}$$
$$= \frac{1}{2}K_s\left(r^2\sin^2\theta + 2pr\sin\theta + p^2\right) \tag{67}$$
$$\tag{68}$$
$$\text{Total Energy, } U = U_1 + U_2 \tag{69}$$
$$= \frac{1}{2}K_s\left(2r^2\sin^2\theta + 2p^2\right) \tag{70}$$

The total energy stored in the springs is therefore given by eq 70, however we should subtract the energy added to the system during construction (as pretension in the springs) to give the effective stored energy:

$$U_p = \frac{1}{2}K_s(2p^2) \tag{71}$$
$$U_e = U - U_p = K_s r^2\sin^2\theta \tag{72}$$

It is therefore simple to calculate the energy stored in the springs as a result of applying a displacment torque to the output link. A contour map of the energy storage, which plateaus as the spring reaches maximum compression, is shown in figure 34.

### 2.5.2 *Variable Impedance Joint Dynamics*

The equation of motion of each joint can be expressed in general form as:

$$J\ddot{q} + d\dot{q} + k(q - q_m) = 0 \tag{73}$$

Defining $\theta = q - q_m$ as the deflection from equilibrium and using eq. 61, this can be rewritten:

$$\ddot{\theta} + \frac{d}{J}\dot{\theta} + \frac{r^2 K_s}{J}\cos(2\theta)\theta = -\ddot{q_m} - \frac{d}{J}\dot{q_m} \tag{74}$$

For small oscillations around $\theta = 0$, $\cos(2\theta) \approx 1$, and therefore:

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2\theta = -\ddot{q_m} - 2\zeta\omega_n\dot{q_m} \tag{75}$$

$$\omega_n = \sqrt{\frac{r^2 K_s}{J}} \tag{76}$$

$$\zeta = \frac{d}{2\sqrt{r^2 K_s J}} \tag{77}$$

For small oscillations, the joint therefore has natural frequency $\omega_n$ and damping ratio $\zeta$ as shown above. Thus by changing $r$ and $d$ the natural frequency and damping behaviour of the joint can be varied.

### 2.5.3 *Scaling up*

From the wide range of variable impedance mechanisms, a design suitable for walking robots has been produced. The specific mechanical implementation will be presented for each robot in the next chapter, but as detailed above, has been selected to reduce energy costs and cope with high torques, as well as considering more pragmatic concerns such as cost and ease of manufacture. This is therefore the fundamental building block of BLUE and miniBLUE, and was decided before beginning the design of either robot.

# 3

# BLUE AND MINIBLUE: NOVEL VARIABLE IMPEDANCE BIPEDAL ROBOTS

In the course of this project, two novel bipedal robots were constructed based around the variable impedance joints discussed in the previous chapter.

This chapter details the design philosophy, criteria, and manufacturing techniques used to produce the robots, before giving specific details on the design of each robot. A modular Ethernet based electronics and communication architecture was developed for use on the robots. The control system is designed to perform accurate low level control of the robot, whilst provide a high-bandwidth, low latency interface to higher level control scripts on a real-time PC. The design of the control boards and communication protocol, is also detailed, along with information on the sensors of the robot. Finally, this chapter outlines the simulation environment which is used for the robots.

The first of robot, BLUE (*Bipedal Locomotion at the University of Edinburgh*) is a planar biped, with six variable impedance joints and a three part foot. The second robot, miniBLUE, is a smaller robot with an additional variable impedance degree of freedom in each hip. miniBLUE has a similar three part foot, with each part being compliantly deformable.

The morphology of both robots is based heavily around human anthropometric data, from the lengths of the individual links to the size of the feet. Typically, bipedal robots have flat single part feet (eg. [38]), which may be exaggerated in size compared to human feet. Certain robots, especially planar robots which are constrained by a boom arm or harness (e.g. [30], have point feet instead, as this simplifies impact dynamics and makes it easier to achieve foot clearance on swing through.

The approach taken in this work is somewhat different. A major goal of the work is to reproduce human-like walking, and specifically the distinctive heel-toe pattern and dynamic rather than static stability. For this reason, we try to produce hardware which is as anthropomorphic as possible, paying particular attention to those parts

of the robot which actually contact the ground - the feet. Again, we seek to copy the style of heel-toe walking, and the efficiency of locomotion it represents, and so need similar kinematics to a human, but we do not need the same style of actuation inside the robot.

## 3.1 DESIGN PHILOSOPHY AND FABRICATION TECHNIQUES

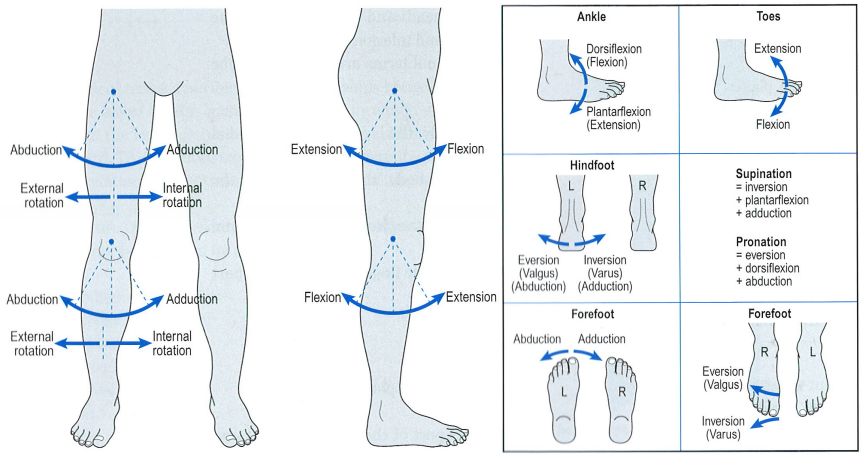
Both the physical construction of a humanoid robot and the goal of reproducing human-like walking are significant challenges. Humans are very flexible and possess many degrees of freedom; a typical simplified representation might show three degrees of freedom in the hip, one in the knee, two about the ankle and possibly a rotation of the toes. However, in actuality the knee is capable of three degrees of freedom, and there are additional active rotations available in the foot, as shown in figure 35[103]. Furthermore the foot is formed from a complex linkage of many bones and exhibits compliant behaviour in more than one plane.[47]

Starting from scratch to build a bipedal platform, it would be too ambitious to attempt to completely recreate the full range of human motion with the first iteration of the design. For this reason, we render the problem more tractable by beginning with a constrained system - the planar biped BLUE - before constructing a more complex system (miniBLUE). In this way the aim is to provide achievable steps on the road to a bipedal robot with fully free 3D motion.

Since the vast majority of joint movement and power in walking (and most other forms of bipedal locomotion) comes exclusively from joints in the sagittal plane[71] it is sensible to create a robot based around these joints first. Although generally requiring support from a boom arm or similar device, it is a normal step when building bipedal robots to begin in this way, and many planar bipeds exist and are capable of locomotion. (e.g. [8], [30])

The first robot, BLUE, therefore has a simple kinematic layout, consisting of six major joints in the sagittal plane - a hip, knee and ankle per leg. The overall layout of the robot can be seen in figure 36. The hardware is designed to allow the attachment of a single DoF rigidly controllable torso joint, although this has not yet been utilised.

Whilst the major motions of walking can be replicated on a sagittal plane biped, there is no capability to "lift the hips" - i.e. perform adduction and abduction of the




(a) Hip and knee (b) Ankle and foot

Figure 35: Rotation about the joints of the leg[103]

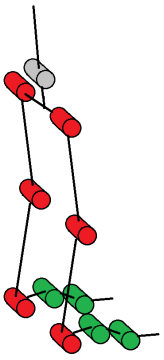


Figure 36: Kinematic layout of robot. Red = Position & Impedance control; Green = Passive; Grey = Position control

legs at the hip. This rotation provides the most power to the human walk, after the rotation of the hips, knees and ankles in the sagittal plane.[71] In addition to allowing a more human-like walk, this degree of freedom also allows the leg to be lifted on swing through, potentially improving swing through clearance. The second robot, miniBLUE, therefore adds this capability.

### 3.1.1 *Foot Design*

The complexity of the human foot was explained in section 1.3.1. From this it is apparent that in order to make a functional foot it is important to replicate the shock-absorbing, energy storing, deformable characteristics of the human foot. Each foot of each robot therefore consists of three parts and has two joints: the transverse tarsal joint, at the apex of the longitudinal arch; and the metarsophalangeal joint, where the toes are hinged.

Typically bipedal robots have had rather simple feet which are not capable of performing the range of functions discussed above. Many robots have shock absorbing pads or bushes, whilst some implement a 1DOF toe joint, either passive or powered; see Davis[15] for a succinct review and an attempt to build an anthropomorphic humanoid foot. Recently an attempt to replicate the windlass mechanism and a variable stiffness arch with a robot foot has been made, and attached to the WABIAN-2R robot.[34] The implementation has limited success and is over complex, but the authors report that "the arch elasticity could absorb a foot-landing force at the plantar contact phase and the windlass mechanism which caused change of the arch height contributed to a strong thrust at the push-off phase." In our robots, the design of the feet allows a windlass mechanism to be evaluated by simply attaching a cable to the underside of the foot. This has not yet been utilised, but in the future the efficacy of this mechanism will be evaluated.

During the second peak of vertical loading in the stance phase of walking the Ground Reaction Force (GRF) just exceeds body weight.[91] This force is transmitted through the arch of the foot, and with the dimensions and target weight of BLUE would result in a turning moment of around *20-25Nm* which occurs at around 80% of the way through the stance phase[83]. In the designed foot some of this moment may be absorbed by the elastic cable of the windlass mechanism, if this is attached.

Hashimoto et al. translate the results of a Japanese paper which claims that the maximum arch stiffness reached during walking is around 8.52 Nm/° for the average human.[34] Maximum deflection in the arch will be around 15°.[83]

The stiffness of the arches of the feet were therefore tailored to support the weight of the robot, as explained in section 3.1.3.

### 3.1.2 *Manufacturing Techniques: Design for Rapid Manufacture*

For reasons of speed, cost, and pragmatic consequences of a lack of available manufacturing facilities, the robots were designed to be as easy to construct as possible. The majority of the structure of each robot was produced using a rapid prototyping technique, and as many off-the-shelf components as possible were utilised, in order to reduce the amount of workshop time required.

#### BLUE

For BLUE, the first robot, the manufacturing technique chosen was waterjet cutting, where sheets of materials (including metals) can be cut accurately to any required profile. This is relatively low-cost and has a fast turnaround, and therefore is an attractive option for designing the chassis of the robot. There are many firms providing waterjet cutting, and once the designs were complete the .stl files were sent to external companies[1] for manufacture. BLUE was designed so that as many parts as possible were cut from uniform thickness aluminium[2].

Since BLUE is a planar biped, each section of the leg is constructed from two plates of waterjet cut aluminium, connected together through crossbars and/or standard size box section. Whilst waterjet cutting provides an acceptable locational tolerance, the rough finish and draft angle on the cut edges mean that precise holes - for example those which will hold bearings - cannot be directly produced. Low draft-angle waterjet cutting can ameliorate one of these issues, the rough finish of the cut is still an issue, and the tolerance of hole may not be sufficient for the required fit. By waterjetting such holes slightly undersize and then drilling and reaming through these

1 e.g. http://www.precisionwaterjet.co.uk , http://www.bigbluesaw.com

2 Aluminium 6082-T6 was utilised, this has a tensile yield strength of around 250MPa. Like most Al alloys, density is 2700kg/$m^3$ (around a third that of steel)

pilot holes, it is possible to create holes with an H6 tolerance (suitable for carrying a bearing) and the locational tolerance of the waterjet cutter.

For bearings, flanged bushings made of Iglidur self lubricating plastic were utilised. These provide appropriate support for the rotating shafts, have excellent loading capabilities, provide a small thrust bearing surface and are very compact, being only 1mm thick.

Whilst effort was made to reduce the number of three dimensional features required for BLUE, there was inevitably a number of such features, for which manual milling and/or turning was required. This work, performed by the workshop in house, took up the majority of the mechanical construction time.

### MINIBLUE

For miniBLUE, the complexity of the two degree of freedom variable impedance hip meant that constructing it purely out of waterjet cut parts would have been very difficult. Furthermore, miniBLUE is a lighter and smaller robot than BLUE, and consequently is under less stress.

The major manufacturing technique utilised for miniBLUE was Selective Laser Sintering (SLS) 3D printing. This technique can be used to create parts with no restrictions on overhangs, and produces largely homogenous parts. We utilise the material "*White, Soft & Flexible*" from Shapeways, printed with a 0.1mm layer size. This material is actually PA 2200 Nylon by EOS GmBH, and has a reported tensile strength of 48MPa.

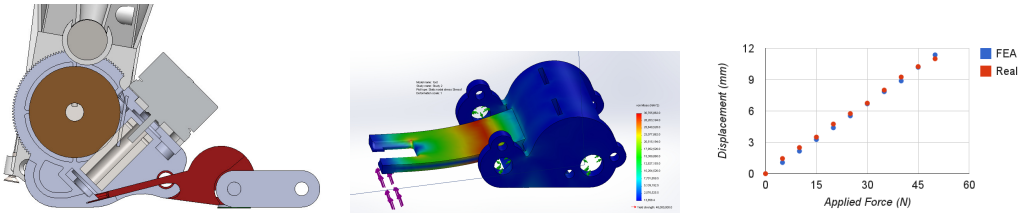
For 3D printed parts, a variety of materials may be used. With Fused Deposition Modeling (FDM) printing, PLA (Polylactic acid) and ABS plastics are most commonly used. For SLS printing, various Nylons (Polyamide) are commonly used, as well as Alumide which blends aluminium and PA powders. With a tensile strength greater than ABS but less than PLA, Nylon is less brittle than PLA, and provides excellent wear resistant properties - important for a robot that will undergo the cyclical loading of walking. SLS printing also reduces the risk of printed parts delaminating, making SLS printed Nylon the material of choice for this application.

Utilising 3D printing allowed us to avoid many of the most time consuming mechanical tasks associated with BLUE when building miniBLUE. An amount of machining was still required, for example turning shafts and milling keyways, however

the number of such tasks was greatly reduced. PA 2200 is quite machinable, and so can be drilled, reamed, milled etc. without melting. Since holes can emerge from the 3D printing process slightly elliptical, especially depending on the print orientation, it was necessary to drill and ream any precise holes during the finishing process.

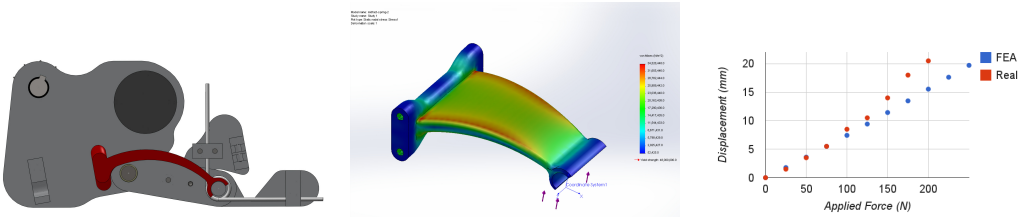
For fixing pieces together on miniBLUE, we primarily utilise three techniques: printing a hexagonal bore for holding a hex nut on one part and a recess for a countersunk machine screw head on the other; inserting threaded brass inserts with knurled outer surfaces, these expand when a screw is inserted to firmly grip the printed plastic; or simply threading the plastic itself. In these ways we create robust assemblies, either clamping parts together, creating robust metal threads in the plastic parts, or where the fixings are only required for components such as circuit boards or sensors (or where the majority of the loading will be in shear) creating threads directly in the plastic parts.

### 3.1.3 *3D Printing For Compliance*




(a) miniBLUE's foot (b) FEA of miniBLUE foot arch (c) Simulated and measured deflection of the arch

Figure 37: Design and evaluation of the compliant feet of miniBLUE




(a) BLUE's foot with 3D printed arch (b) FEA of BLUE foot arch (c) Simulated and measured deflection of the arch

Figure 38: Design and evaluation of the 3D printed compliant arch in the foot of BLUE

The flexible properties of materials such as PA 2200 can be used advantageously in robotic structures, for example to introduce compliance.

Compliant materials are beneficial in the feet of legged robots, which are subject to ground contact. A human foot is composed of a great number of small bones which make it capable of moulding to the contours of the ground and flexing to absorb shocks and store energy in ways which a rigid foot could not. The toes are particularly important since they spread the weight bearing area which is available during the latter stages of the stance phase of walking, before toe-off.

The longitudinal arch of the foot is an important structure for absorbing shocks and providing elasticity in the foot. Ker et al [50] conducted experiments to determine the elasticity and energy storage capability of the spring in the arch of the human foot. They determined that the arch of the foot stores enough strain energy to make running more energy efficient. It can be observed that the arch deforms during walking [26], additionally, there is evidence that the arch of the foot is stiffer towards toe-off. [48] We aimed to ape the functionality of the human longitudinal arch by creating compliant arches in the feet of both robots.

*Compliant Foot Arches for the Feet of Bipedal Robots*

In our robots, we created an elastic joint of high stiffness in the middle of the foot. When the elastic range is exhausted, the joint becomes rigid rather than plastically deforming or breaking.

The placing of the mid-part of the foot in miniBLUE is shown in Fig. 37a. Before the elastic limit of the part is reached, a hard stop is hit and the foot becomes effectively solid. While the arch is shown as a separate part in this figure it can be fabricated as one piece along with the rear foot. In this way, the part is primarily quite rigid, but has a flexible arch with the compliant characteristics we desire, all in one piece.

Such elastic elements were designed in the feet of the robots, and manufactured using 3D printing. The geometry was optimised using FEA to reach a given stiffness, keeping below the tensile strength of the printed material.

FEA analysis was conducted using Solidworks software, using static loading conditions which could be compared to experimentally derived values. The material properties were set to those from the PA 2200 datasheet. The parts were constrained

with bearing conditions at their mounting points - the four shaft bores in the case of the miniBLUE foot part, and the four threaded bores in the case of the BLUE foot arch. For the miniBLUE foot part, the load was applied as a uniformly distributed load, equally across the two prongs of the foot part, to an area at the end of the prongs, which would interface with the toe part in the actual robot. For the BLUE foot arch, a bearing load was applied to the end of the part, to simulate the load transferred from the shaft which would be connected on the robot.

As previously mentioned, in human walking the ground reaction force borne through the longitudinal arch approaches the body weight[83]. For miniBLUE, bearing the entire weight of the robot on one arch corresponds to a loading of 2.25Nm about the hinge of the arch. The analysis was used to tune the arch deflection to reach 14 degrees at this loading, at which point the arch begins to contact the solid support - allowing it to bear additional load without risking damage to the compliant arch itself.

Fig. 37c shows a plot of the applied load on the arch versus the maximum linear deflection, for both the simulated part from FEA analysis and from load testing on an actual fabricated part. As can be seen from the graph, there is a good match between the deflection predicted by simulation and the deflection on the real part.

3D printing can therefore be used to create load bearing parts with tailored compliance as well as complex shapes.

The production of the three dimensional features of BLUE was the most time consuming aspect of the mechanical build. Many of these elements, for example the crossbars, are not under tremendous load, and could be more quickly produced using 3D printing.

The spring in the arch of the foot of BLUE was remade using SLS 3D printing. This is a part which is under considerable load, which must be capable of controlled elastic deformation, and which connects together the two sides of the chassis of the foot.

Fig. 38b shows FEA analysis used during the design of the new arch part. The width and thickness of the part decrease as it nears the shaft connection in order to equalise bending and keep stresses roughly even over the part. At either end of the part, flares with M6 holes allow for connection to the waterjet cut aluminium sides of the foot. These holes are printed slightly undersize, then drilled and tapped.

Fig. 38c shows the FEA simulated deflection for the designed arch, versus measurements taken from loading one of the fabricated parts. In simulation, the arch displays a linear stiffness profile, reaching a deflection of around 20mm under a load of 250N. As can be seen from the figure, the real arch is not as stiff as predicted by simulation, exceeding 20mm deflection under a load of 200N. After this point, additional loading is carried by the metal parts of BLUE, rather than the arch itself, so there is no danger of the fabricated part breaking, even though it is softer than desired. The nonlinearity of the deflection of the fabricated part under heavy loading suggests that empirical analysis should be undertaken when utilising such 3D printed parts under large loads.

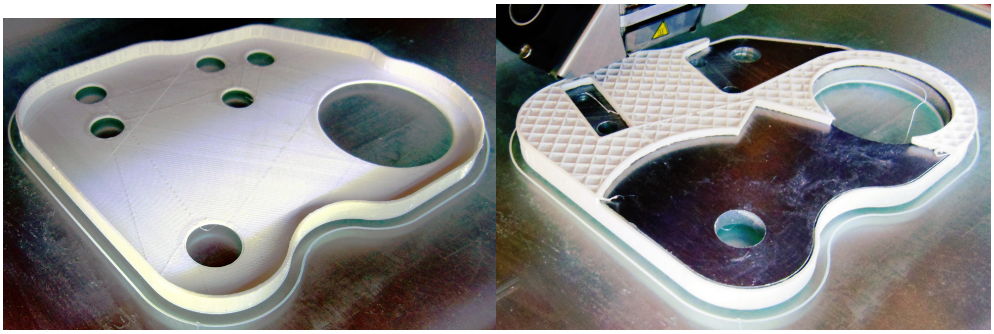
3D printing provides a very easy way to produce the three dimensional features of a robot which is primarily constructed from waterjet cut parts. Parts such as sensor and electronics mounts, cable routing parts, etc. are the most obvious application of 3D printing when supplementing waterjet cutting, but as we have shown, load bearing components can also be produced.

### 3.1.4 *Producing hybrid parts with FDM printing*

Fused Deposition Modelling is another 3D printing technology which prints in layers, but instead of laying down an even layer of powder for every layer, FDM techniques only place material where it is required - typically by extrusion. This means it is possible to introduce additional parts into a partly formed FDM part, before continuing the print.

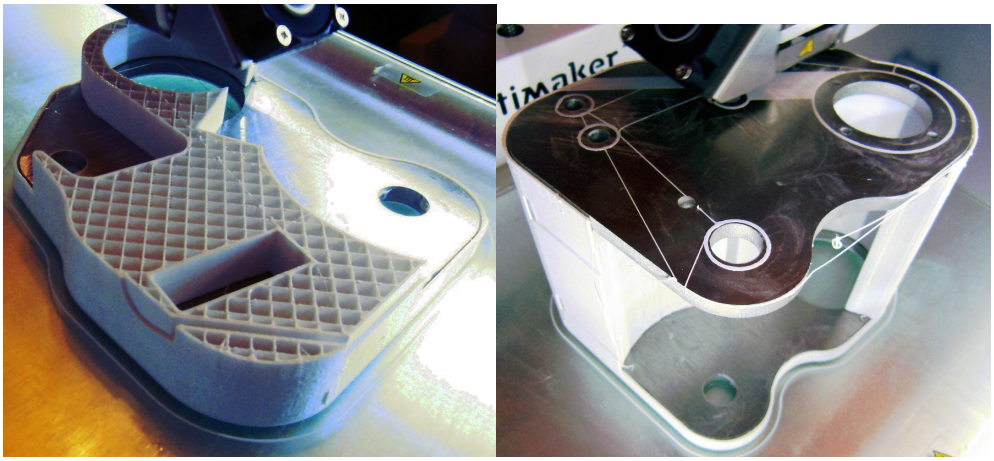
Waterjet or laser cut parts are perfect for coupling with FDM printing, as they are typically a uniform thickness. We can therefore print a shell for a metal part, pause the printing, insert the metal part, and then resume printing. When printing resumes, we can print on top of the metal part, sealing it in and producing a hybrid, single piece part of plastic with encased metal. Alternatively, any sheet material suitable for waterjet or laser cutting may be used, such as acrylics.

Fig. 39 shows a demonstration part produced using this hybrid technique. The part was produced on an Ultimaker 2 printer, which is commercially available at relatively low cost. This printer can be instructed to pause the print at particular layer heights, and for this part it is paused twice, firstly for insertion of the first plate, and a second



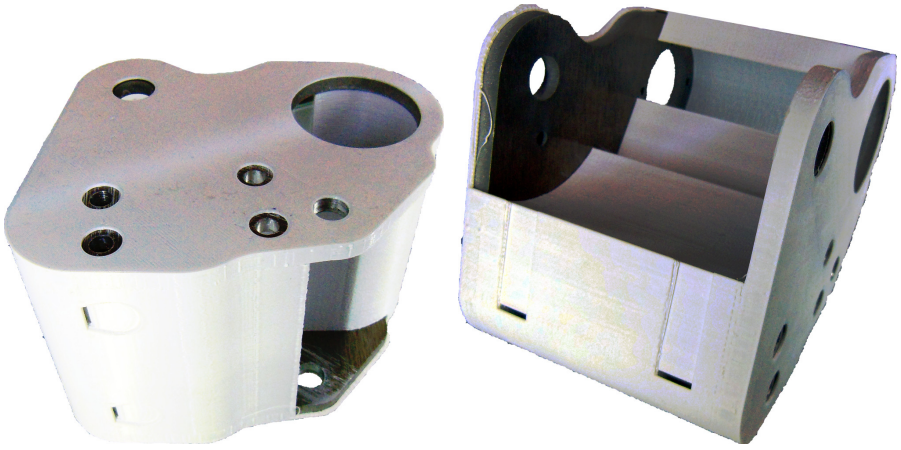
(a) Before first metal piece

(b) After first metal piece




(c) Channel for sensor and recess for crossbar

(d) All metal pieces in place



(e) Complete hybrid part

Figure 39: FDM printing of hybrid part, encasing waterjet cut aluminium

time for insertion of the metal crossbar and the second plate. It is necessary to coat the face of any inserted metal parts with adhesive, otherwise the plastic will not adhere correctly and the print will fail.

Note that in this method, the plastic completely surrounds the metal parts, creating a mechanical fixing between the plastic and the metal. Unless a suitable metal & plastic adhesive is used, there will not be a strong adhesive bond between the metal and the plastic, but there does not need to be as the metal cannot move relative to the plastic. In our example part, the three metal parts - two plates and one crossbar - are mechanically fixed together using screws, and the main stresses in the assembly are borne by this metal "skeleton".

Our experiment shows that this is indeed a feasible construction method. In the demonstration part, there are two metal plates and one metal crossbar, which have all been waterjet cut. The plastic shell holds the crossbar in place, further simplifying the manual finishing process by allowing the drilling and tapping of the crossbar in-situ, without a jig or a mill.

This hybrid technique produces a composite part which is strong and has three dimensional features, and is a technique which can be utilised by labs with limited resources. It is a simple matter to integrate sensor mounts and cable routes, as is shown on our demonstration part in Fig. 39c

## 3.2 BLUE: A 3/4 SIZE PLANAR BIPED

As stated above, the first robot constructed was BLUE, a sagittal plane biped. In order to produce a robot with interesting dynamics that is still a manageable size for working with in the lab it was decided to produce the robot at $\frac{3}{4}$ scale of a full size human.

### 3.2.1 *Specifications*

The robot is a scale model of a human, with link lengths between the hip, knee, and ankle, and foot sizes scaled from anthropometric data. Relevant anthropometric data was taken from two large scale US government studies[16][102] and scaled to the appropriate size as shown in table 3.

| Measurement | Human | Robot |
|---|---|---|
| Hip (Trochanteric) height | 927 | 695.25 |
| Knee height, midpatella | 504 | 378 |
| Foot length | 269 | 201.75 |
| Ball of foot length | 195 | 146.25 |

Table 3: Sizing data and corresponding $\frac{3}{4}$ scale sizes used in the robot. All dimensions in mm.

It is also vital to be able to reproduce human joint ranges with the robot. We can look at the data for typical maximal angles reached during walking/running etc. as well as (where available) the total movement commonly available. Reproduction of the required joint ranges for locomotion is a critical design parameter, and full recreation of possible movements is a design goal. There is a great deal of inter-subject variability in joint ranges, and although the qualitative patterns of joint angles during walking are much the same in normal subjects, there is again variability from person to person. We can therefore look at typical maximum angles for walking, running etc. but should try to exceed these in the design to allow greater flexibility. Some typical maximum angles are shown in table 4 along with the maximum achievable angle for the 95th percentile (i.e. most flexible) of subjects from an anthropometric study.[102]

Regarding the extension of the knee, it should be noted that full extension is normally treated to be 0°, i.e. the lower leg parallel to the upper leg. Extension past this point is referred to as hyperextension (or the disorder *Genu Recurvatum*) and is actually detrimental to gait efficiency and stability[24]. In contrast to other joint rotations, where it is desired to maximise available movement, we will therefore require a mechanism in the robot to prevent hyper-extension and provide support when the knee is fully extended.

A remaining critical design specification is the torque/speed/power which will be required from each of the joints. As previously discussed, walking is a dynamic task with torques varying throughout the gait cycle. To gain an idea of how powerful the motors will need to be we can look to the literature, where torque and power throughout the walking cycle are given normalised by leg length and body weight.

| Movement | Walking | Run/sprint | Possible |
|---|---|---|---|
| Hip flexion | 35° | 65° | 148° |
| Hip extension | 15° | 15° | |
| Knee flexion | 60° | 100° | 145° |
| Knee (hyper)extension | 0° | - | 0° |
| Ankle plantarflexion | 20° | 30° | 79.6° |
| Ankle dorsiflexion | 10° | 20° | 20° |
| MP plantarflexion | - | | 40° |
| MP dorsiflexion | 40° | | 60° |

Table 4: Typical maximum joint angles during locomotion, and (where available) maximum reachable angle for 95th percentile. MP = MetatarsoPhalangeal (Toe)[71][103][70][102]

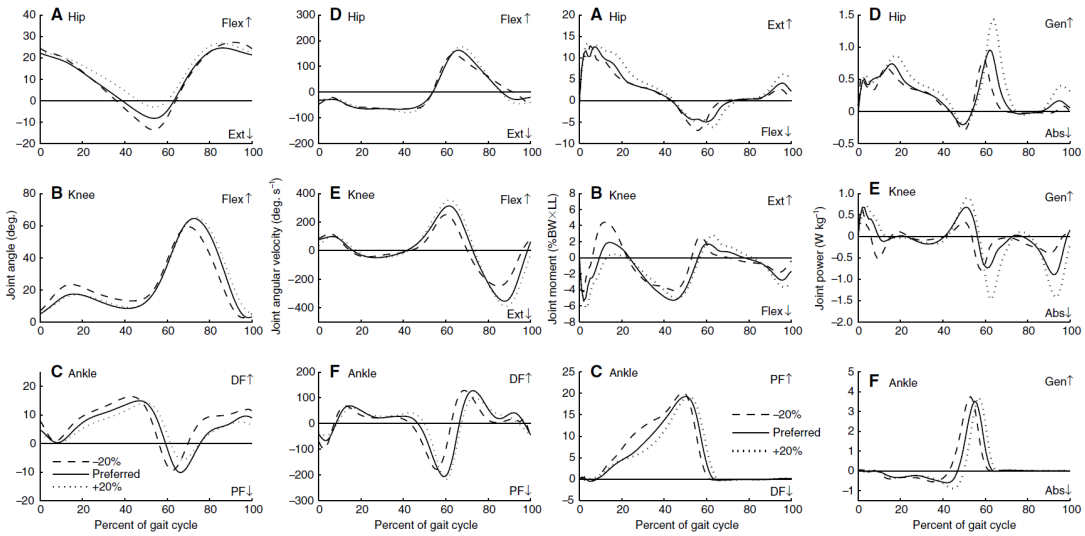



Figure 40: Sagittal plane joint angles, velocities, torques and powers for the hip, knee and ankle at and around preferred walking speed. From Umberger2007[91]

| Joint | Max Velocity | Max Torque | Max Power |
|---|---|---|---|
| Hip | 150°/s | 21.5Nm | 37.5W |
| Knee | 350°/s | 6.9Nm | 25W |
| Ankle | 200°/s | 34.3Nm | 100W |

Table 5: Approximate design requirements for biped with mass 25kg and leg length 700mm

Figure 40 shows the joint angles, velocities, moments and powers during walking at and around a preferred speed.

$$P = \tau.\omega \tag{78}$$

Looking at the classical equation for mechanical power (as shown above in eq 78) we can see that the joint power at any moment is equal to the product of the torque and velocity. Since all three of these vary greatly during the joint cycle, but we cannot easily change (e.g.) the gearing ratio to provide less torque at greater speed, it is necessary to ensure that we can reach the required speeds whilst still being able to provide the required torque elsewhere in the cycle.

The values for torque in figure 40 are normalised by leg length and body weight, and the power values are normalised by body mass. We have already specified the leg length as ≈700mm, and will now define a target weight for the robot as *25kg*. This allows us to extrapolate approximate minimum requirements for motor speed, torque and power, as shown in table 5. The numbers in this table are extrapolated from human studies[91], for walking at a self-selected preferred walking speed. The numbers were later checked in simulation

Since the mass distribution of the robot is different to that of a human, the joint torque and power requirements will be different. With a CAD model of the designed robot, the mass and inertia properties can be used in simulation to check if the motor specifications are adequate and suitable. Fig. 41 shows an example run, with the robot producing successful human-inspired walking using the methods of Chapter 5. In brief - the walking trajectory is generated to produce a heel-toe gait similar to human walking, but considering the additional constraints of a sagittal plane only robot and ensuring foot clearance on swing through.

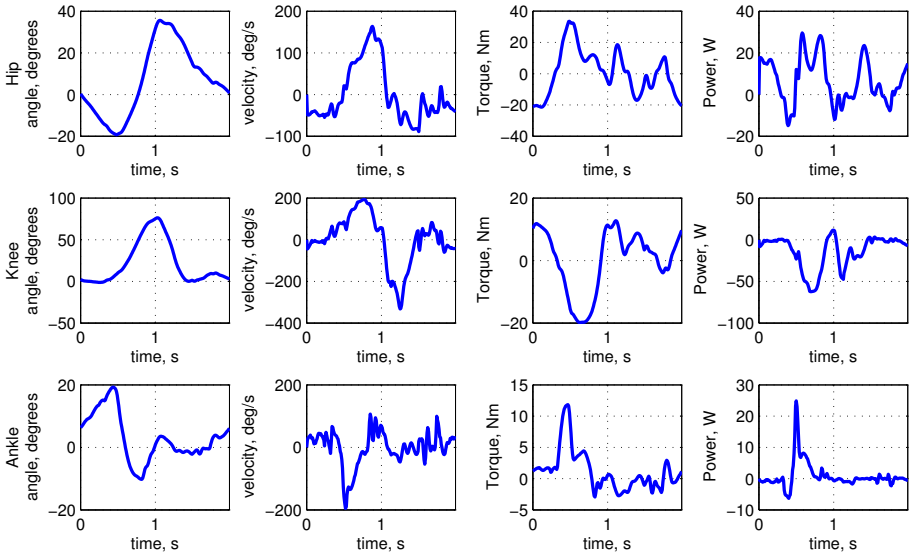



Figure 41: Example joint angles, velocities, torque and power for the left leg of the robot during simulated walking. The trajectory used for this example was generated to be suitable for the robot, using the method described in section 5.2.2, run at a walking period which produced similar joint velocities to the self selected human walking speeds.

It was observed from running simulations that the power and torque required for the knee were significantly higher than extrapolating from human walking would suggest, while the magnitude of the torque and power spike on the ankles was lower.

### 3.2.2 *Motor Selection*

Crucial to the satisfactory operation of the robot is the selection of motors and gearing which are capable of handling the torque/speed/power requirements, whilst not being too heavy or inefficient. It is a typical catch 22 situation in mobile robotics that in order to get the required power you must use heavier motors, which in turn increase the power requirements, necessitating heavier motors, and so on. It is important to strike a balance and employ the motor & gearing solution which provides the necessary power at as light a weight as possible.

The problem of motor selection is of course limited by what is commercially available, and the reasons behind the selection of the drive motors and stiffness adjustment motors will be explored in the following sections. In addition to just choosing a motor, for the drive motors it is necessary to also choose how to gear down from the high speed/low strength output of the motor to the low speed/high strength movement we require for the joint positions.

#### 3.2.2.1 *Main drive motors*

We can expect a maximum motor efficiency of around 90%, and a gearing efficiency in the region of 70%, meaning that to reach the peak ankle power shown in table 5, a motor power of around 150W is temporarily required. Whilst this power is only required for a short amount of time per cycle, and it is possible to overdrive motors above their rated power for a short period of time without doing any damage, it was decided that it was better to select a motor with this power. This means that we will not be overdriving the motor on each cycle, but can do it if we require power in excess of that used in a normal walking cycle. In keeping with the modular design of the robot, and recognising that the final robot will be required to perform other movements as well as walking, it was decided to use the same motor in each of the joints (adjusting the gear ratio where necessary).

| Motor | Type | Power | Weight | Nom. speed | Nom. T | Cost |
|---|---|---|---|---|---|---|
| Maxon RE-40 | Brushed | 150W | 480g | 7000rpm | 0.184Nm | $273 |
| Emoteq QB01702 | Brushless | 156W | 470g | 7800rpm | 0.184Nm | $1134 |
| Maxon EC-4pole 30 | Brushless | 200W | 300g | 15800rpm | 0.059Nm | $522 |

Table 6: Comparison of three motors

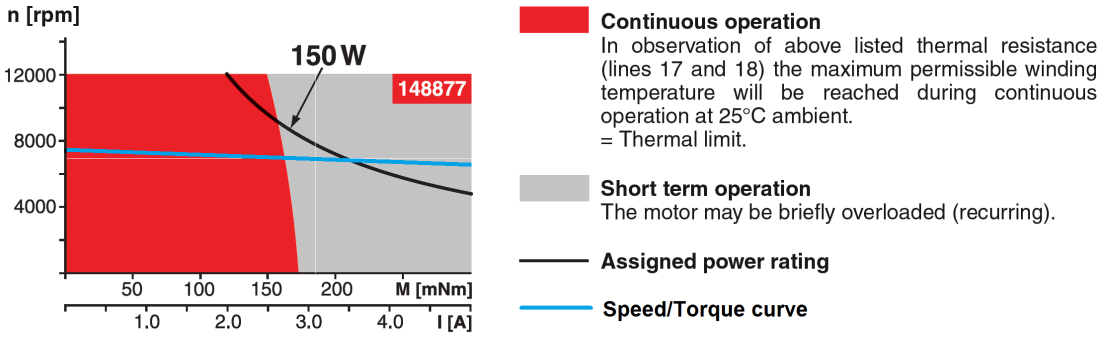



Figure 42: Maxon RE-40 operating range, from http://www.maxonmotor.co.uk

The motors themselves can be either brushed or brushless, the more traditional brushed motors requiring more simple controllers than the brushless motors, which require electronic commutation in order to work. Brushless motors suffer less from wear, and can often be lighter for the same power, however they tend to spin faster (requiring higher gearing) and are usually considerably more expensive. Neither type of motor is inherently more efficient, with good quality motors of both types achieving maximum efficiencies of around 90%. For comparison, three motors of a similar overall specification are shown in table 6.

From table 6, either of the Maxon motors appear to be good choices. The brushless *Maxon EC-4pole 30* is more powerful and lighter, however as a brushless motor it requires a more complex and expensive controller than a brushed motor. It is also almost twice the price and will require a higher gear ratio than the slower, brushless motor - possibly incurring extra weight. Since the weight advantage is not as clear cut as indicated by the table, it was decided to utilise the brushed *Maxon RE-40* motors as the main drive motors. Figure 42 shows the operating range for this motor, along with the speed/torque curve which the motor follows. The no-load speed of the motor is 7580rpm, and the stall torque is 2.5Nm. The 48 volt version of the motor is selected as it is slightly more powerful, and allows higher powers at lower currents.

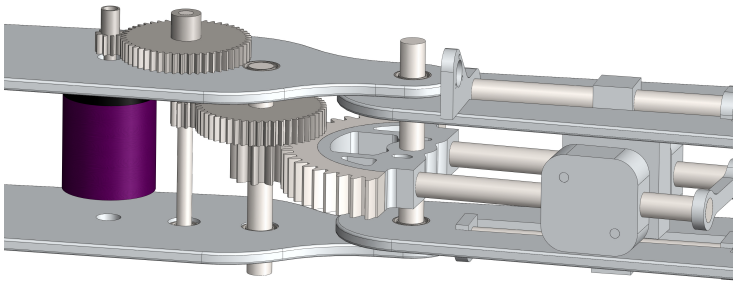


Figure 43: Possible spur gearing implementation. 125:1 ratio

With the motors selected it was necessary to choose the method by which they were geared down. The principle gearing options are: spur, planetary, worm and harmonic. Harmonic drive gearing is a complex mechanism which can achieve high gear ratios in a single stage with efficiency of >70%, however the price of harmonic drive units is very high, beyond the budget of this project. Spur gearing is perhaps the most efficient way to gear down, with each stage of gearing having a theoretical efficiency of up to 98-99% per stage. It should be possible to achieve a ratio of 5:1 per stage, and so with a three stage spur gear box we would achieve an overall gearing ratio of 125:1.

A possible design incorporating this gearing is shown in figure 43, this has a ratio of 125:1 and should achieve an efficiency of >90%. The maximum torque the output stage of the gearing can accommodate the teeth break is 38.8Nm, although any misalignment between the gears will reduce this maximum torque as well as the efficiency. The overall weight of the gearing is around 300g (+ the weight of the shafts), although a larger safety factor should be in play, and increasing the maximum permissible output torque would increase the weight of the gearing significantly. This arrangement of gearing is also quite bulky, which is not ideal since there is a lot to fit into the legs.

Worm gearing provides a way to achieve high reduction ratios in a single stage. To achieve an overall ratio of 100:1 (for example) would probably involve a first stage spur gear at 4:1 ratio, followed by a double start worm stage achieving 25:1. Whilst reasonably efficient, the total weight of such a set-up would be around 1kg, and thus far too heavy to use in this application.

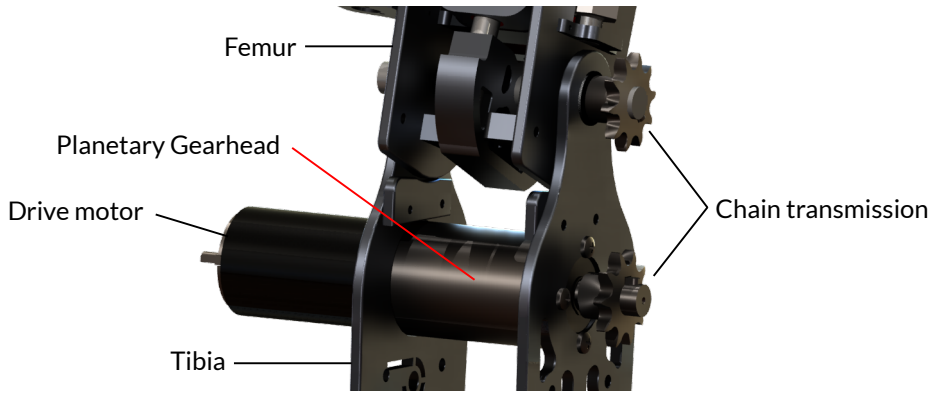



Figure 44: Knee joint of the robot. Maxon RE-40 drive motor with planetary gearhead is in the tibia, with a chain transmission to the output (chain itself not rendered)

Planetary gearing involves the use of multiple gears per stage, and because of this is less efficient than spur gearing, but can handle higher torques. It is also probably the most simple method of gearing to implement, since Maxon provide planetary gearheads at a range of ratios from 3.5:1 to 1296:1 which can be ordered ready-attached to the motors. A three stage planetary gearhead with a reduction ratio of 74:1 has a maximum efficiency of 72% and a weight of 460g. Nominal output torque is 15Nm, and "intermittently permissible" output torque is 22.5Nm. Whilst total strength is not given, it is likely to be in the region of 35Nm.

The final transmission to the shaft which drives the intermediate arm of the variable stiffness mechanism is by a chain drive. The 8mm chain utilised has a breaking strength of 5000N, since the smallest pulley used has a pitch diameter of 25.9mm, this corresponds to a maximum transferable torque of 64.75Nm. A 2:1 ratio in this final stage can increase the intermittently permissible output torque to twice that of the planetary gearhead: 45Nm (The maximum torque before breakage is not known, but is likely to be around 70Nm). A render of the chosen motor and gearhead with chain drive pulleys is shown in figure 44. The efficiency of the chain transmission will is theoretically around 98%, although this will decrease with the use of chain tensioners.

For the ankle the gearhead ratio is chosen to be 74:1, followed by a 2:1 reduction at the chain drive, giving a total ratio of 148:1. This means that the nominal output torque will be approximately 20Nm at a speed of 280°/s. The maximum required torque of 34.3Nm can be provided at a speed of around 260°/s, faster than required

by the data shown in figure 40 and table 5. The knee has to move faster with less torque, and therefore utilises a planetary gearhead with a ratio of 43:1 followed by a chain ratio of 2:1. Nominal performance is then with an output torque of 11Nm at a speed of 488°/s. Again this exceeds the requirements, leaving room for overloading and the performance of more taxing movements. The hip utilises the same gearing as the knee.

#### 3.2.2.2 *Stiffness adjustment motors*

The motors which turn the leadscrews to adjust the position of the springs, and hence the stiffness, do not need to be as powerful as the main drive motors. When the mechanism is in equilibrium the only forces that must be overcome are those due to friction and the movement of the springs and mounts (which do not have a great inertia and thus do not require much force to move). Out of equilibrium there is a component of the spring force which will project down the intermediate arm, and any adjustments to make the mechanism stiffer will have to work against this force. Just as the normal component of the spring force, and hence the applied torque, is given by $T = 2rK_s \sin\theta \cos\theta$, the parallel component acting down the intermediate arm is given by $F_p = 2rK_s \sin^2\theta$.

When considering the energy cost of changing stiffness, it does not make sense to consider making the joint more stiff whilst keeping deflection constant, as this would imply that the external torque had increased dramatically. If we assume that the torque stays the same we can observe that the deflection from equilibrium, and hence the energy stored in the springs, will decrease (see figure 34). Some of this released energy will in effect help to move the output link closer to equilibrium, while the rest must come from the stiffness motor. We can consider the energy used to change stiffness whilst keeping the amount of energy stored in the system constant.

The force exerted by the springs, $F$, is therefore fixed, and we can define the deflection angle in terms of this force and the arm radius:

$$\sin\theta = \frac{F}{2rK_s} \tag{79}$$

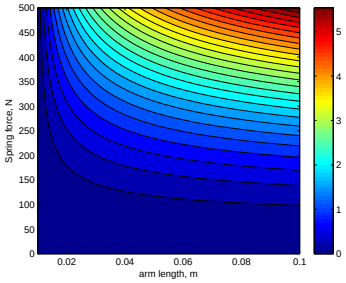



Figure 45: Energy required (J) to change stiffness with different internal spring forces

And since the component of force acting down the intermediate arm is $F_p = F \sin\theta$,

$$F_p = \frac{F^2}{2rK_s} \tag{80}$$

The energy used working against this force to change stiffness can therefore be calculated by a simple integration:

$$E = \int_{r_1}^{r_2} F_p \, dr \tag{81}$$

$$= \frac{F^2}{2K_s}\big(\ln(r_2) - \ln(r_1)\big) \tag{82}$$

Using realistic numbers, to go from minimum to maximum stiffness with $r_1 = 0.01$ and $r_2 = 0.1$ against a spring force of $F = 500N$ with spring constant $K_s = 50000$ would use 5.75J of energy. The efficiency of the lead screw transmission mechanism is roughly 44%[3], and therefore the energy required from the motor would be around 13J. To make this stiffness change within 1 second would therefore require a motor power of 13W. The pitch of the leadscrews used in the design is 3mm, so to move 90mm would require 30 revolutions of the leadscrew in one second, or 1800rpm. This is an oversimplified analysis, as in actuality there will be additional frictional forces which are not modelled, but it serves as a rough guide to how little power is actually required to adjust the stiffness (compared to moving the joints).

The motors used for stiffness adjustment should therefore be lightweight and high speed, but do not need to be too high powered. From the above analysis, and using a gearing ratio of 2:1 from the motor to the leadscrews, the required motor torque

3 $\eta = \frac{\cos\theta_n - \mu_s \tan\alpha}{\cos\theta_n + \mu_s \cot\alpha}$, where $\mu_s$ is the coefficient of friction, $\theta_n$ is the thread angle, $\alpha$ is the helix/lead angle.

would be 34.3mNm at a speed of 3600rpm (neglecting minimal energy losses due to the timing belt transmission[4]).

For BLUE, we utilised Maxon EC45 flat 24V 50W brushless EC motors. These motors have a high power to weight ratio, weighing 110g, and produce a nominal torque of 83.4mNm at 5240 rpm, with a stall torque of 780mNm.

### 3.2.3 *Joint Design*

As mentioned previously, the MAwAS mechanism is a modified version of the AwAS actuator by Jafari et al.[44] The equilibrium position is set by an intermediate arm, and the drive is transferred from this intermediate arm to the output link through two springs. By changing the distance of these springs from the rotational axis the stiffness of the joint is regulated. The possible stiffness ranges are shown in figure 33 for two different commercially available springs. The major changes made in this design is that the spring mounting points are kept parallel at all times through a pivoting linkage, and that the intermediate link and output link are in different planes, allowing a large deflection at lower stiffness ranges without hitting mechanical limits.

The implemented design is shown in figure 46, where a cutaway of the side view, a front view, and a render of the joint deflected from equilibrium are shown. The diagram on the left shows the two compact wave springs, and the bars of the intermediate arm on either side. The equilibrium position is set by driving the shaft of the joint, to which the semi-circular base of the intermediate arm is attached. The drive motor is located a short distance from the axis of rotation, and drive is transferred through a chain mechanism. This allows the transfer of large moments across the required distance without requiring heavy gearing, and allows for an easy adjustment of gear ratios on the robot by simply changing the ratio of the motor and shaft pulleys. The two spring mounts on either side of the intermediate arm are free to slide on the bars, their distance from the rotational axis controlled by two small guide links to the pivoting spring mount of the output arm.

The position of this spring mount is controlled by moving the shaft around which it pivots, and which transfers the drive to the output arm. Each end of this shaft is connected to a hexagonal nut on a leadscrew, and in this way the drive is transferred

4 Timing belt efficiency is $\leqslant$98%

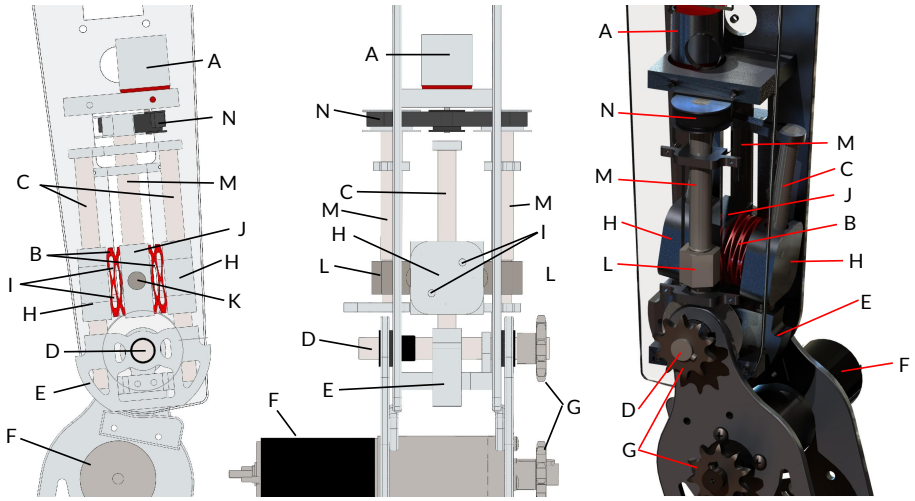



Figure 46: Example Implementation of MAwAS mechanism. The components are: A) Stiffness adjustment motor B) Wave springs C) Intermediate arm bars D) Joint shaft E) Intermediate arm base F) Drive motor G) Chain transmission H) Spring mounts I) Spring mount guide links J) Pivoting spring mount K) Pivoting spring mount shaft L) Hex nuts M) Leadscrews N) Timing belt transmission

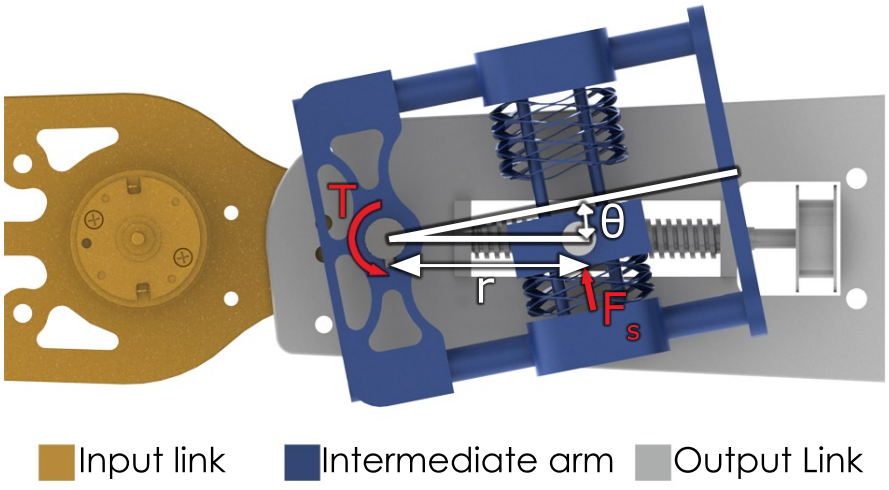
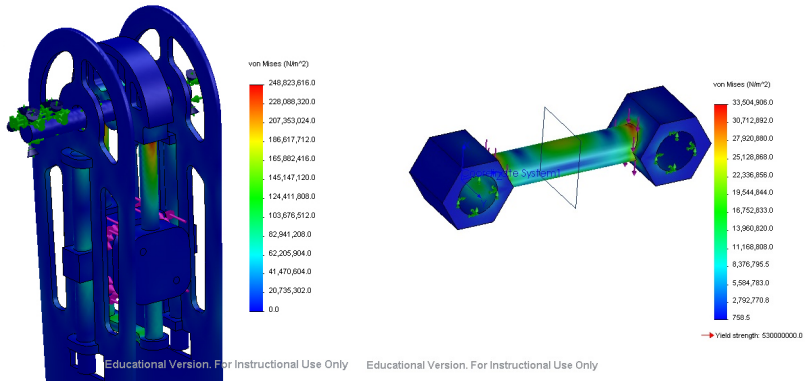



Figure 47: MAwAS mechanism with large wave springs




(a) Major transmission components

(b) Spring pivot shaft

Figure 48: FEA for MAwAS mechanism transmitting 80Nm

symmetrically to both sides of the output link. The leadscrews are driven simultaneously by a timing belt transmission from a central drive motor, this ensures that the two nuts holding the spring pivot shaft move together and provide a smooth stiffness change.

To ensure that the designed mechanism can cope with the kind of forces it will be subjected to when used in the bipedal robot. As well as carrying out some basic strength/shear force calculations, Finite Element Analysis (FEA) was conducted on the designed MAwAS mechanism. Some example diagrams from the FEA analysis are shown in figure 48, in which a torque of 80Nm was simulated. This represents a safety factor of around twice what the mechanism should ever have to endure, and the induced stress is still short of the 250MPa yield strength of the aluminium, and ~500MPa strength of the steel.

Figure 49 also shows the connectivity of the variable damping motor, which is braked to apply damping torque to the joint. The motor is connected in parallel with the drive motor and variable stiffness link, and in BLUE helical gearing is used to connect the damping motor and output link.

Figure 50 shows the theoretical torque and stiffness curves for the variable stiffness units used in the hips and knee/ankles of BLUE. The performance of the joints was later tested by performing a simple squatting experiment - the robot repeatedly produced a slow squatting behaviour while the stiffness was changed - and the results compared to those from simulations.

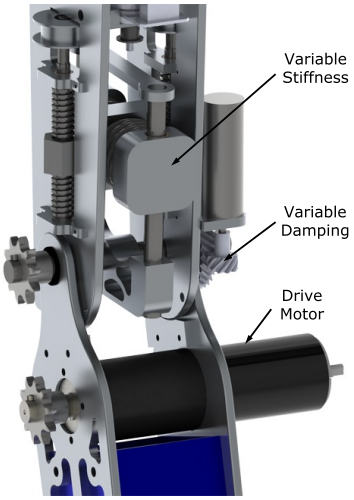
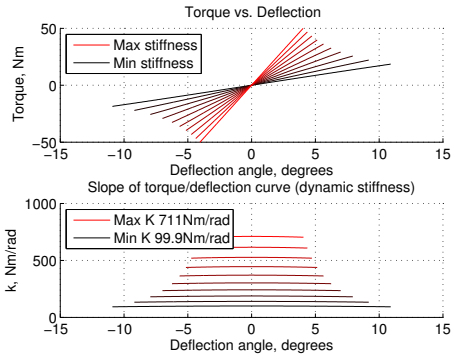
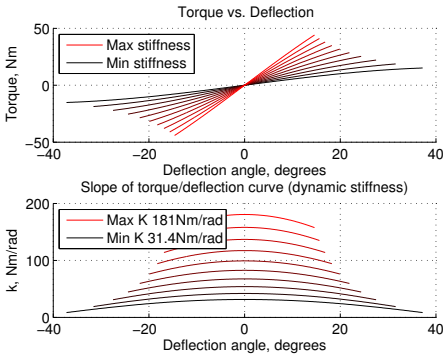



Figure 49: The variable impedance joint in the knee




(a) Hip




(b) Knee, Ankle

Figure 50: The theoretical torque and stiffness characteristics for the variable stiffness joints used in BLUE

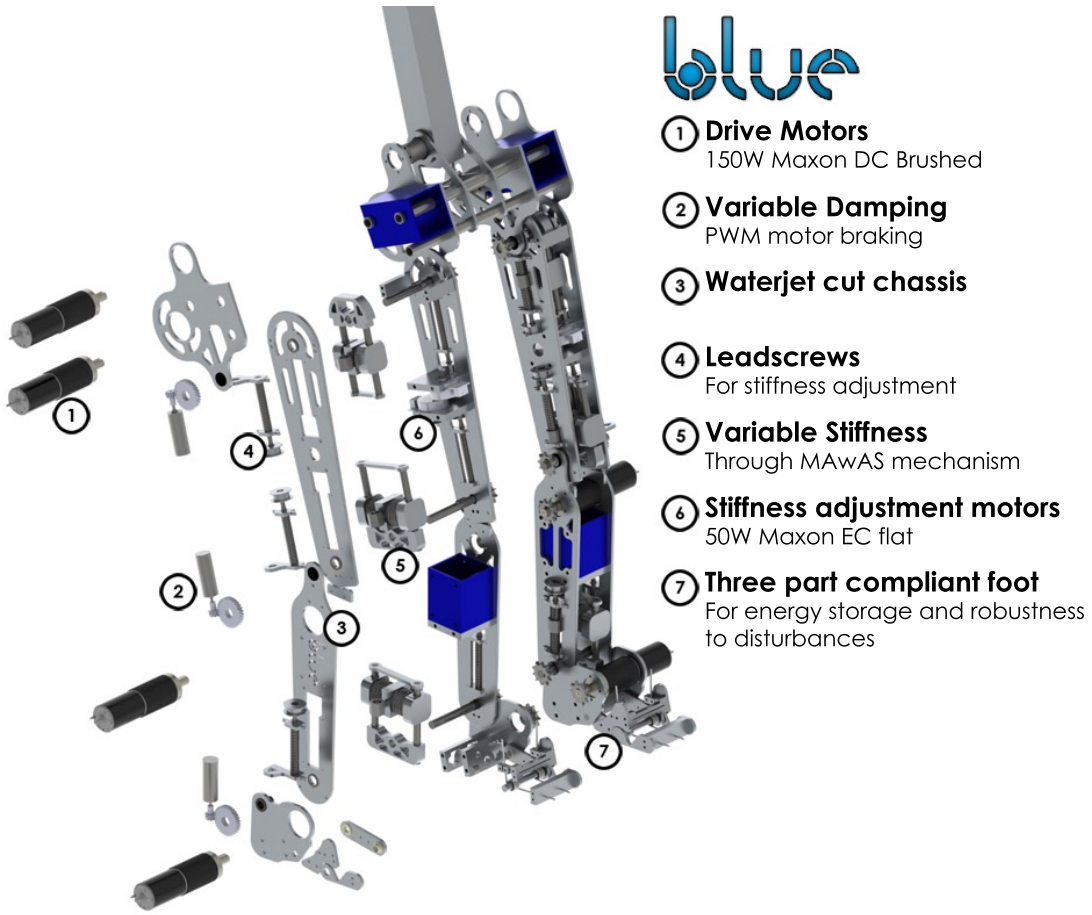



Figure 51: Exploded view of BLUE

### 3.2.4 *Mechanical Design*

As previously detailed, the chassis of BLUE was constructed from waterjet cut aluminium. An exploded view of the design is shown in Fig. 51, and shows the variable impedance mechanisms of each joint along with the drive motors, and the crossbars and box section. As shown in the figure, the top part of each leg - one side of the pelvis - consists of two aluminium plates with a piece of box section in between. Three holes pass through this assembly, and these holes are used to mount each leg on a set of three bars.

Knee hyperextension is prevented through a mechanical stop. Small waterjet cut pieces of aluminium are attached to the inside of the lower leg, as shown in Fig. 44, and prevent the knee rotating below zero degrees. These parts are designed such that a small piece of rubber is affixed by adhesive to the load bearing surface of the knee

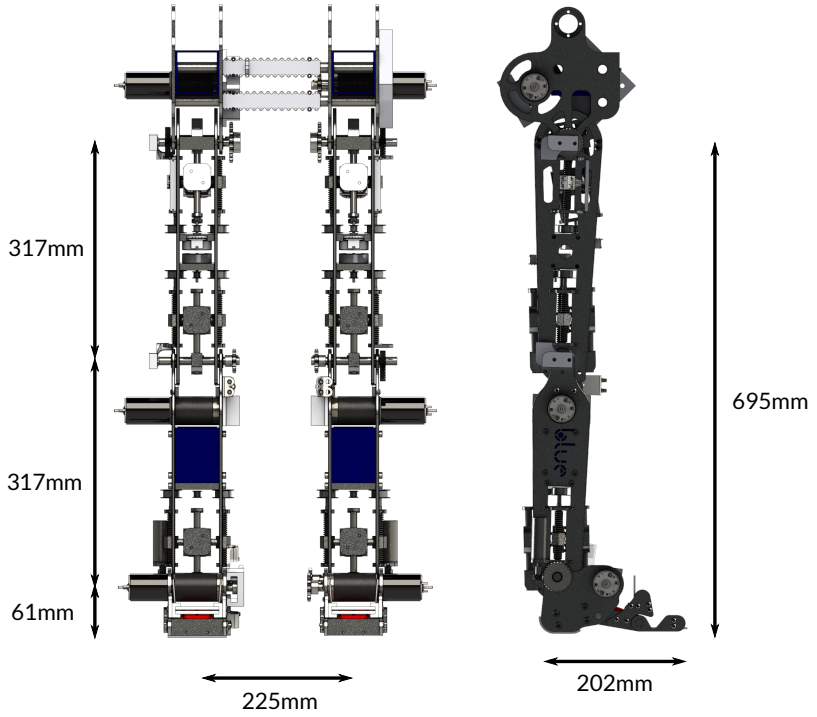



Figure 52: Dimensions of BLUE

stops, in order to cushion the impact and provide shock absorption when the knee stops are used.

To provide the required lateral support to BLUE, a boom arm was designed out of tubular steel and extruded aluminium. The boom arm has a centrally rotating pole to provide forward/backward motion, with a hinged joint at the top allowing vertical motion. An exploded view of this boom arm is shown in Fig. 53. The lower section of the boom has a number of feet which can be screwed down into an appropriate floor or board. A solid 25mm steel rod extends upwards from the centre of this base section, and the upper section rotates around this. Oilite bushings position this rod within the central tube of the lower section, and provide a thrust bearing surface for the upper section to rotate on.

The upper section fits around the steel rod, with two large needle bearings allowing for rotational movement around the rod. Towards the top of the upper section is the vertical movement hinge, and the boom itself. Further down the upper section, an arm extends from the central tube, allowing a bungee, webbing or rope etc. to be attached between this arm and the distal projection of the boom. In this way the

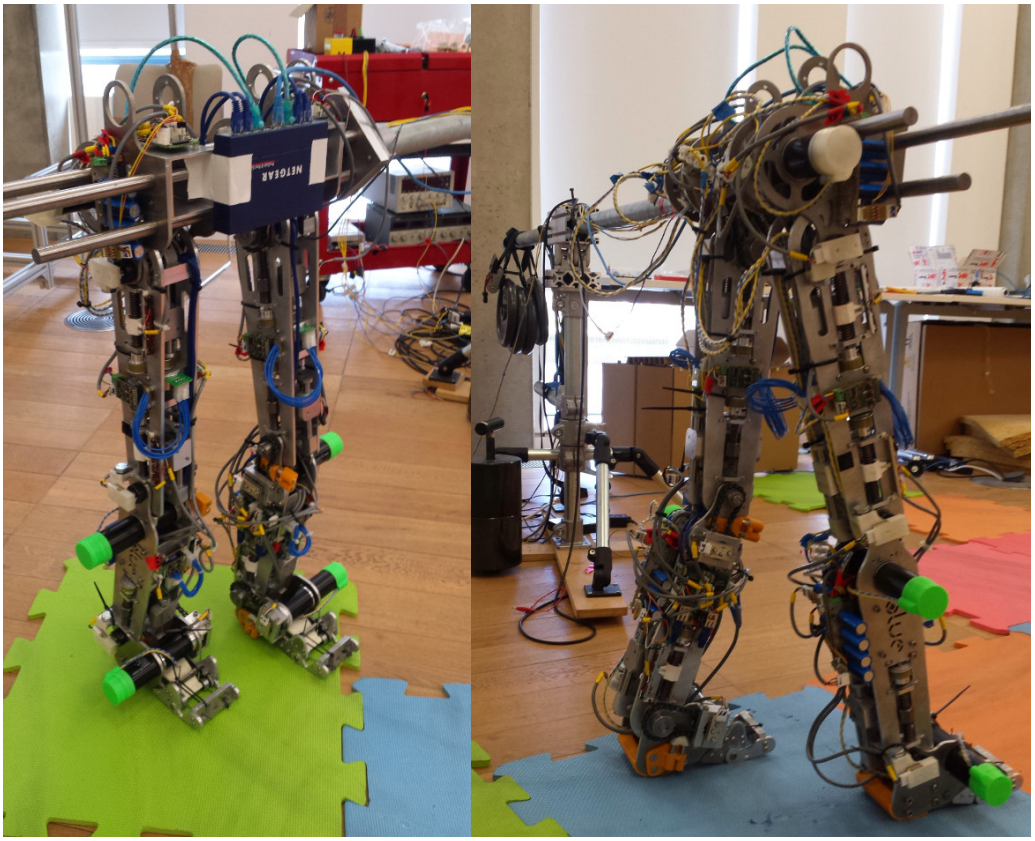
Figure 53: Exploded render of the boom arm designed to provide lateral support to BLUE

vertical motion of the boom can be limited, for example to ensure that if the robot falls it does not fall all the way to the floor.

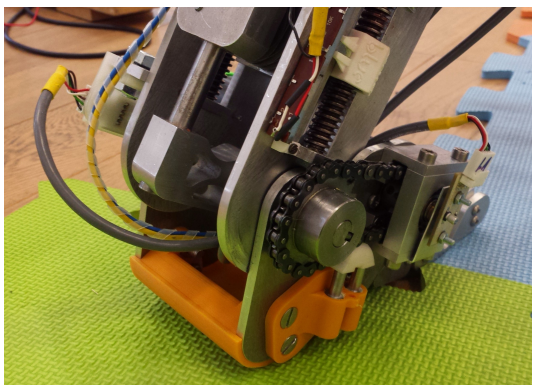
### 3.2.5 *Completed Robot*

The completed robot is shown in Fig. 54. Overall, the robot has performed largely as it was designed to do, it is capable of producing locomotion and other behaviours, and all the variable stiffness mechanisms function. There are two main issues with the mechanical hardware.

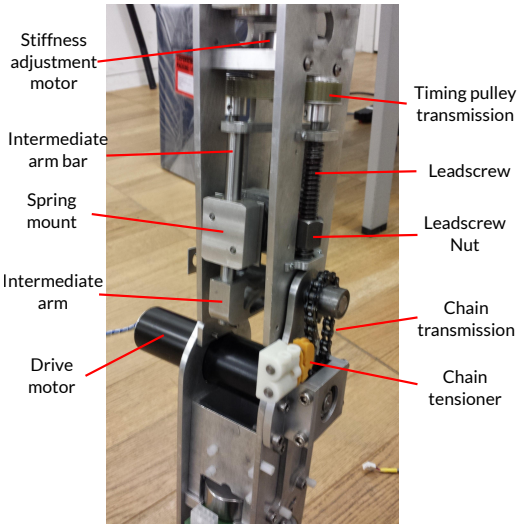
The first is that when the joints are under loading, it is very difficult to adjust stiffness. This is due to large amounts of friction created on the low-cost trapezoidal leadscrews, and restricts the times at which stiffness can be adjusted. This problem could be addressed in two ways in future versions: by using ballscrews rather than leadscrews; and by transferring load through linear guides (i.e. solid shafts) rather than through one or more linear actuators themselves. This problem was addressed



(a) (b)



(c) Left foot




(d) Left knee (without electronics)

Figure 54: Views of the completed robot BLUE

in miniBLUE with a more robust stiffness adjustment mechanism, using linear rails and a precision leadscrew.

Further to this, the stiffness adjustment motors should be geared down further, to ensure that stiffness adjustment can be done under load. This would be at a trade-off against the rate at which stiffness could be adjusted.

The second issue is one that has become worse over time - backlash around the shaft keys. There was always an amount of backlash in the joints, due primarily to the chain transmission, slack around the leadscrews, and partly from the planetary gearhead. However, there are three keys in the drive train between motor and intermediate arm, and over time the large oscillatory loading on these keyways has worn them, so that there is now a large amount of backlash in the joints of the robot.

In a future design, BLUE would utilise harmonic drives with no chain transmission, and any connections to shafts would be done with locking collars, or at least with shafts larger than 12mm. However, at the time when miniBLUE was designed, the backlash problem was not evident in BLUE, and hence miniBLUE also utilises keyed transmissions.

Regarding the electronics, the resolution of the deflection and angle sensors should be improved. The resolution of a third of a degree is not sufficient to provide accurate torque information. The sensors could also be arranged in such a way that they can detect backlash in the transmission explicitly, and factor this into control and evaluation algorithms.

While the electronics can motor the current through the drive motors, the current through the stiffness motors cannot be sensed. This information could be useful for evaluating overall electrical energy usage in a scenario where stiffness was being varied dynamically, but is also useful as a safety check in case the motor stalls. As previously discussed, the motors struggle to adjust the stiffness of BLUE under load, and so detecting if a motor had stalled would be very useful.

In a future design, smaller, lighter EC motors could be used. As well as reducing overall weight, this would allow for more compact EC motor drivers to be used. miniBLUE utilises only EC motors.

If possible, the mass distribution of the robot could also be improved, particularly to reduce the mass of the foot, which is very heavy on BLUE.

Nevertheless, BLUE still provides a functional platform on which to explore bipedal motions at various stiffness settings. The validation of the robot, and experiments on it, will be presented later in the thesis.

## 3.3 MINIBLUE: A 1/2 SIZE BIPED WITH 2DOF HIPS

The planar joints of BLUE are capable of recreating the vast majority of the motion required for human-like locomotion. The next step towards a bipedal robot with full three dimensional freedom is to introduce hip adduction/abduction, as this is the next most important rotation in terms of power usage during walking motion[71]. Furthermore, it was observed from working with and generating trajectories for BLUE that there were sometimes issues ensuring foot clearance on swing-through. The ability to abduct and adduct the hips allows a far greater control of the height of the foot during swing through.

In addition to this additional motion, it was also desired to produce a robot which is more light-weight and compact than BLUE. This is for the pragmatic reason of creating a robot which is easier to work with in the lab, but also to create a platform with different dynamics to BLUE, in order to best test the generalisability of any control routines. miniBLUE is also designed to have non-backdriveable joints, so that the springs in the joints can be loaded without the need for the drive motors to apply force. This has a direct effect on the electrical energy usage of the robot and so provides another contrast for analysis.

Other issues noticed whilst working with BLUE are addressed in miniBLUE. Specifically, there is increased friction on the leadscrews of BLUE when under loading, meaning that in practice the stiffness of a joint cannot be changed whilst the joint is significantly loaded. The variable stiffness mechanisms of miniBLUE are designed to avoid this problem, and in general to use a more highly specified linear adjustment mechanism. The construction methods utilised for miniBLUE - primarily SLS 3D printing - were also designed to minimise the amount of time required for construction.

miniBLUE is also designed to be inherently modular, with the variable stiffness units being provided on distinct "pods" on the side of each joint. As well as simplifying the construction, this also allows the dynamics of each joint to be more easily

| Measurement | Human | Robot |
|---|---|---|
| Hip (Trochanteric) height | 927 | 463.5 |
| Knee height, midpatella | 504 | 252 |
| Foot length | 269 | 134.5 |
| Ball of foot length | 195 | 97.5 |

Table 7: Sizing data and corresponding $\frac{1}{2}$ scale sizes used in miniBLUE. All dimensions in mm.

altered, or even for an entirely different variable stiffness mechanism (or rigid transmission) to be installed on the joints.

miniBLUE is therefore a smaller, lighter, and more mobile bipedal robot than BLUE, but is based around the same type of variable impedance mechanisms. This robot can therefore be used as a useful validation for control routines, as well as providing a route for beginning to expand them into the third dimension. The modularity built in to miniBLUE should also make it useful as a general purpose bipedal research platform.

### 3.3.1 *Specifications and Motor Selection*

miniBLUE is designed to be $\frac{1}{2}$ of the size of an adult male, with link lengths as given in table 7.

#### 3.3.1.1 *Motor Selection*

After a positive experience with Maxon EC45 flat Brushless DC motors, used as the stiffness adjustment motors on BLUE, this range of motors was investigated for use in miniBLUE. The range includes 70W and 30W versions, each of which have an outer diameter of around 45mm. The power density of these motors is very high, with the 70W motor weighing 141g and costing around £75. Maxon also supply motor driver boards suitable for commutating and driving these motors. Of these, the DEC Module 50/5 is a lightweight, compact, barebones drive module suitable for inclusion into custom electronics. The DEC Module 50/5 weighs 9 grams and costs around £50.

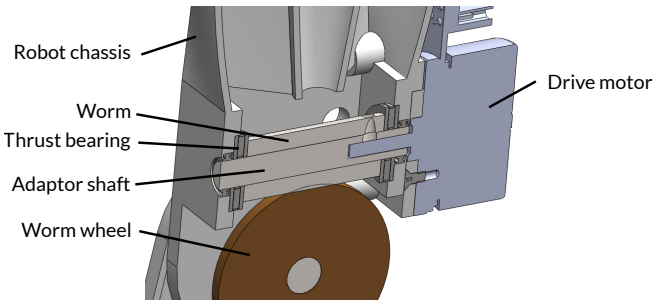



Figure 55: Cross sectional view of the drive motor and worm gearing utilised in the joints of miniBLUE. The drive motor is mounted externally for ease of access, and an adaptor shaft interfaces the motor with the worm gear. The worm gear is sandwiched between two thrust bearings to transfer the load to the chassis of the robot rather than to the motor shaft.

Worm gearing was utilised to gear down the drive motors of miniBLUE, as shown in Fig. 55. A single start worm and wheel with a modulus of 1, achieving a 50:1 reduction weighs around 200g, giving a total weight of 340g for the drive motor and gearing.

For the stiffness adjustment motors, Maxon 24V EC45 flat 30W motors were utilised, driven by DEC 24/3 controller boards. These motors have a mass of 75g, and produce a nominal torque of 54.8mNm at a speed of 2940rpm. A 3:1 reduction is achieved through spur gearing (0.7 mod, weight 20g) before the connection to the lead screw. High precision lead screws from Automotion Components were utilised, with a pitch of 2mm, giving a nominal linear speed of 33mm/s.

### 3.3.2 *Joint Design - The Modular MAwAS*

For miniBLUE, a modular variable stiffness unit was developed, based on the same Modified AwAS mechanism used in BLUE. This unit was kept distinct from the drive motors, and attached to the chassis of the robot, to make it as easy as possible to replace. This allows the specifics of the variable stiffness transmission to be more easily varied, or even for an entirely different series elastic element to be used. Alternatively, the variable stiffness "pod" can be replaced with a link with a high rigidity or fixed stiffness. In this way, miniBLUE can be used as a multipurpose platform for evaluating different joint mechanisms.

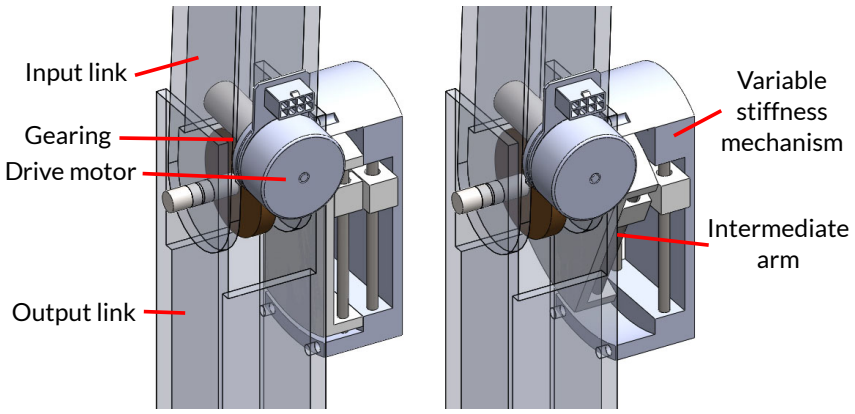
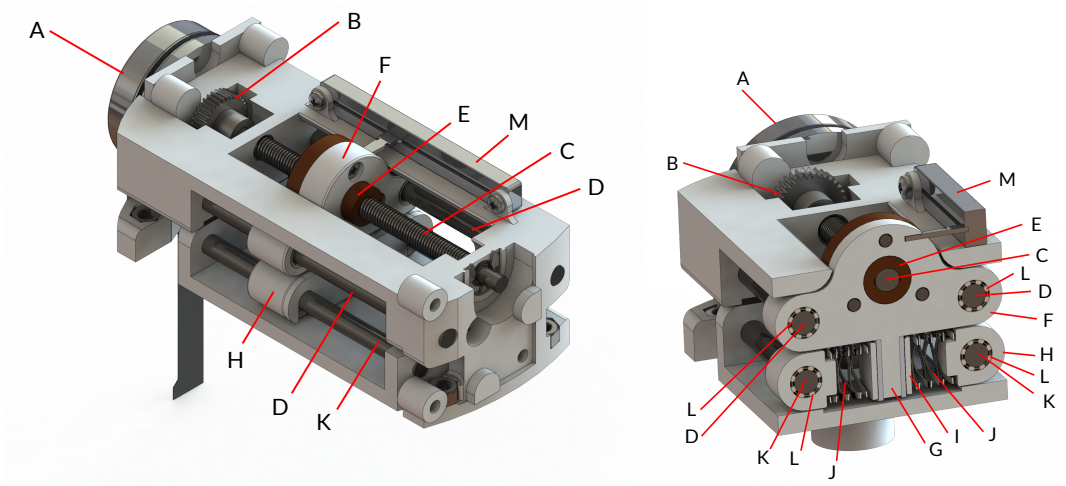



Figure 56: An early concept render of the modular variable stiffness concept used in miniBLUE. The drive motor and gearing are central, and the variable stiffness mechanism attaches to the side of the leg.

Fig. 56 shows an early image of the concept of the modular mechanism. Initially, a "pod" was designed which incorporated the MAwAS mechanism - an intermediate arm attaches to the drive shaft - the output from the worm gearing. This intermediate arm carries a spring carriage with a pair of compression springs, which transfer drive to the output link - the outer part of the pod. As before, the distance of the spring carriage from the axis of rotation of the joint is varied in order to change the stiffness of the joint.

Fig. 57 shows a more detailed render of the MAwAS pod designed for miniBLUE. The stiffness adjustment motor (A) can be seen at the far end, connected via spur gearing (B) to the leadscrew (C). The leadscrew (C) is flanked by two linear shafts (D), which carry the loading on the joint. The round leadscrew nut (E) is incorporated into the design, it slots into the output slide (F) and is attached by three screws.

As best seen in the cross-sectional view of Fig. 57b, a central pivot (G) descends from the output slide (F) and into the spring carriage (H). A block (I) sits on this pivot, between the pair of compression springs (J) which provide the driving force. The block (I) is encased within the spring carriage (H), such that movement of the output slide (F), effected by the lead screw (C), causes the spring carriage (H) to move also. The spring carriage (H) and the output slide (F) transfer load to their linear slides (K, D) through linear bushings (K) made of iglidur self-lubricating plastic.




(a) Full assembly (b) Cross-section

Figure 57: Renders of the modular MAwAS "pod" used in miniBLUE. The components are as follows: A) Stiffness adjustment motor B) Spur gearing C) Leadscrew D) Output link linear slides E) Leadscrew nut F) Output slide G) Central pivot H) Spring carriage I) Spring block J) Compression springs K) Intermediate link linear slides L) Linear bushings M) Potentiometer

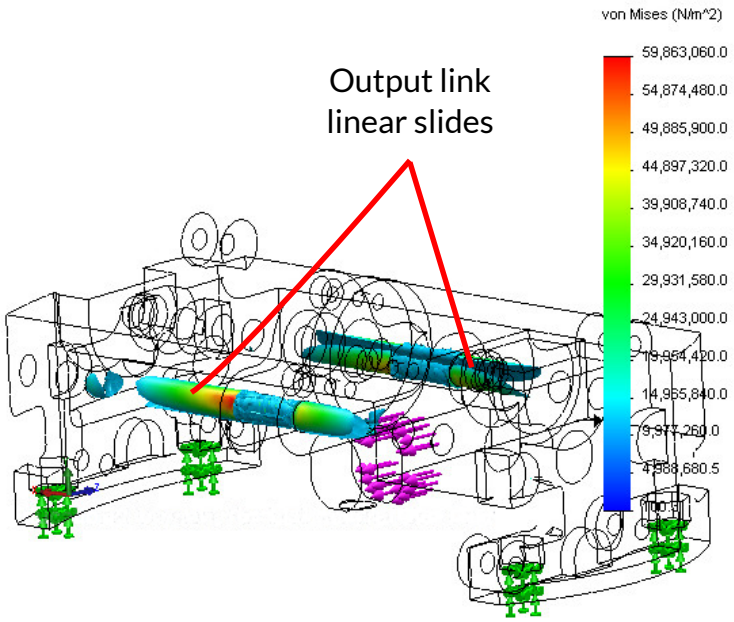



Figure 58: Snapshot of one of the FEA analyses performed during the design of the MAwAS pods for miniBLUE.

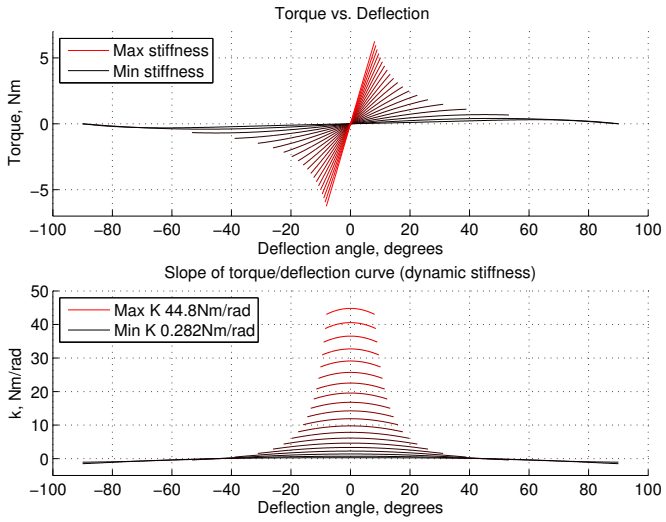



Figure 59: Theoretical Torque and stiffness curves for the variable stiffness mechanism in miniBLUE, for a particular pair of springs

A linear potentiometer (M) is attached to the outer surface of the pod, and the arm of the potentiometer is inserted into a corresponding slot in the output slide (F). This provides the positional feedback for the stiffness control system.

The design of the pod also allows for the spring carriage to be moved until the springs are directly over the axis of rotation of the joint - allowing the joint to become completely free rotating. Example torque and stiffness graphs for the implemented variable stiffness mechanism are shown in Fig. 59.

Fig. 60 shows a full joint of miniBLUE. Like BLUE, miniBLUE utilises geared DC motors connected in parallel to realise variable damping. Since the chassis of miniBLUE is 3D printed, the gearing for the damping can be directly integrated into the parts, as shown here. The gearing utilised is 0.7 modulus, which is can be printed accurately by the SLS printing process used for miniBLUE's parts, and which is strong enough to carry the required damping loading.

### 3.3.3 *Mechanical Design*

Fig. 61 shows the development and design evolution of miniBLUE, alongside the final design, which is shown in Figs. 61e and 61f. From the beginning, the design concept of variable stiffness pods attached to the side of each joint has remained unchanged.

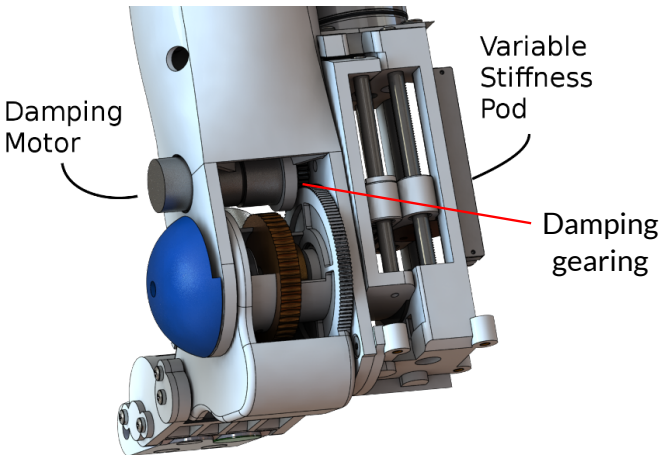



Figure 60: An assembled joint of miniBLUE, including variable stiffness pod and damping motor connected in parallel

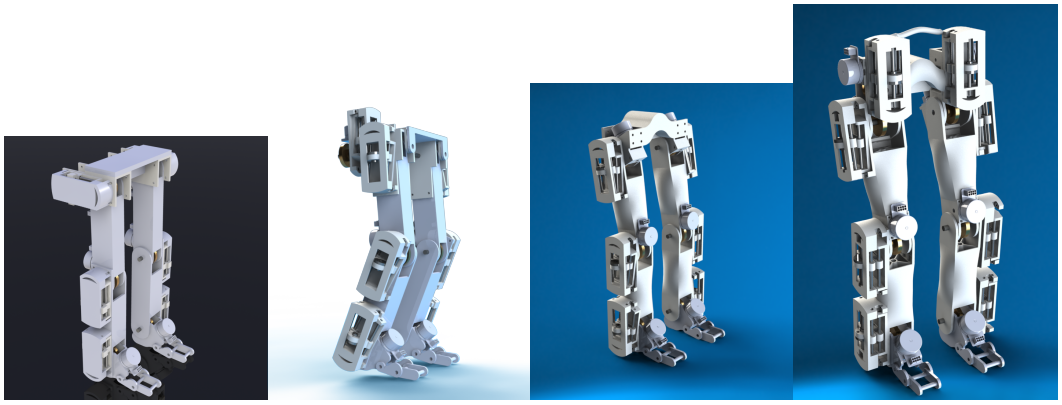
The size of these pods presented the biggest headache of the design, especially for the hip joints, which each have two degrees of freedom. Eventually, to keep the design relatively compact, a small separation had to be introduced between the axes of the two hip rotations. In the final design, the hip ab/adduction axis is located 60mm above the flexion/extension axis.

The use of 3D printing allowed the chassis of the robot to be kept very lightweight whilst retaining enough strength to support the robot. As shown in Fig. 62, the parts have a low density, but are shaped so as to be very strong. In the tibia part, a strong central core extends down the middle of the part, with arced fins supporting the superstructure. Space was made for electronics between this central core and the superstructure, particularly at the front of the robot.

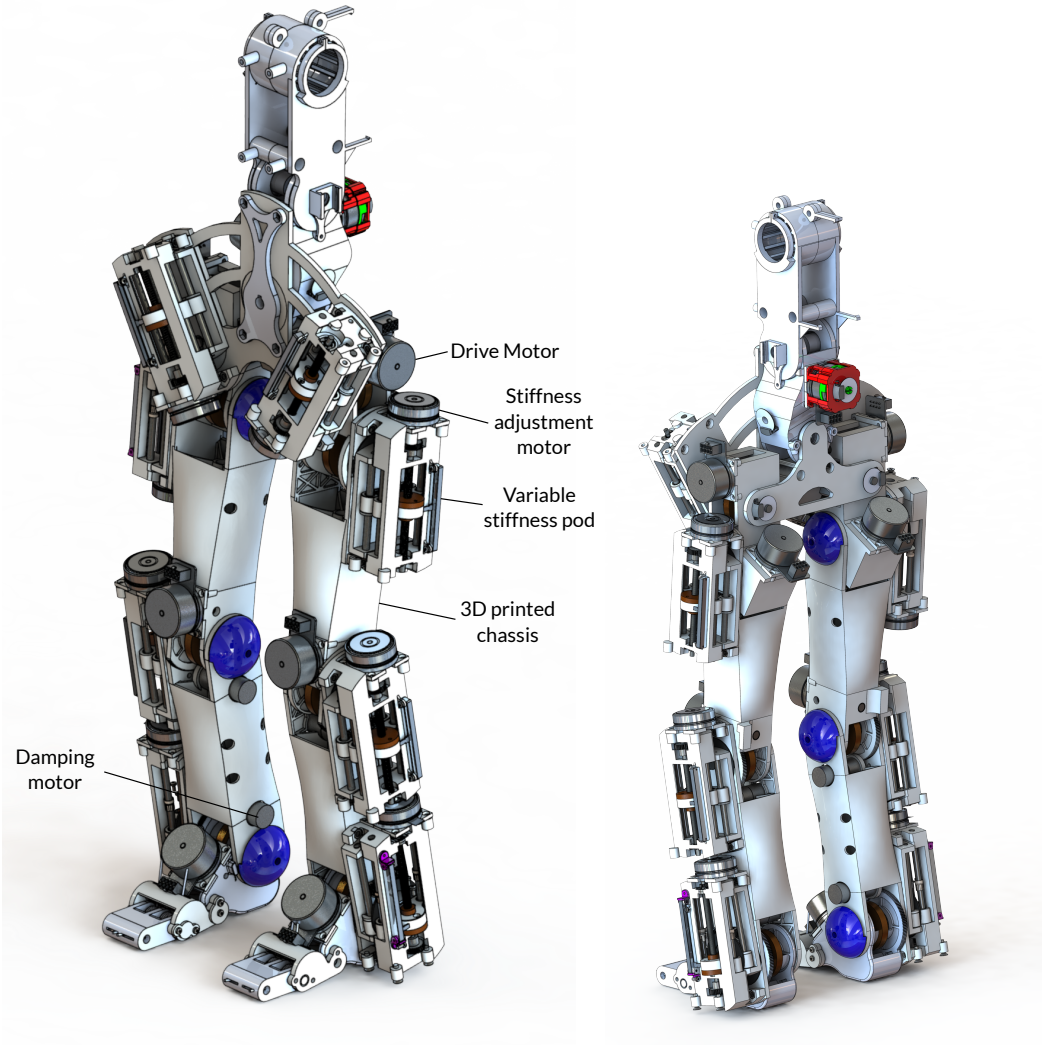
In this robot, positional feedback is provided by optical encoders, which fit around the drive shaft. One encoder is mounted inside the joint, next to the worm wheel, and the second encoder mounts externally. A cover (shown in blue on the renders in Fig. 61) is installed around each external encoder to protect it from damage.

### 3.3.4 *Completed Robot*

Fig. 63 shows the completed robot in its harness. miniBLUE has been powered up, and has been made to produce motion from human walking data. The stiffness adjusters function well, and the robot can adjust stiffness under loading more easily



(a) (b) (c) (d)




(e) (f)

Figure 61: The evolution of miniBLUE

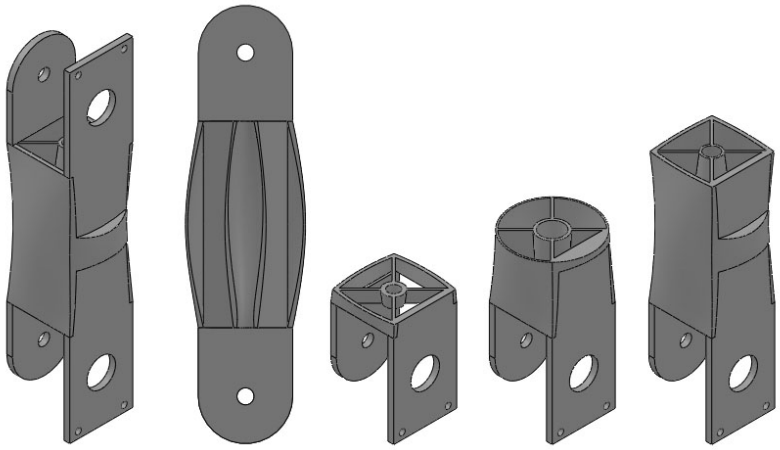
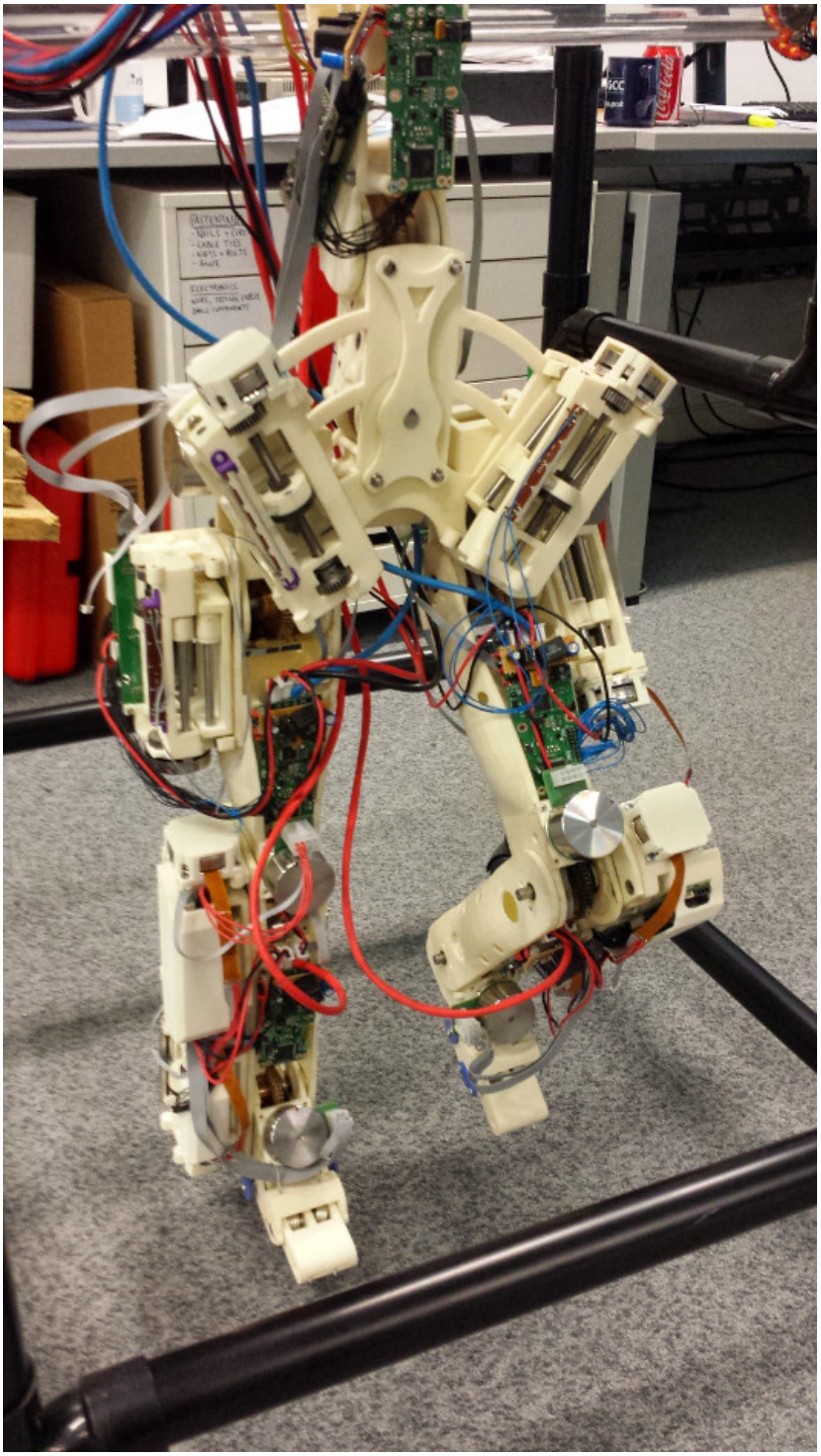


Figure 62: Cross sectional view of the tibia part of miniBLUE

than BLUE can. However, some issues with the tolerances of the 3D printing production method used meant that the mesh between the worm wheel and worm in the main drive gearing is too tight for all but one of the joints. Even with grease, this leads to large amounts of friction, and precludes the smooth running of the robot. This problem could be solved in the future either by reprinting parts of the robot, or potentially by using a worm wheel with 49 rather than 50 teeth, although this might lead to a loose mesh.

## 3.4 SIMULATION AND VALIDATION

To evaluate robot designs, and the potential performance of a movement algorithm before trying it on the actual robot, it is necessary to utilise dynamic simulations. We first implemented a model of BLUE in Choreonoid, and used this to evaluate the design, and later to perform validation of the hardware. Later, we also developed a simulation of BLUE, complete with boom arm, in V-REP. This V-REP simulation was used for the majority of simulations, since V-REP provides multiple easy ways to interact with and control the dynamic simulation, including by using ROS.



(a)

Figure 63: The completed robot miniBLUE

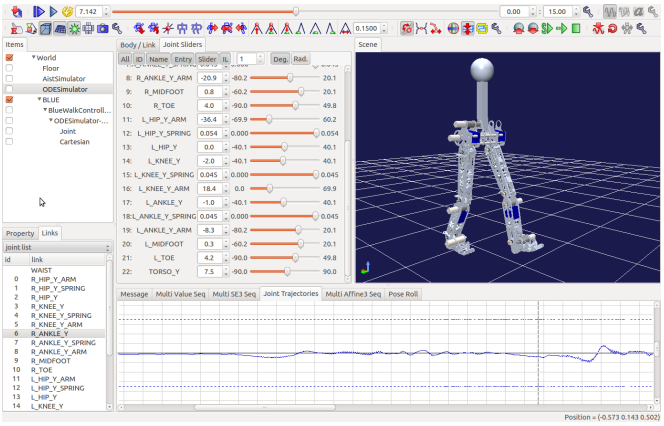


Figure 64: Choreonoid simulation of BLUE

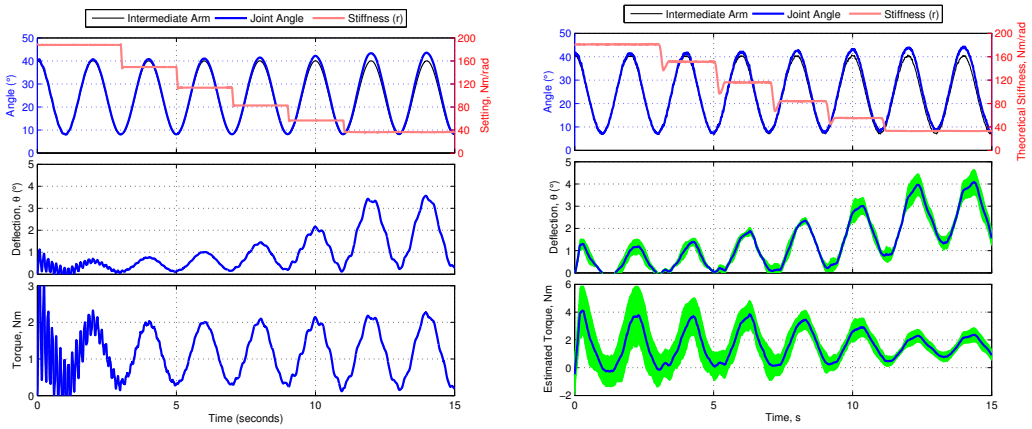
### 3.4.1 *Choreonoid Simulation and Validation*

The Chorenoid software is developed by Shin'ichiro Nakaoka, who altered the program to be capable of simulating only in two dimensions. For the initial simulations, BLUE was simulated using this 2D capability, eliminating the need to also simulate the boom arm. Choreonoid utilises a simple text format for representing models, in which the connectivity, dimensions, mass properties, and graphics files for the robot are specified.

The data for this file was extracted from the SolidWorks models of the robot, ensuring that the inertia properties were extracted from the appropriate point and with the appropriate orientation. The model file was built for the robot, with the connectivity for each variable impedance joint being *input link* $\rightarrow$ *intermediate arm* $\rightarrow$ *output link*. Revolute joints were defined between each of these three components, with a motor connecting the input link and intermediate arm, and a spring connecting the intermediate arm and output link.

Choreonoid allows the definition of a spring function between two links, such that at each pass through the dynamics engine, the torque between the joints is set by this function. For the model of BLUE, the spring function was set according to the theoretically determined equations for the modified AwAS variable stiffness joint, and depends on the position of the prismatic joint which sets stiffness.

To check the performance of the hardware robot against the dynamic simulation, a repeated squatting behaviour was effected whilst the stiffness was changed. The same




(a) Simulation (b) Hardware

Figure 65: Squatting whilst changing stiffness: Knee. For the hardware, error plots are shown in green behind the average trace.

motion was played in the Choreonoid simulation and on the hardware, as shown in Fig. 65. This figure shows that the intermediate arm for each joint accurately achieves the commanded sinusoidal trajectory in both simulation and on the hardware, and that as stiffness decreases, the deflection of the compliant joint increases, as would be expected.

In Fig. 65, the upper subfigure shows the intermediate arm position, the joint output position, and the joint stiffness. The compliant deflection is the difference between the joint output position and the intermediate arm position, and is shown in more detail in the lower subfigure. During the motion, the stiffness is decreased in steps from the highest stiffness to the lowest stiffness setting. In this case, the stiffness is adjusted as fast as possible, and under the lowest loading conditions. We find that, in practice, it is difficult to adjust the stiffness of the joints under heavy loading, due to the increased friction on the leadscrews.

In Fig. 65, it is shown that compliant deflection of the joint does indeed increase as stiffness is decreased, as was expected. The central plot of Fig. 65b shows the average and standard deviation of the deflection reading from three runs of the experiment. However, the average deflection readings are higher than predicted by the simulation. This is likely caused by backlash that was not modelled in the simulation. Additionally, the angle sensors for deflection have a resolution of 0.33 degrees. For calculating

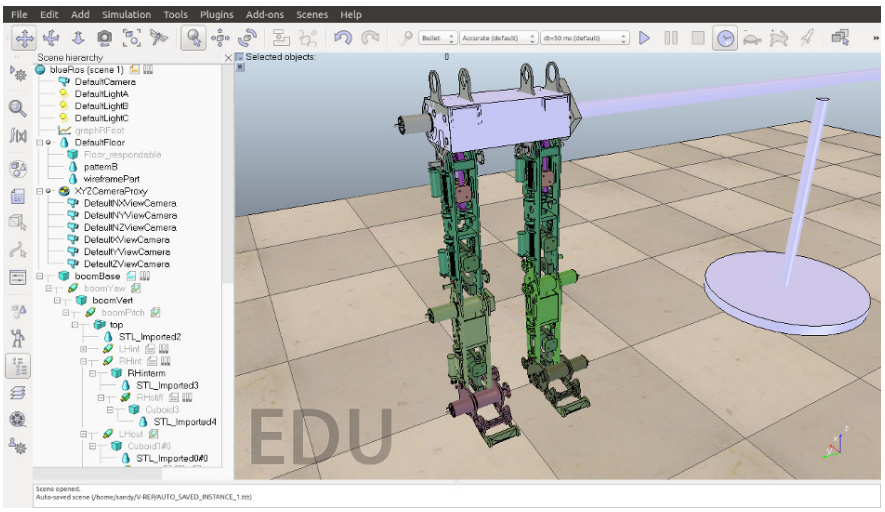



Figure 66: V-REP simulation of BLUE

the torque from the deflection, there is therefore a very large margin of error, especially at high stiffness values, where the 0.33 degrees deflection corresponds to a significant difference in calculated torque. The results show that whilst the robot is definitely capable of varying stiffness, it may not be possible to gain accurate estimations of joint torque.

### 3.4.2 *V-REP simulation*

V-REP is a robot simulator program developed by Coppelia Robotics. As well as being able to simulate a large number of mechanisms and sensors, V-REP also has many options for controlling the simulations, from external scripts to ROS interfaces.

Fig. 66 shows a screenshot of the V-REP simulation of BLUE. The model of BLUE is constructed in V-REP in a hierarchical manner, starting at the base of the boom arm, which is considered as being fixed to the ground. Each segment of the robot has two parts, a primitive shape which assumes its rough dimensions and is set with the mass properties of the real segment, and a linked .stl mesh, which gives the simulation the appearance of BLUE. As with the Choreonoid simulation, the mass parameters are determined from the SolidWorks simulation of the robot.

The intermediate arm segments are driven by motors from the input links, and the output links have a custom callback script to determine the torque on them. These callback scripts are executed by the dynamics engine, and use the torque equations

of the joint, along with the position of the prismatic stiffness adjuster, to determine the torque on the joint.

Each foot is comprised of three parts, as on the real robot. Fixed stiffness springs determine the torque between segments, and hard stops are included to enforce the same joint limits as on the real robot. Contact segments are positioned under each part of the foot, connected by force sensors. There are therefore three force sensors on each foot of the simulation.

The V-REP simulation can be run in real time on a standard laptop, and when it is initiated, it launches an external control script which controls the joints of the robot. This occurs in a similar way to the client scripts of the real robot, and the external script can gather information about the state of the simulation, including data from sensors, as well as set target positions for all the motors. The simulation includes flags for resetting the simulation, such that when the external control script sets a reset flag, the simulation is returned to its original state, and V-REP sets a flag to show that the external script can continue.

The data retrieved from the simulation includes the position of and torque on each joint, the pitch and yaw of the boom arm, and three dimensional force sensing from each of the foot force sensors.

V-REP can also be interfaced with ROS. In this case, when the simulation is started a script is launched which launches a ROS node, and makes services available for interacting with the simulated robot. Alternatively or additionally, the ROS script can subscribe to topics publishing command data, and publish its own state data on a ROS topic.

# 4

# ELECTRONICS & COMMUNICATION

For both robots, we developed a highly modular, robust, high-performance electronics and communication architecture. The system was first developed for use in BLUE, and then adapted slightly for use in miniBLUE, the major difference being the different control boards utilised by miniBLUE, each of which featured two microcontrollers. The designed system utilises a set of control boards based around ATMEL microcontrollers, all of which communicate using a custom protocol built on Ethernet. These boards take care of all low level control and safety procedures, and a FitPC running a stripped down ArchLinux distribution is responsible for mid-level control - e.g. joint trajectories.

For this section of the work, Andrius Sutas, an undergraduate student helped out with several aspects. Andrew did the board layouts for the various circuit boards produced, wrote the code for low level ethernet communication on the control board and PC, wrote the daemon script introduced in section 4.7.1, and numerous other contributions to bits and bobs of the firmware.

## 4.1 REQUIREMENTS AND SYSTEM SPECIFICATION

Each joint of BLUE has three motors (drive motor, stiffness adjustment motor, damping motor), requires position sensing for both the equilibrium position as well as the elastic deflection, stiffness setting, and should have some form of current feedback on at least the drive motor. We require a system which can robustly perform low level control of the joints, and specify a low level control loop frequency of 1kHz. Such a frequency gives a period much faster than the mechanical time constant of the system, and should give a smooth, high-performance behaviour. It is important that the frequency of the control loop is maintained with high precision, or else the behaviour of the closed-loop controllers could become unpredictable. For this reason, it was decided to use dedicated microcontrollers to perform the low-level control.

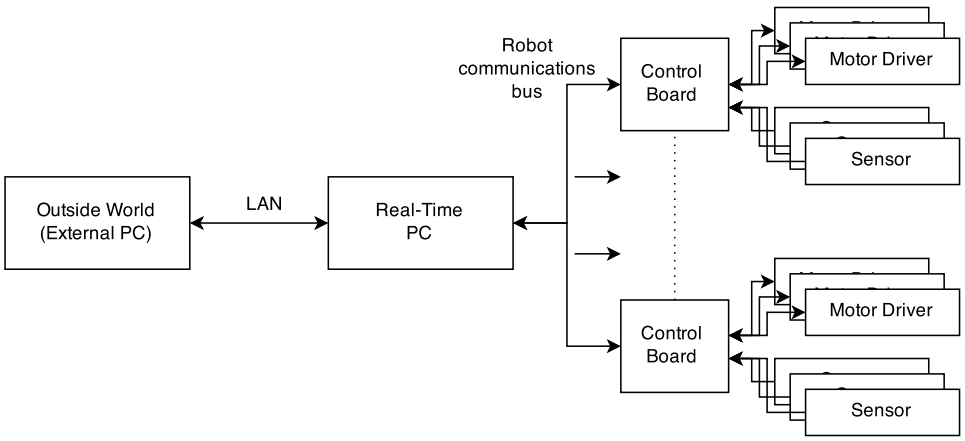



Figure 67: High-level schematic of the electronics & communication architecture

The electronics architecture is designed to be highly modular, in order to give the best flexibility, and to allow parts to be quickly swapped in and out. We utilise a series of microcontroller based control boards, which will take care of the lowest level of control. Each control board can interface with a number of motor driver boards. These motor driver boards are kept distinct from the control boards, so that a motor driver suitable for a specific motor can be used, and can easily be changed should a different motor be selected or should a driver board fail.

A central communications bus connects the control boards, allowing the easy addition or removal of a control board. Whilst the control boards are responsible for low level control, a computer is connected to the communications bus to give commands to the control boards in real time. This computer, in turn, connects the robot to the outside world, and can be sent high-level commands.

An overall view of the system is shown in Fig. 67. From this, it was necessary to select or define the communication bus and protocol, the connectivity of the control boards, and the number and type of sensors.

BLUE has six joints. With three motors and at least four sensors per joint, the robot contains 18 motors and at least 24 sensors. If a two-byte value is sent for each motor, and a two-byte value received for each sensor, this would represent a total of at least 84 bytes to set every motor and read every sensor, ignoring communication overheads. In practice, there may be additional status information to be read from the control boards. If this amount of data were to be sent every 1ms, the overall data rate would

be 672,000 bit/s, wherein 288,000 bit/s is sent to the control boards, and 384,000 bit/s is received from them. If communication overheads and additional information is factored in, the total bandwidth of the system can reasonably be expected to reach 1mbit/s. In practice, it is unlikely that the motor positions would be set at such a high rate, but the system should be capable of such behaviour, and should certainly be able to return sensor readings at this rate.

We therefore required a two way communications bus capable of speeds exceeding 1mbit/s. The communication bus was also be routed around large motors, and hence needs to be robust against noise. The communication system should also be multi master in order to give the greatest amount of flexibility - in addition to communications initiated by the real-time PC, it can be useful to have frames initiated by control boards. For example, if one board encounters an emergency stop condition, it can immediately signal this to the other control boards and the real-time PC, allowing for the whole robot to take an appropriate action without requiring interaction from the controller PC. Additionally, the ability to have multi-master control enables functionality such as low-level reflexive loops, which could be useful if a neurologically inspired controller were to be developed for the robot.

## 4.2 SELECTING A COMMUNICATIONS INTERFACE

There are numerous possible communications buses commonly used for robotic systems, and supported by many microcontrollers. A selection of those most likely to be useful is briefly explained and evaluated below.

### 4.2.1 *Common Communication Technologies*

#### $I^2C$ / SMBUS

$I^2C$ (Inter-Inter Circuit), or its extension SMBus (System Management Bus) are supported in hardware by a large number of micro-controllers. It is a relatively simple communications interface which uses two lines - data, clock, in addition to +ve and ground. The data and clock lines are bidirectional, are pulled high in their default state, and are asserted low by the microcontroller. The bus has defined start and stop

bits, and each frame begins with a master device sending such a start bit, followed by the address of the slave device.

$I^2C$ is a multi-master protocol with built in procedures for arbitration between masters, and the ability to broadcast to all devices. However, the protocol is not very robust to noise, as it utilises standard open-drain data and clock lines at either 3.3v or 5v. The data rate of standard $I^2C$ can be up to 100kbit/s, although higher speed systems exist, with some microcontrollers capable of 400kbit/s or even higher.

### SPI

SPI (Serial Peripheral Interface) is a synchronous serial interface which allows the daisy chaining of slaves and their selection via dedicated slave-select lines. It is a single master, full-duplex protocol with four lines - clock, data out, data in, slave select, in addition to power and ground. The maximum data rate is not specified by the protocol, and thus can often be up to the clock frequency of the microcontroller. In practice, higher data rates can be difficult to achieve robustly, and the system is quite vulnerable to noise.

The need to have a separate slave-select wire for each slave device severely comprises the modularity of an architecture built on the SPI bus. It is possible to use a type of SPI such as mSPI (miniSPI) to use multiple slaves on the same slave-select line, and specifically address the slave in the first byte of the communication frame.

### CAN BUS

CAN bus (Controller Area Network Bus) is a multi-master bus developed specifically for use in automotive applications. Because of this, it is designed to be robust against noise, and to operate over long cable lengths. The buses uses only two wires in a twisted pair, with no clock signal. The protocol has lossless arbitration, and includes priority setting, addressing, CRC (cyclic redundancy check) for error detection, and acknowledgement. CAN bus does however limit the length of frames to eight bytes.

The maximum speed of CAN Bus is 1mbit/s, although the achievable speed decreases as distance increases.

### ETHERNET

Ethernet is the name given to a family of networking technologies which have become the standard for networked computers. Specific implementations, for example IEEE 802.3 - 100BASE-TX, define the physical and data link layers of the network. Mentions of Ethernet from now on will refer to 100BASE-TX, unless otherwise specified. Ethernet utilises two twisted pairs of cables, and allows full-duplex, multi-master communications at speeds of up to 100mbit/s. For normal networks, a TCP/IP stack is built on top of Ethernet, above the data-link layer, but this is not required for Ethernet.

Ethernet frames include CRCs for error detection, and the protocol and associated hardware automatically account for collisions and corrupted data - resending as appropriate. Whilst it can be (almost) guaranteed that an Ethernet frame will be received correctly, it cannot be guaranteed that frames will be received in the order they were intended to be sent. Ethernet frames can be from 72 - 1526 bytes in length, including addresses and other overheads.

Ethernet is therefore a high-bandwidth, multi-master protocol with good robustness to noise. Ethernet connectivity is commonly available on microcontrollers, although may require an additional physical interface chip.

### ETHERCAT

EtherCAT (Ethernet for Control Automation Technology) is a protocol developed for precise, low jitter control of automation applications. Unlike standard IEEE 802.3 Ethernet described above, EtherCAT networks have a single master which initiates frames, and these frames are passed through slave devices, being interpreted or interacted with on the fly, as the frame passes through. EtherCAT includes a system for measuring the delay associated with each slave device, and calibrating clocks such that commands can be issued with a precise timing.

While a software implementation of EtherCAT is possible for a master computer with a common 100BASE-TX Ethernet controller, it is necessary for each EtherCAT slave device to have EtherCAT hardware in order to properly pass and process the packet. Such EtherCAT controllers are much less common on microcontrollers. Using full duplex, EtherCAT can utilise almost all of the 2x100 mbit/s bandwidth available

| | Bandwidth, mbit/s | Multi-master | Common hardware | Robust to noise |
|---|---|---|---|---|
| $I^2C$ / SMBus | 0.4 | • | • | - |
| SPI | 40+ | - | • | - |
| CAN Bus | 1 | • | • | • |
| Ethernet | 100+ | • | • | • |
| EtherCAT | 100+ | - | - | • |

Table 8: Potential communication technologies

with 100BASE-TX. EtherCAT is specifically designed to have short cycle times, and network update rates can exceed 30kHz.

EtherCAT is therefore a high precision, high bandwidth, single master protocol, which requires specialist hardware in the slave devices.

### 4.2.2 *Evaluation and Selection of a Communication Technology*

A summary of the communication protocols described above is presented in Table 8, with regard to the most important criteria for application in BLUE and miniBLUE.

EtherCAT, whilst being high performance and giving precise control of timing, requires specialist hardware and does not allow for multi-master systems. Whilst it is possible for slaves to communicate with each other using EtherCAT, the communication must be initiated by the master.

CANBus fulfils many of the requirements, but with a maximum data rate of 1mbit/s, falls too close to our requirement of 1mbit/s, as in practice the communication system cannot be used at 100%. Furthermore, CANBus places restrictions on the size of the packets which can be sent, which may be prohibitive to the flexibility of the constructed system.

Standard 100BASE-TX Ethernet provides the best platform for communications on the robot, it has a high enough speed to cope with the required data throughput. Even though collisions can occur, and thus the timing is *de jure* non-deterministic, in practice the low network utilisation should mitigate this. If necessary, a protocol for precise timing can be implemented on top of Ethernet, for example utilising broadcast messages. On the hardware side, many affordable microcontrollers include hardware Ethernet controllers, which work in tandem with commonly available physical transceiver chips.

For our control systems, we therefore utilised Ethernet for the physical and data-link layers, and implemented a custom protocol on this to enable high-performance control of the robot.

## 4.3 SENSORS

Both robots require a substantial number of sensors. Some sensors are required for each joint - for example measuring joint position, deflection, current, stiffness position, etc. Other sensors are required in particular places, for example for measuring ground reaction forces, or for inertial measurement.

### *Rotary Sensors*

Rotary sensors are used to measure the drive motor output position, and the deflection of the elastic elements of the joint. This means that at least two rotary sensors are required for each joint in each robot.

Various types of rotary sensors are available, each with particular advantages and disadvantages.

**Rotary Potentiometers** are the most simple type of rotary sensor, with a track and wiper system which varies the observed resistance with changes in position, to give an absolute output value. Typically, the potentiometer is used as a voltage divider, and the wiper voltage fed into an ADC on a microcontroller. The most common type of rotary potentiometer has a movement range of less than a full rotation, although multiple turn potentiometers are available, as are potentiometers which wrap around,

albeit with a dead zone. As a purely analogue device, potentiometers are subject to noise, especially when near high powered motors.

The available resolution depends on the resolution of the ADC into which the signal is provided. For example, in a system with a 10-bit ADC, and a linear 0-5v voltage range over 300 ° of rotation, the resolution is 0.3°. In such a system, the ADC provides an output value which varies with 4.9mV/step, this means that even small amounts of noise can affect the measured position. Most potentiometers are provided with shafts to be rotated, although a small number of through-shaft potentiometers exist.

**Optical encoders** are often used in high-performance applications, as they are inherently insensitive to electrical noise. Absolute optical encoders are available, but very expensive, and hence incremental encoders with index marks are far more common. Even so, a typical 3-channel incremental optical encoder with 500 counts per revolution may still cost £40. Such an encoder provides a quadrature signal which gives information on the direction of rotation of the sensor, and gives an effective resolution of 2,000 steps/revolution.

Optical encoder codewheels are typically provided with bores, ready to be mounted on shafts, and require a precise locational arrangement with the encoder itself. In addition to the expense of such devices, incremental encoders also require that the codewheel is moved through its index position, in order to obtain an absolute position reading. After this, every signal change must be logged to keep to keep the absolute position accurate.

**Magnetic rotary encoders** (as distinct from resolvers) use the changes in a magnetic field to give positional information. In one form, a diametric magnet may be positioned above a magnetic encoder chip, and the chip will then accurately determine the rotation of the magnet. Various resolutions of encoders are possible, from 10 bit to 16 bit or above, at a relatively low cost.

Chips such as the Austria Microsystems AS5040-ASSU provide multiple potential outputs, include a PWM signal, analogue signal, or synchronous serial interface (SSI). If such a digital interface is used, the position reading is very robust to electrical noise. Magnetic noise from the motors is possible, but since the magnet is positioned very close to the encoder chip, in practice little-to-no interference is observed.

Other types of rotary sensors, such as resolvers, resistive or capacitive rotation sensors, will not be detailed here.

**Measuring multiple rotations** is an issue with all of the above described position sensors, with the exception of multiple-turn potentiometers, which suffer from very low resolution. During the course of this work, a solution to this problem was found, which utilises two wrap-around absolute position sensors to give a high-resolution absolute measurement over multiple revolutions. This method works by requiring a gear ratio of $n : o$ between the two encoders, wherein $n$ is not an integer multiple of $o$, and $n$ is greater than $o$. This allows the first and second encoders to produce a combined reading which repeats only every $m/n$ revolutions, where $m$ is the lowest common multiple of $n$ and $o$.

Such a method is useful in situations where it is not feasible or desired to have a sensor measuring the absolute final output position. For example, it can allow the relocation of sensors further up the gear train, or could measure the position of a turntable being driven by a motor. Alternatively, such a sensor could be used to determine the absolute position of a linear actuator, such as a leadscrew, without the need to drive the leadscrew into a calibrating stop. Whilst useful, this type of sensor was not utilised in BLUE, and further information on the system can be found in Appendix. A

#### SELECTING ROTARY SENSORS

Due to issues of noise and robustness, resistive potentiometers are not used in either robot for rotary position sensing. Instead, in BLUE, two Austria Microsystems AS5040-ASSU 10-bit magnetic encoders are used for each joint. These are read through a SSI, which includes error flags and a parity check. The encoders can support clock frequencies of up to 1 MHz, meaning that the data can be read at high speed from each chip.

Each encoder gives a 10-bit reading over a full rotation, giving a resolution of 0.35° for both the motor output and the elastic deflection.

In miniBLUE, it was desired to have more accurate position sensing, and each joint comprised two Avago AEDB-9140-A13 3-channel, 500 CPR incremental optical encoders. These provide 2000 steps per revolution, giving a resolution of 0.18° for each encoder.

Later, when the torso joints for miniBLUE were developed, it was decided to utilise a combination of optical encoder and magnetic encoder to give the position of these. The optical encoder was positioned on the rear of the Maxon EC45 flat drive motor, and the absolute magnetic encoder is positioned at the output shaft. In this arrangement, an absolute position reading can be given immediately on startup from the magnetic encoder, without the need to rotate the output joint through the index position. This is convenient since miniBLUE utilises worm gearing, and hence cannot be backdriven by hand.

Furthermore, the incremental encoder rotates at 50 times the speed of the output joint, giving a theoretical resolution of 0.0036°, although the backlash of the gearing must also be considered. This arrangement gives a highly accurate positional reading and an instantaneous absolute reading with 10-bit accuracy, and would make velocity control of the joint more reliable as a smoother derivative is obtained from the motor position signal.

*Current Sensing*

To ensure the motors of the robot are not damaged, it is necessary to sense the current each motor is using. Additionally, this current data can be used in calculations for energy usage, and as an approximation to motor torque. In both robots, current sensors are used which give analogue output values, which are supplied to the ADC of the joint's control board, where they are digitally filtered. The current sensors are included in the motor driver board, and hence not on the control board itself.

For BLUE's drive motors, the LMD18200T H-bridge chips utilised have a current sense output, which outputs 377μA/A. This is converted to a voltage on the driver boards, for input to the ADC on the control boards. On miniBLUE, the drive motors each have a Diodes Incorporated ZXCT1109 current sensing chip, giving a resolution of 10mA on the control boards.

*Linear Position Sensing*

The stiffness adjusters on both BLUE and miniBLUE function by linearly moving a pair of compression springs. In order to obtain feedback on the present stiffness

setting, it is necessary to measure this linear position. Whilst this could be achieved with the multiple-turn rotary sensor detailed above, for this application it suffices to use a linear potentiometer. For example, the Bourns PTA6043-2015DP-B103 is a 10K linear slide potentiometer, and is used on miniBLUE. It has a 60mm travel, giving a resolution of 0.059mm. The noise on the analogue signal is ameliorated using digital low pass filtering on the control board.

*Torque Sensing*

Since the joints of the robot have compliant elements, it is possible to measure the torque on the joints by measuring the deflection of the complaint elements. Using the formulas defined for the modified AwAS variable stiffness joint, along with the observed stiffness setting, $r$, and deflection from equilibrium, $\theta$, the torque on the joint can be calculated. It is also possible to use the motor current from the drive motors of BLUE as another source of information on joint torque, as the current used by the motor should be proportional to the torque in the joint.

*Ground Reaction Forces*

The control boards are designed with spare ADC and GPIO ports to enable the connection of additional sensors. In the initial design, force sensitive resistors were to be used to give information on the ground reaction forces. Two FSRs would have been placed on each section of the three part foot, as part of a potential divider. However, FSRs have a highly non-linear response, exhibit large amounts of hysteresis, and are sensitive to the manner in which they are loaded.

Instead, a redesigned foot plate allows the mounting of strain gauges of the type used in electronic scales. This change will be made to BLUE shortly, using CZL928F micro-load cells by Dongguan South China Sea Electronic Co Ltd..

| Sensor | Type | BLUE | miniBLUE |
|---|---|---|---|
| Joint angle & Intermediate arm angle | Rotary position | Magnetic encoders (Austria Microsystems AS5040-ASSU) | Optical Encoder (Avago AEDB-9140-A13) |
| Stiffness setting | Linear position | Linear potentiometer (Bourns PTA6043-2015DP-B103) | |
| Joint torque | Calculated | Virtual | |
| Drive motor current | Analog readout | From H-Bridge (LMD18200T) | Diodes Incorporated ZXCT1109 |
| Ground reaction force | Force magnitude and distribution | Micro load cell (3 per foot) | ... |
| Vestibular | Inertial measurement and tilt | Connectivity for IMU (accelerometer & gyros) | |

Table 9: Summary of the sensors on BLUE and miniBLUE

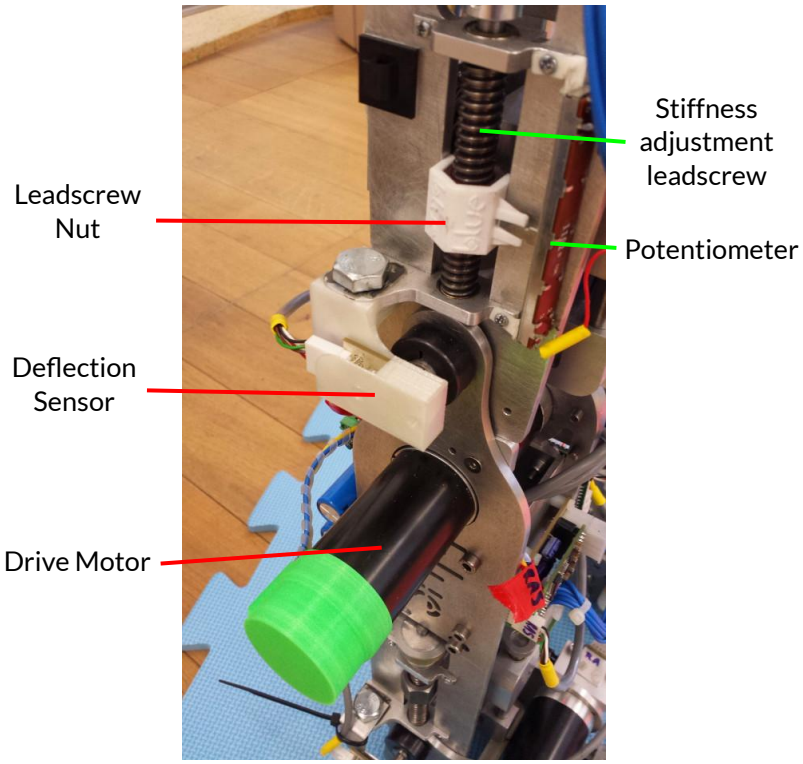



Figure 68: Knee of BLUE with deflection and stiffness sensors indicated

### 4.3.1 *Summary of Sensors*

BLUE and miniBLUE have a very similar electronics architecture, but some differences in the sensors. A summary of the sensors on each robot is given in Table 9. As mentioned above, certain joints of miniBLUE utilise an incremental optical encoder in tandem with an absolute magnetic encoder.

## 4.4 CONTROL BOARDS

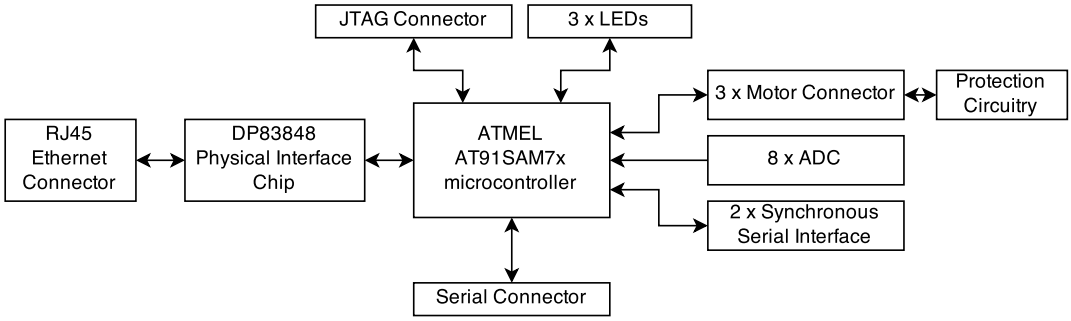
The control boards for BLUE and miniBLUE are different, but based on the same architecture, and both based around an ATMEL AT91SAM7x256 microcontroller. Both boards have the same Ethernet physical interface hardware, and both can be programmed by JTAG. The control boards for BLUE are designed to each control one joint, whilst the control boards for miniBLUE control two joints each. Additionally, the control boards for miniBLUE have an additional microcontroller, an ATMEGA164A, which monitors the optical encoders and provides additional ADC and I/O ports.

### 4.4.1 *BLUE Control Boards*

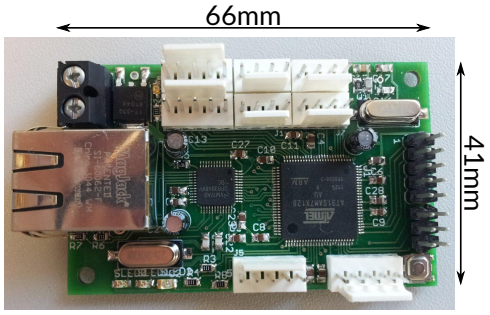
Fig. 69a shows a schematic of the control boards used in BLUE. At the core of each board is an ATMEL AT91SAM7x256 microcontroller, running at 48MHz. This chip interfaces with a National Semiconductor DP83848 Ethernet physical layer transceiver chip to provide the Ethernet connection on the board. A Stewart Connector MagJack module provides the physical RJ45 port on the board, to which a standard Ethernet cable can be attached.

Each board provides the following connections:

- 1 x RJ45 Ethernet connector
- 1 x JTAG connector for programming
- 1 x Serial output for debugging and communications
- 3 x 5 pin motor connector (PWM, Direction, Brake, Fault, Ground)




(a) Schematic of BLUE control board




(b) Example of control board

Figure 69: Control Boards for BLUE

- 1 x Dual Magnetic Encoder SSI connector (Ground, Data1, Data2, CSn, Clock), for connecting to two magnetic encoders
- 4 x Dual ADC interface connectors (8 ADC inputs total)

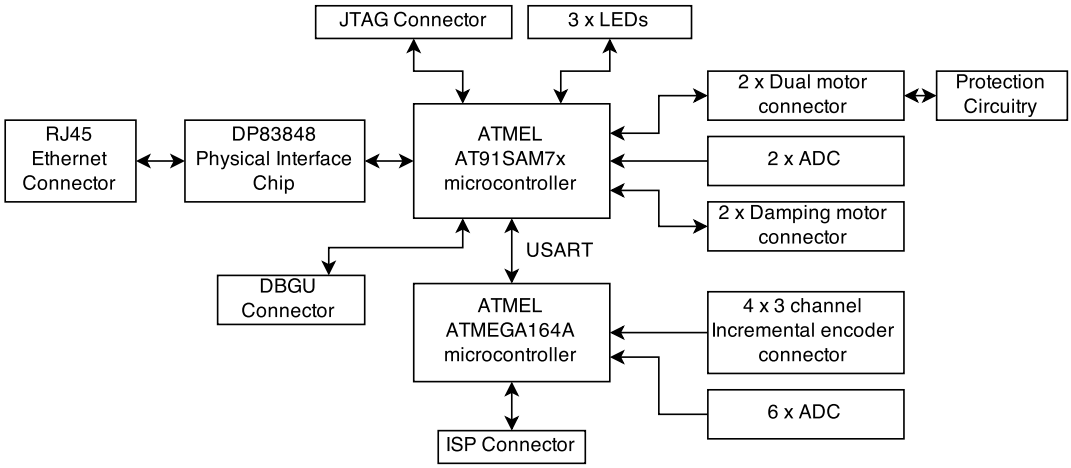
A fabricated control board can be seen in Fig. 69b. The board additionally includes three LEDs, all of which are switchable by the microcontroller. The board includes a voltage regulator for powering the microcontroller. Current and voltage protection circuitry is implemented as a small in-line interface board between the control board and the motor driver. Current feedback is provided to one of the ADC interface connectors, as is the signal from the stiffness sense potentiometer.

The control board reads the signals from two magnetic encoders, each of which has a Synchronous Serial Interface (SSI). Since the control board operates at a significantly higher frequency than the 1MHz maximum frequency of the SSI link, and in order to save processing time and physical pins, the same clock and chip select signals are used for both encoders. At each clock edge, the control board reads and logs the signals from both magnetic encoders, thereby simultaneously reading both encoders. The data read from the encoders includes an error flag and a parity bit, and if an error is detected the encoder will be re-read.

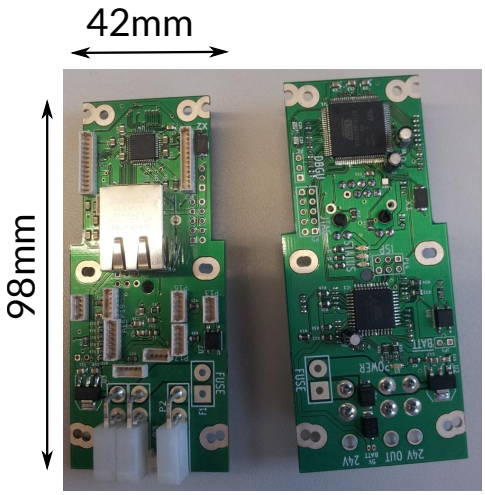
### 4.4.2 *miniBLUE Control Boards*

The control boards for miniBLUE share the same core as those for BLUE, but have expanded functionality. The miniBLUE control boards are expected to read four incremental optical encoders, and it is vital that no pulses from the encoders are missed in order to maintain an accurate position reading. The boards therefore include a second, less powerful and smaller, microcontroller (an ATMEGA164A) which is responsible for reading the optical encoders. A USART serial communications link between the two microcontrollers is used to transfer data.

The miniBLUE control boards also include onboard damping hardware, so that the damping motors can be directly connected. This hardware consists of a pair of ASSR-1510 high current solid state relays, which allow the connected damping motors to be braked as required. The secondary microcontroller on the control boards generate




(a) Schematic of miniBLUE control board




(b) Example of miniBLUE control board

Figure 70: Control Boards for miniBLUE

the PWM signals to modulate damping, and also provide additional ADC and I/O ports.

miniBLUE's control boards utilise smaller, lower profile connectors, and have many more ports, since they control twice as many motors and interface twice as many sensors. Each motor driver board for miniBLUE also drives two motors, and hence each motor connector has pins for controlling two motors.

Each miniBLUE control board has the following ports:

- 1 x RJ45 Ethernet connector
- 1 x JTAG connector for programming main microcontroller
- 1 x ISP connector for programming secondary microcontroller
- 1 x DBGU output for debug info
- 2 x Dual motor driver connector (four motors total), including current sense inputs
- 2 x Damping motor connector
- 4 x 3-channel incremental encoder connector
- 1 x 6-channel ADC interface
- 2 x 1-channel ADC interface, for potentiometers

A fabricated control board can be seen in Fig. 70b. The board is designed to be powered from a 24V supply, in order to decrease the number of cables running down the robot, since the motors are also powered from the same 24 supply. Each board splits the 24V supply to reduce the number of connectors needed in the robot. As with BLUE's control boards, each miniBLUE control board includes three microcontroller switched LEDs for providing status information.

### 4.4.3 *Control firmware*

The same base firmware is used for the main microcontroller on both the BLUE and miniBLUE controller boards. A preprocessor macro is used to set the board type and thereby include any special code only required for that board - for example the serial

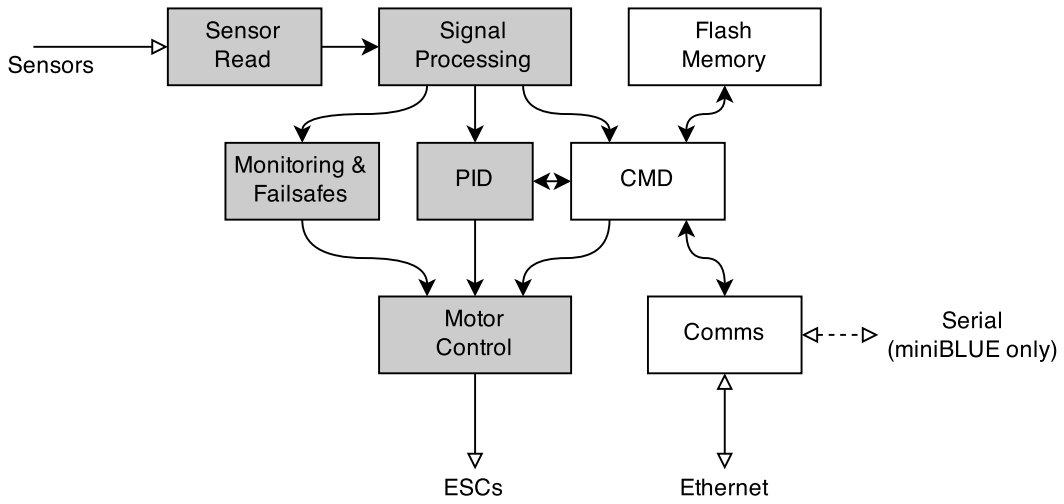



Figure 71: Block diagram of control board firmware

communications code required to communicate with the secondary microcontroller on the miniBLUE board. Setting the board type also includes a definitions file relevant to that board, specifying the number of motors, sensors, etc.

The firmware is modular, such that motor objects are initialised with specified ports, and linked to specified sensor objects. The sensor objects, in turn may be linked to a physical sensor, such as an optical encoder, magnetic encoder, or potentiometer. One sensor object is a 'spoof' sensor, linked to when a sensor does not exist or is not connected, for example when a motor does not have a current sensor. The definitions file for each type of control board specifies the instantiations of motors, sensors, etc. for that board.

A block diagram showing the principle elements of the main microcontroller firmware is shown in Fig. 71. This diagram does not include the initialisation routines, for example where data is retrieved from flash memory and used to initialise controller gains, safety limits, etc. The elements shaded on the block diagram are part of the main control loop, which must be executed at a precise, reliable frequency. The remaining elements - Ethernet communications, command execution - are executed in the remaining time, where the control loop algorithm has finished for that iteration.

Fig. 72 shows a high level representation of the activities of the microcontroller over time. To enforce the strict 1kHz control loop frequency, a timer is set up and

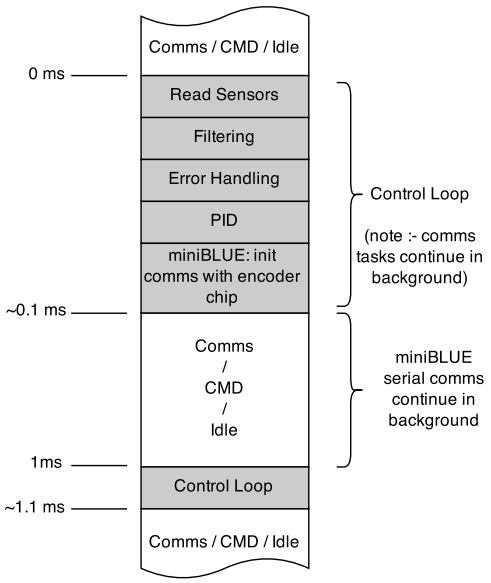



Figure 72: High level timing diagram of control board firmware

set to interrupt every millisecond. [1] The communications hardware operates in the background at all times, but communications are only interpreted, and commands executed, under normal code execution - can be interrupted at any time by the control loop, which takes priority.

### FLASH MEMORY

The Flash memory on the main microcontroller is used to store parameters including:

- board MAC address
- Maximum and minimum safe positions for a joint
- Electrical current safety limits
- Deflection sensor calibration
- Motor polarity information
- PID gains

1 Board master clock frequency is a nominal 48,054,741 Hz. Timer clock is set to MCK/2, and set to interrupt every 24,027 counts. Nominal period is thus 0.9998 ms.

Listing 1: Moving Average Filter

```
runningTotal = runningTotal - history[i] + newInput
history[i] = newInput
i = (i+1)%filterSize
out = runningTotal>>log2(filterSize)
```

- Flash checksum

On power up, this information is read from the flash memory and moved into appropriate places in RAM, to be used during operation. To ensure that the data has been correctly stored and retrieved, a checksum is stored alongside the data, and verified before the configuration data is used. An error is produced if the checksum does not validate. Commands can be sent to change any of the parameters, and when these are executed the variables in RAM will be updated, before the flash memory is updated with new values and a new checksum.

### FILTERING

Two types of filter are implemented in the control board firmware: a moving average filter and a simple type of decaying thresholded integrator. The moving average filter is used to smooth sensor readings, including the position reading and current sense readings. An efficient implementation of the moving average filter was devised, and is summarised in listing 1

Note :- the algorithm in listing 1 assumes the filter size to be $2^n$, for efficiency. Other filter sizes can be used, in which case a normal division operation replaces the right shift. The code is designed to work with unsigned integers, since the raw sensor readings, and the eventual motor commands, are unsigned ints.

To prevent short current spikes tripping a safety limit, a second filter is applied to the low-pass filtered current readings, and used for detecting when a current limit error should be thrown. This filter is a simple type of decaying integrator, which allows prolonged periods of high current usage to be detected. The filter operates as shown in listing 2.

The filter for detecting current past the safety limit is therefore the moving average filter cascaded into the integrating filter, and behaves as can be seen in Fig. 73. If the

Listing 2: Filter applied to smoothed motor current reading

```
if movingAverage(current) > threshold
  filterValue += inc_size
else
  filterValue = max(0, filterValue - dec_size)
```

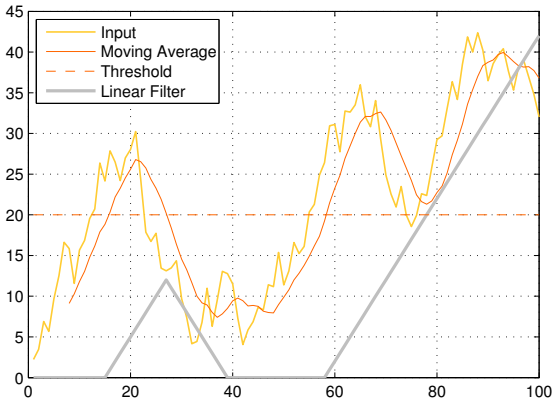



Figure 73: Digital signal filtering for detecting motor overuse. Short spikes do not cause a large response from the linear filter, but a prolonged period of high input will

output from the current filter passes a threshold, a motor current error is thrown by the system. The parameters of the filters, including the thresholds, and the rise and decay constants of the integrating filter, can be set by the user, and are stored in Flash memory.

### ERROR HANDLING

At each pass through the control loop, several checks are made to make sure the robot is operating within the set limits. Firstly, if a sensor read should fail - for example if a magnetic encoder returns an error several times, or if an optical encoder has detected a jump - the error handler is called. The error handler is also called if a joint leaves its safe operating position (i.e. if it is in danger of hitting a hard stop), or if an motor overuse condition is detected, as described above.

The error handling routines define which behaviour will be taken, depending on the severity of the error, as well as logging the error code itself, to be reported to the control PC. A common action is to deactivate the motors, to try and prevent any damage occurring.

### PID CONTROL

The most fundamental part of the control loop is the closed loop control itself. For each motor, a digital PID loop is run, with gains tunable by the user and saved in flash memory. The PID loop has a standard implementation, with a limit on the size of the integral component to prevent integral windup. The output of a PID loop is used to control the pulse width of the PWM signal to the relevant motor.

### MOTOR CONTROL

The hardware PWM generators on the microcontroller are used to generate the PWM signals for the motor. The control boards have the capability to brake the motors, and to enable and disable them completely. On power up, motors are disabled, and must be enabled by commands from the control PC.

### ETHERNET COMMUNICATIONS AND COMMANDS

When the control loop is not executing, the board continuously checks to see if the Ethernet controller has received a frame. This frame will only be available to

the firmware if it has been fully received and the checksum correctly verifies. If a frame is received, the firmware immediately checks to see if in acknowledgement is required, begins the sending procedure if so, and interprets the command. A list of the commands will be given later in this section.

### 4.4.4 *miniBLUE specific control firmware*

In addition to the all of the function performed on BLUE's control boards, miniBLUE's control boards feature a secondary microcontroller. This microcontroller is responsible for reading the optical encoders, reading six ADC channels, and producing PWM signals for controlling damping. An RS232 serial link operating at 1Mhz connects the two microcontrollers. For each pass of the control loop, the main microcontroller transmits two damping values to the secondary microcontroller, and receives four optical encoder statuses and six ADC values. The total length of this communication is 23 bytes, and the transmission therefore takes around 185 µs. The communication is largely handled in the background, allowing the microcontroller to continue interpreting and executing commands without significant slowdown.

For each encoder, two bytes are returned from the secondary microcontroller. A 10-bit position value is contained in the least significant bits, and a 2-bit status value is in the most significant bits. The encoder status can be one of: OK; no-index; index-jump; invalid-state. Initially, each encoder starts with a status of no-index, transitioning to OK when the index mark is observed, and the absolute position determined. If an error occurs, and an invalid state change is detected from the optical encoder, the invalid-state status is returned, to signify that the position can no longer be replied upon. Should the index mark be encountered where it was not expected, the index-jump status will be set.

#### INCREMENTAL ENCODER MODE

A flag can be set in the preprocessor to enable the secondary microcontroller to work with the encoders in incremental mode. In this mode, another sensor such as a magnetic encoder is used to determine the absolute position of the joint at startup, and this information is read by the primary microcontroller and transmitted to the secondary microcontroller. From then, the secondary microcontroller returns a 16-bit

value for each encoder, allowing multiple rotations of the codewheel to occur, without resetting the position when the index mark is detected.

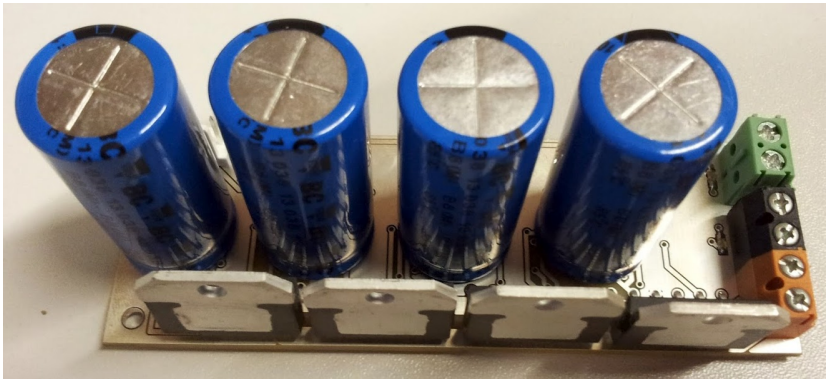
## 4.5 MOTOR DRIVERS

An example motor board for the drive motors of BLUE is shown in Fig. 74a. The board is designed to be extremely robust, and features four LMD18200T H-bridge chips in parallel. These chips include a current sense output, and allow for the disconnection of the motor, as well as motor braking. The boards operate at 48V, the nominal voltage for the drive motors, and a high current voltage clamp circuit prevents voltage spikes which might damage the H-bridges. A polyfuse is included to deactivate the motors if they are being overdriven or if stalling occurs, as an additional line of safety above the software failsafes.

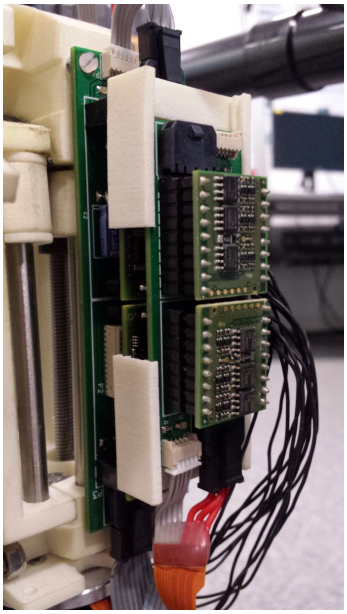
On BLUE, the stiffness adjusters are driven by Maxon DEC50/5 modules on a custom breakout board. A similar technique is used for miniBLUE, as can be seen in Fig. 74b - a driver board for two 70W drive motors is shown on the left, and one for two 30W stiffness adjusters on the right. The stiffness motor drive board uses two Maxon DEC24/3 modules. The driver boards for miniBLUE are more complex than for BLUE, with each board driving two motors. miniBLUE's driver boards include current sensing chips, since the Maxon modules do not include this.

## 4.6 ETHERNET COMMUNICATIONS

The control boards and real-time PC are all connected together on an Ethernet network. The custom protocol is applied above the data link layer, meaning we utilise the standard IEEE 802.3 format for packets. This is shown in Fig. 75, which shows all the bytes sent at the physical layer of Ethernet. The shaded sections (MAC address of destination and source, length, and payload) are ones which are directly set in our system, and the others (preamble, SoF delimiter, Frame check sequence CRC) are the responsibility of the Ethernet hardware. We follow standard IEEE 802.3 protocol, enabling the use of an Ethernet switch in the system. Not shown in the diagram is the 12-byte inter-packet gap.



(a) BLUE drive motor driver board. Dimensions 105x35mm



(b) miniBLUE motor driver boards. Large driver boards are sized 33x110mm and drive two 70W motors. Small driver boards are sized 28x71mm and drive two 30W motors.

Figure 74: Motor driver boards for BLUE and miniBLUE.

| | Preamble | Start of Frame | MAC dest | MAC source | Length | Payload | CRC |
|---|---|---|---|---|---|---|---|
| bytes | 7 | 1 | 6 | 6 | 2 | 46-1500 | 4 |

Figure 75: Ethernet Packet. Shaded blocks are set in software, unshaded blocks are handled by the hardware

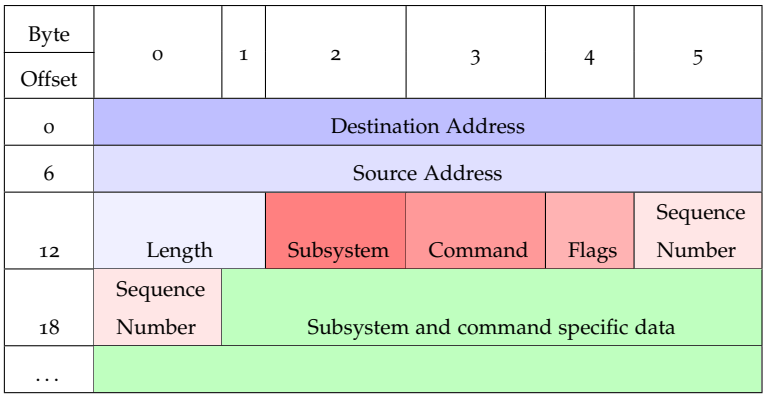



Figure 76: Request Frame

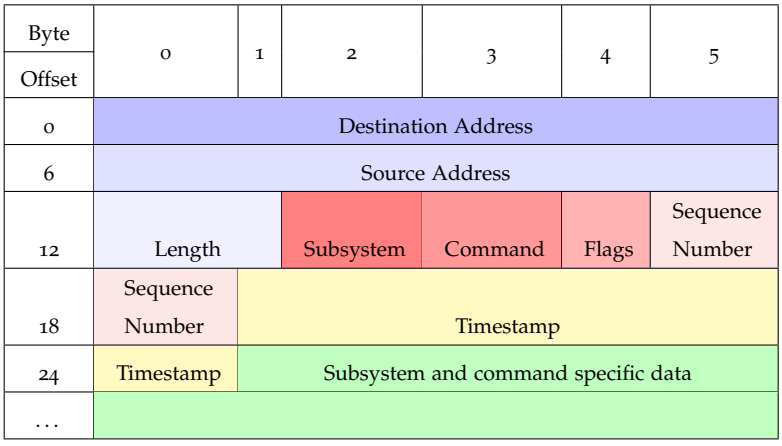



Figure 77: Reply Frame

Figs. 76 and 77 show the structure of the request and reply frames used for communication on the robot network. The first 14 bytes of each frame, containing MAC addresses and packet length, are according to IEEE 802.3 protocol. From byte 15 onwards, in the payload section of the Ethernet packet, is the custom robot communication protocol. The framework is very extensible, but a selection of the implemented commands will be outlined later in this section.

Under our protocol, we define that numbers in the network frame are set as follows:

- Integers: Big-Endian (i.e. network order, RFC 1700)
- Floating Point: IEEE 754

#### FLAGS

The flags are common between request and reply frames, and are a single byte of settings. Presently only two bits of the *flags* field are used:

- Bit 0 (LSB): *ACK_REQ*
- Bit 1: *ACK_REP*

Setting the LSB of *flags* in a frame will therefore trigger an acknowledgement from the recipient device. The acknowledgement itself will have bit 1 of *flags* set. Acknowledgements are typically requested for critical commands, such as activating or deactivating motors.

#### SEQUENCE NUMBER

The *sequence number* is a two-byte integer which is used to identify which request package a reply or acknowledgement relates to.

#### TIMESTAMP

The reply frame includes a timestamp field, for example to allow the accurate determination of when exactly sensor data was retrieved. Although not presently implemented, in concept the timestamps on each control board can be synchronised through a broadcast packet. Alternatively, the real-time PC can sample the timestamps from each board under low network-utilisation conditions, to allow board-specific timestamp offsets to be calibrated.

| Code | Subsystem |
|---|---|
| 0x00 | Internal |
| 0x01 | ADC |
| 0x02 | Motor |
| 0x03 | Damping |
| ... | |
| 0x80 | PID |
| 0x81 | Error |
| 0x82 | Filter |

Table 10: Control subsystems on the control boards

#### SUBSYSTEMS

The individual subsystems of the control board are addressed through the *subsystem* byte of a request frame. The subsystems on the control board are shown in Table 10.

Each subsystem has a number of defined commands with a set syntax, which are listed in Appendix B. For an example, the command specific data for reading from a number of filters would consist of a field specifying the number of filters, followed by that many filter IDs. The reply frame from the control board would be returned with the sequence numbers, and would contain the number of filters, and then the data from each filter in the specified order.

### *Theoretical Performance of the Communication System*

A design criterion for the communication system was that it should be capable of setting all motors, and reading from all sensors at a rate of at least 1kHz. As illustrated above, the Ethernet protocol requires a certain amount of overhead, including two MAC addresses, preamble, and CRC. The overhead on each packet (i.e. the size of the packet excluding the payload) is 26 bytes, and there is also a 12 byte interpacket

gap requirement. At 100MHz, each packet therefore necessarily includes 3.04 µs of overhead.

Furthermore, Ethernet is most efficient with large payloads; most of our communication requires relatively short payloads, and the majority of packets must be padded to reach the 46 byte minimum payload size. The majority of the packets through the system therefore have a total length of $7+1+6+6+2+46+4=72$ bytes. Including the 12 byte interpacket gap, this increases to 84 bytes, or 672 bits. The time to transmit one of these packets and wait for the interpacket gap at 100MHz is then 6.72 µs.

If each cycle requires one write to each board, and one read from each board, the total time required for a communications cycle to six boards is 80.64 µs. At an update rate of 1kHz, this would mean a network utilisation of 8.06%. Even with the necessary Ethernet overheads, the system is therefore more than capable of meeting the update rate criterion.

Furthermore, the architecture of the system means that the network is very flexible for adding devices. For example, if a camera, LIDAR, tactile array, or other sensor producing large amounts of data was added to the robot, the Ethernet based packets are suited for handling the data throughput.

## 4.7 HIGH LEVEL CONTROL

The real-time PC which connects to the on-board robot network forms the gateway between the robot and the outside world. In our implementation, the real-time PC is a FitPC 2i by Compulab, which has a 1.6 GHz single core Intel Atom Z530 chip, but has the advantage of providing two gigabit Ethernet controllers and ports. This allows a wired connection to the Ethernet switch of the robot network, and leaves open the option for a wired connection to a LAN. Additionally, the FitPC generates a wifi network which any device can connect to.

A block diagram of the high level control architecture is shown in Fig. 78. The software running on the real-time PC is split into two distinct sections, a daemon and client scripts. The daemon software is activated on startup and handles the communications to and from the robot. Client scripts utilise the daemon to provide an easy to use interface to the robot, and libraries are provided to control the robot's functions.

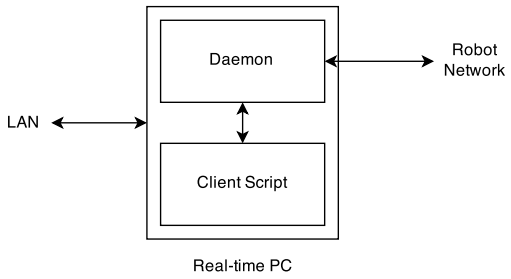



Figure 78: Block diagram of the high-level control architecture.

A variety of client scripts have been written, including ones which enable the remote teleoperation of the robot.

### 4.7.1 *Daemon*

The daemon script runs in the background on the real-time PC, and handles the communications to and from the robot. It opens a raw socket at the packet level, allowing access to the Ethernet packets themselves, as required by the communications protocol. In addition to this socket, the daemon opens a standard Unix socket locally, through which client scripts may communicate with it.

The daemon script checks if an acknowledgement or reply is expected from a request frame, and awaits as appropriate, giving an error if the ACK or reply does not arrive within a specified time window. If an ACK is not received when expected, the daemon will resend the command and try again, a set number of times. This attempts to ensure that critical tasks, such as setting parameters or enabling/disabling motors, are definitely received by the control boards.

### 4.7.2 *Client Scripts*

Client scripts provide the basic user interface to the robot, with libraries giving easy access to all the commands and parameters of the robot. The libraries are hierarchical, allowing a user to select the granularity of control they require. At the lowest level, there are functions for assembling commands suitable for communication with a single board, and scripts work with board-readable numbers. Above this, there are

Listing 3: Example of a high level client script

```
bluerobot blue;
blue.init();
if (blue.readErrors()) {return false;}
blue.enableDriveMotors();
float angles = {0.35, 0.7, -0.35, 0.35, 0.7, -0.35};
blue.setPos(angles);
```

commands which work in radians, and can set positions/read sensors/check for errors with multiple control boards. The highest level of abstraction provides objects for the robots, and convenient functions for working with input/output data files.

Listing 3 gives an example of a client script utilising the highest level of abstraction. This short script would establish a connection to the robot, check for errors, turn on the drive motors, and then move to a squatting position. The *blue* object contains a socket connection to the daemon script, as well as calibration data and parameters for the robot, simplifying the control for the user. The libraries also include functions for loading data files, running trajectories, and performing basic evaluation of trajectories and data.

A special client script, which will be explained in later in the context of teleoperation, sets up the FitPC to enable control of the robot from a remote computer. This client script opens a socket connection on the external LAN, and runs a control loop which allows trajectories and discrete movements to be initiated and executed with precise timing. An external computer can work in tandem with this network control script to enable the robot to be available as a ROS node.

## 4.8 SUMMARY

We have therefore created a robust, high performance electronics and communication infrastructure, which operates both robots, and could easily be adapted to work with other robots. The system takes care of low level control, timing, and communications issues, allowing the user to simple write control algorithms for the robot. If desired,

we have also produced a network interface for the robot, which will be described later in chapter 6, and which also allows control of the robot through a ROS system.

## 4.9 HARDWARE CONCLUSIONS & LESSONS

For this project, two unique variable impedance bipedal robots have been produced. The burgeoning field of variable impedance mechanisms was evaluated to determine which mechanisms were best suited for a legged robot, and a modified version of the AwAS variable stiffness actuator developed and implemented in the robots. Rapid manufacturing techniques were explored for quick production of the robots, and to allow the outsourcing of parts, since the manufacturing facilities we have access to are limited. It has been shown that modern manufacturing techniques such as waterjet cutting and 3D printing can be used to produce novel, complex, hardware, quickly and cost-effectively. Furthermore, a novel production technique which combines waterjet cutting and FDM 3D printing to produce plastic parts with embedded metal has been detailed.

A high performance modular electronics and communication infrastructure was developed, again predicated on affordability and accessibility. The architecture operates over several layers of abstraction, providing precise and accurate low level joint control, sensor reading, and failsafe checking, real-time processing for instructing trajectories, and higher level control, which can be externally located through a network link. Our system shows that Ethernet, which is commonly available on low-cost microcontrollers, can be used to produce a high performance robot control system. Our system achieves update rates of 1kHz with a theoretical network utilisation of around 8%, and allows for multi master control. This is advantageous as it adds another level of capability and security to the robot - functionality such as local reflex loops can be implemented, and an error on one board (for example breach of a current limit) can instantly trigger an appropriate signal to all other control boards.

Of the two robots, miniBLUE is designed to be the most flexible. It is a lightweight robot with 2 DoF hips, and a unique modular design which allows for different types of variable stiffness actuator to be tested. However, a manufacturing fault means that it is difficult to get reliable behaviour from miniBLUE at present. Until this can be

fixed, we concentrate all effort on BLUE, which is a large sagittal plane biped with considerable inertia.

The next part of the thesis will consider the production of human-like walking on BLUE.

# Part III

# PRODUCING MOVEMENT

BLUE was designed specifically to produce human-like locomotion. In this part, the human walking cycle is analysed, and a parameterised representation based on transition events is developed. This transition based model is then used with the kinematics of BLUE to produce walking trajectories with a specified step length, clearance height, and using either specified transition timing, or default timing taken from analysis of human data. The generated trajectories are evaluated in simulation and on the hardware, and we find that locomotion is produced, and the trajectories have particular speeds at which they are most effective. The effect of varying stiffness on a trajectory is investigated, and preliminary results indicate that for slower speeds, a stiffer robot may be better, but at faster speeds, a soft robot may be advantageous.

This provided us with a method of associating parameters such as step length and speed with particular trajectories, which had been learnt on the robot, and for which the best stiffness was known. We also have a system which can detect gait parameters from human walking data. Assembling the two of these provides a teleoperation system which allows a user to control the robot, but wherein the robot copies high level parameters from the walk, whilst using joint angle trajectories and stiffness settings that are more suitable for it.

# 5

# ACHIEVING LOCOMOTION AND INVESTIGATING STIFFNESS

The previous sections of this thesis have dealt with the design and construction of robotic hardware which is capable of physically varying the impedance of its joints. In contrast to rigid robots which can simulate different compliances with careful torque sensing, the constructed robots inherently exhibit this compliance, and can physically store and release energy. When the robot is moving, it must drive its links through the series elastic transmission, and the dynamics of the robot can vary significantly depending on the impedance setting.

It was therefore hypothesised that for a given motion, changing the stiffness of the joints of the robot would have a real effect on the efficiency of that motion. Furthermore, changing the stiffness will change how the robot reacts to external disturbances, both in terms of adapting to unexpected terrain, and in how shock loading is transferred to delicate components like the motor gearheads. It was postulated that, especially at higher speeds, there would be an advantage to the robot being softer. In terms of energy efficiency (i.e. electrical energy consumed by the motors of the robot), it was postulated that there might be a fixed stiffness which gave an optimal performance.

BLUE is designed to ape human like motion, and therefore the human walking cycle was analysed, and trajectories generated for the robot which aimed to reproduce the distinctive heel-toe walk. Trajectories were learned and extracted from human walking data, but also generated from a parameterised ideal of the gait cycle, which took into account the kinematic layout of the robot.

Trajectories were evaluated in simulation before being tested on the actual robot, and were run at a variety of fixed stiffness levels. The robot has large amount of compliance, and exhibits backlash and other dynamic artefacts which affect how a given trajectory is produced, and therefore algorithms to produce a given trajectory, or given gait determinants, were utilised.

The trajectories generated are based on the kinematics of the robot, and it is expected that in the future they could be used as seed trajectories for more dynamic movement generation methods, which would take into account the dynamics of the robot as well. Nevertheless, we find that it is possible to produce locomotion both in simulation and on the hardware, with the trajectories generated herein.

## 5.1 WALK ANALYSIS

BLUE was specifically designed with the aim of reproducing human-like heel-toe walking. In order to do this, it is necessary to understand the joint trajectories and motions involved in more detail, so that similar motions can be recreated on BLUE. The literature on human walking tends to break down a walking step into a number of phases, with a well defined transition event between each phase. Whilst the exact joint angles vary from person to person, a person with a non-pathological gait should produce all of the transition events during normal walking.

### 5.1.1 *Overview of walking cycle*

A typical breakdown of the walking cycle is shown in Fig. 5. A full cycle consists of two steps: two periods of double support and two periods of single support. Each leg alternates between being the stance leg and the swing leg, with periods of double support between swing phases.

Each cycle passes through a number of transition events, four of which are shown in Fig. 79. These occur in the order Feet Adjacent (FA) $\rightarrow$ Tibia Vertical (TV) $\rightarrow$ Initial Contact (IC) $\rightarrow$ Toe Off (TO).

During each swing phase, the swing leg will pass monolithically past the stance leg, through a Feet Adjacent (FA) transition event when the feet are aligned. This FA transition occurs while the swing leg knee is near its most flexed. The knee begins to straighten out, and the swing leg passes through the Tibia Vertical (TV) transition, before the foot hits the ground and begins a period of double support.

Each period of double support begins with an Initial Contact (IC) transition event, typically the heel of the previous swing foot hitting the ground. There is then a loading response where the weight of the person is transferred to that foot, before

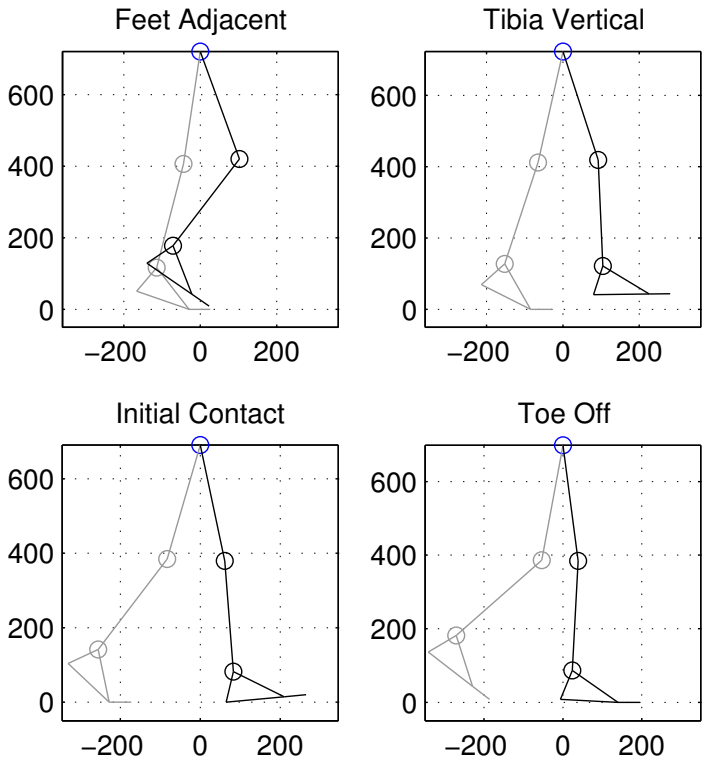



Figure 79: Subset of walking cycle transition events. These occur in order FA → TV → IC → TO → FA

the other foot leaves the ground at a Toe Off transition. This begins a single support period. During the single support period, the stance leg will 'roll-over' the foot, with loading transferring from the back to the front of the foot. After the loading has transferred towards the ball of the foot, the heel of the foot lifts of the ground. The toes flex, and the weight is borne by the ball and the toes of the foot, until the next period of double support.

For the swing leg, the leg initially lifts off the ground to get clearance, before swinging forwards.

### 5.1.2 *Gait transition events*

Fig. 79 shows four walk transition events on a simplified leg model. These four transition events are selected as they allow the characterisation of the double support phase (bookended by the IC and TO events), and the swing phase (TO, FA, TV, IC).

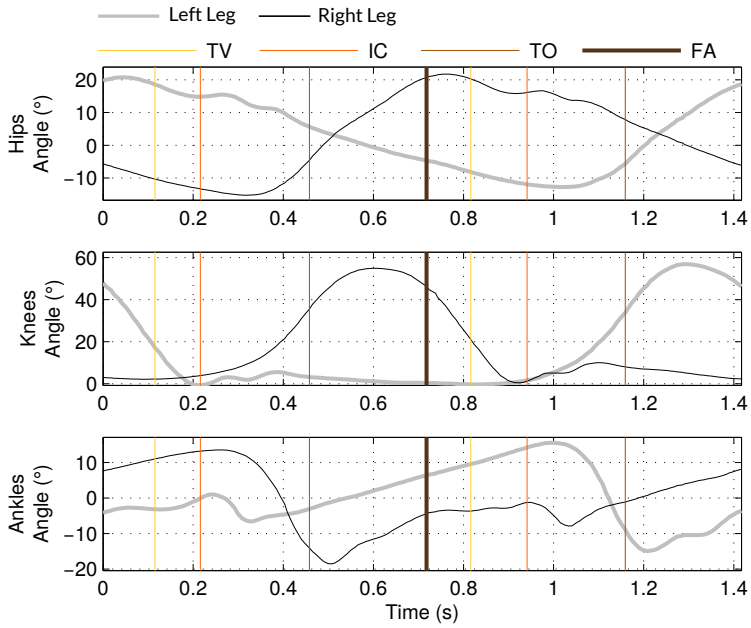



Figure 80: Sagittal plane joint angles for a walking cycle, consisting of two steps. Transition events are marked, FA = Feet Adjacent, TV = Tibia Vertical, IC = Initial Contact, TO = Toe Off. The graphs begin and end at FA.

The Heel Off (HO) transition is not explicitly shown, as it is a natural by-product of the changing loading on the stance foot.

Each of the four transition events shown in Fig. 79 occurs at a well defined instant in time. During the swing, the swing foot cleanly passes to become in front of the stance foot, and the tibia passes monolithically through the TV transition, as can be seen on the joint angle graphs in Fig. 80. For the IC and TO, there may be some small amount of bounce, but we consider the first point at which the foot hits the ground, and the final point at which the toe leaves the ground.

### 5.1.3 *Analysis of human walking data*

Human walking data was collecting using an xsens MVN motion tracking system with a treadmill, as shown in Fig. 81. The xsens system includes a number of trackers, effectively one per link of the human body, including a tracker on the pelvis, on each femur, on each tibia, and on each foot. The trackers are placed as close to a bone as possible to make the resulting signals more clean. Each tracker includes

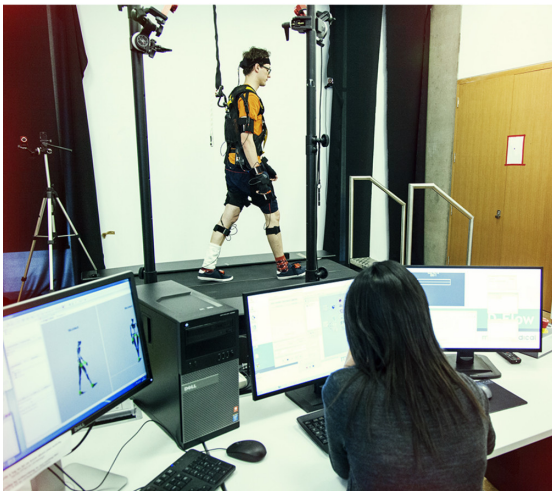


Figure 81: xsens MVN system used to gather walking data

inertial sensors including a three dimensional accelerometer and a three dimensional gyroscope, as well as magnetometers.

The individual xsens trackers are wired together, but the worn system is battery powered, and communicates wirelessly with a base station attached to a PC. The PC is a Windows machine, which runs the MVN Studio software to interpret the data from the trackers. The measurements of the subject are input into MVN Studio, and the software maintains a body skeleton with appropriate dimensions and kinematics, processing the data from all the trackers to provide link positions and orientations as well as three dimensional joint angles.

The data from the trackers can either be saved from MVN studio into a parsable file format at the end of a run, or can be streamed in real time over a network connection. For the initial data analysis, data recorded by another PhD student, Hsiu-Chin Lin, was utilised. This data was for five different subjects, and included walking at five different nominal speeds.

A Matlab script was written to parse the walking data, and detect each of the four transition events shown in Fig. 79. The script utilised a kinematic model of the subject to obtain positions for the salient points of each leg: hip, knee, ankle, heel, ball of foot (metatarsophalangeal joint), and the distal end of the toes. The position of the floor was calculated to be the lowest of the four points: left heel, left ball, right heel, right ball. The toes were assumed to be at a neutral, zero degree angle, unless that would

push them through the floor, in which case they were deflected to make them flat on the floor.

In this way the lack of explicit data on the position of the toes was compensated for during the important stance phase. However, the position of the toes during swing is always assumed to be "straight" - if the subject were to lift their toes during swing through, this would not be detected in the system. However, since the robots have only passive toe joints, the lack of explicit data about this is not encumbering to our control strategies.

The xsens log data was imported into matlab via an *mvnx* file saved from MVN studio. This is an XML data format which includes segment length information, and for each frame includes Cartesian coordinates for each joint, quaternions describing the orientation of each segment, and XYZ angles representing the three principle rotations of each joint. The joint angles are given according to a co-ordinate frame in which y is up when the leg is straight, z represents the sagittal rotation axis, and x is set according to the right hand rule. The three angles nominally therefore represent adduction/abduction, internal/external rotation, and flexion/extension, in that order.

It was found that the coordinate data often relayed exaggerated twist of the legs and feet, resulting in trajectories with foot collisions. It is speculated that this is because the information from the accelerometers regarding the gravity vector is more reliable than that of the magnetometers regarding the location of the magnetic north, especially on a treadmill.

To ameliorate this, a script was written which read in the segment lengths from the mvnx file, and used these in a forward kinematic model of the subject. The orientation of the pelvis quaternion was first used to rotate the body frame, and then the kinematic transforms for each leg were applied. In this case, the angles representing internal/external rotation were set to zero, resulting in more realistic output poses.

We also wished to rotate the data such that it always represented a subject moving forward in the x-direction, which is not the case in the raw data. Since, however, the subjects were walking on a treadmill with a constant direction, there is a fixed rotational offset around the vertical axis. We calculated the average deflection of the pelvis from the coronal plane, and applied this as an offset around the vertical axis.

This takes the form of a matrix $R_{zrot}$, which applies a course-corrective rotation about the z-axis.

In our system, the z-axis is up, the x-axis is forwards, and the y-axis is set according to a right handed coordinate frame.

From the pelvis quaternion: $Q_p = q_r + q_x i + q_y j + q_z k$, a kinematic matrix can be derived:

$$R_{pelvis} = \begin{bmatrix} 1-2q_y^2-2q_z^2 & 2(q_xq_y+q_zq_r) & 2(q_xq_z-q_yq_r) & 0 \\ 2(q_xq_y-q_zq_r) & 1-2q_x^2-2q_z^2 & 2(q_yq_z+q_xq_r) & 0 \\ 2(q_xq_z+q_yq_r) & 2(q_yq_z-q_xq_r) & 1-2q_x^2-2q_y^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{83}$$

The kinematic chain for each leg is therefore

$R_{zrot}R_{pelvis}T_{pelvis\to hip}R_{xzy,hip}T_{hip\to knee}R_{xzy,knee}T_{knee\to ankle}R_{xzy,ankle}$,

followed by translations representing the move from the ankle to the heel, or ball of the foot, and then a further rotation and translation representing the toes. $R_{xzy,joint}$ represents the rotation at a joint, with the order in which the rotations are applied set to match the data from the mvnx file:

$$R_{xzy} = \begin{bmatrix} c_zc_y & 0-s_z & c_zs_y & 0 \\ c_xs_zc_y+s_xs_y & c_xc_z & c_xs_zs_y-s_xc_y & 0 \\ s_xs_zc_y-c_xs_y & s_xc_z & s_xs_zs_y+c_xc_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{84}$$

Where $s_x$ is the sin of the rotation about the x-axis, and $c_x$ is the cos of the rotation about that axis. $T_{joint\to joint}$ represents the translation between two joints, with distances extracted from the segment lengths in the mvnx file.

These kinematics were applied for each frame of the walking data, to give coordinates for each of the body points with a corrected internal/external rotation value, whilst also ensuring that the coordinates represented forward motion in the x-direction.

From the corrected 3D point data, transition events were detected.

Fig. 82 shows the detected transition timings for the five subjects, at three different walking speeds. The results show that there is some inter–subject variability in tran-

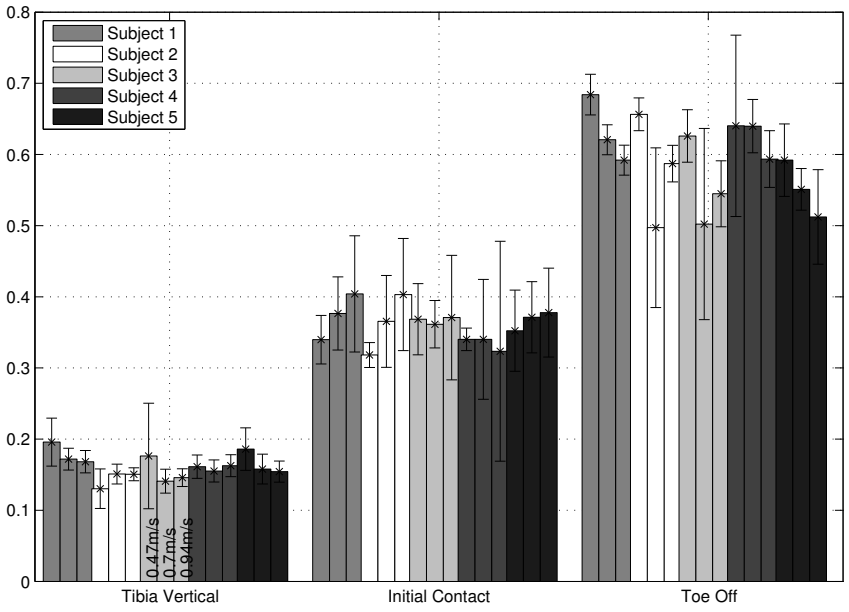



Figure 82: Transition event timings for the five subjects, at three different speeds. Starting from a Feet Adjacent condition, the timings for the Tibia Vertical, Initial Contact, and opposite Toe Off, are shown as a fraction of the gait cycle.

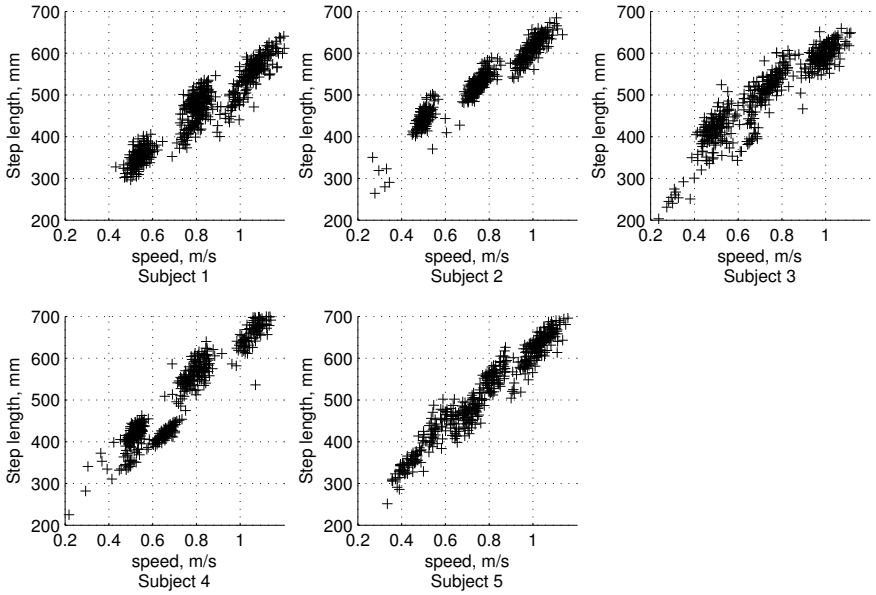



Figure 83: Step Length vs. Walking Speed

sition timings, but for the majority of the subjects, increasing the speed of the walk also slightly increases the proportion of time spent in the swing phase, since the toe off event occurs earlier in the cycle.

In addition to detecting gait transition events, the Matlab script also calculates step length and walking speed from the xsens data. Fig. 83 shows how the step length of the subjects changes with walking speed. As would be expected, at greater speeds, the subjects increase their step length. The relationship between walking speed and step length appears near-linear, but does not result in a constant step frequency - subjects take more, longer steps at higher speeds, rather than just taking longer steps, or more steps.

The information from this walking analysis was used in two ways: firstly, to provide ground truth to a walking trajectory generator designed to produce human-like locomotion on BLUE; and secondly, to aid the development of a system for detecting and characterising locomotion on-line, for teleoperation of the robot.

## 5.2 SOURCES OF WALKING TRAJECTORIES

Walking trajectories can be generated in a variety of ways, from specifying ZMP or center of mass trajectories, to specifying sinusoidal foot movements. Neurologically inspired methods can be used, for example those which utilise Central Pattern Generators and/or reflexive motor control.

Whilst it is unlikely that simply playing back a human walking trajectory will produce the most efficient locomotion on a robot, it is still possible to mine sequences of human walking data for promising trajectories. Therefore, initially trajectory data for walking was gathered by evaluating human walking data in simulation, and selecting segments which produced promising motion. This consists of utilising a moving window score function which rewards consistent forward motion across a long sequence of human walking data, and extracting segments with a positive score. The process is explained further in section 5.2.1.1.

The nature of BLUE as a highly anthropomorphic robot, designed to walk with a heel-toe gait, gives this simple method a far higher likelihood of succeeding, compared to robots with a flat foot. Furthermore, BLUE is a sagittal plane robot, and therefore does not need to worry about lateral balancing.

After extracting functional trajectories from human walking data, a method for generating idealised walking trajectories according to given parameters was investigated. This method uses high level parameters such as step length and clearance height to determine the required robot pose at each of the main gait transition events, and is informed by the analysis of human just presented. For the production of human-like locomotion, it will be necessary for the robot to pass through each of these poses.

### 5.2.1 *Human walking data*

As mentioned above, walking data from five subjects, at five different walking speeds, was available for analysis. To investigate if the robot is capable of producing properly human-like locomotion, this data was mined for segments which would produce locomotion on the robot. This also provided initial data on the feasibility of direct human teleoperation of the robot.

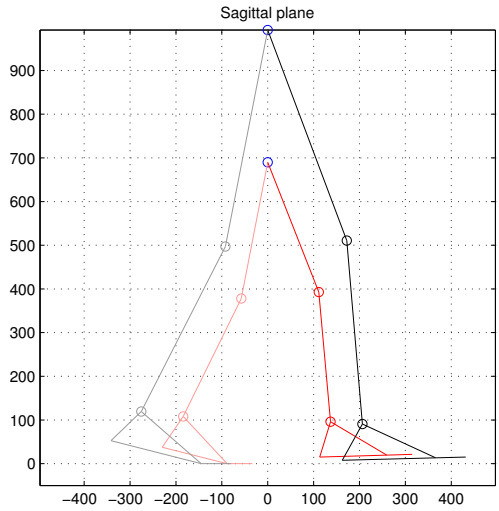



Figure 84: Kinematic effect of direct sagittal joint angle transfer between human data and BLUE. The (tall) subject is shown in black, while BLUE is shown in red.

#### 5.2.1.1 *Evaluation in Simulation*

The xsens walking data was initially played back on the V-REP simulation of BLUE, with the robot set to high stiffness. To produce the angle data used in these first simulations, it was necessary to adjust the hip angle data to reflect angles from the vertical, rather than from the pelvis. The xsens log files include quaternions representing the orientation of each body segment with respect to a global frame, and so the rotation in the sagittal plane was extracted from the pelvis quaternion and used as an offset to the hip joint rotation angles.

For a first experiment, only the sagittal plane joint angles (flexion/extension) were extracted from each frame of the walking data, and were transposed 1:1 onto the simulated robot. Fig. 84 shows an example of the effect of this transfer kinematically. In this case, the subject is quite tall, and due to abduction/adduction which BLUE cannot copy, there is a small discrepancy between the heights of the swing feet. Due to the anthropomorphic design of BLUE, there is, however, a close match between the foot trajectories, and we therefore explored whether a direct joint angle transfer would lead to locomotion.

The trajectories generated from the xsens data were played back at different speeds in simulation, at a speed close to that at which they were recorded, and at slower

speeds. For each input data point, the V-REP control script output a line to a log file, containing information on the target and actual position of each joint, the compliant deflection, the boom arm angles, and the force sensor data. These log files were then analysed to determine the performance of these trajectories in simulation. The torque on each joint can be calculated from the torque function of each variable stiffness mechanism.

Due to the compliance in the joints, the final output position of the joints is deflected from the input trajectory. Often, this leads to the robot shuffling its feet, or not producing the desired forward motion, and seemingly increases torque as the robot attempts to drive limbs through ground contact positions.

When locomotion was produced, the efficacy of it was evaluated according to a cost function which was designed to reward judder-free forward motion, and penalise torque on the joints. To select segments of the logs which represented viable forward motion, a moving window reward function was used, with the form:

$$S_{\text{forward}} = \sum_{i=n}^{n+N} \begin{cases} 1 & \text{if} \dot{\alpha}_{z,i} \geqslant 0 \\ -2 & \text{if} \dot{\alpha}_{z,i} < 0 \end{cases} \tag{85}$$

Where $n$ is the starting index of the window, $N$ is the window size, and $\dot{\alpha}_z$ is the derivative of the yaw angle of the boom arm - i.e. the rotation of the boom arm about the vertical axis. This function therefore remains above zero if a segment produces forward motion, but penalises backward motion. The function is also independent of step size, meaning a walk which produces smaller steps is not unduly penalised.

The window size was set to encompass a nominal 2.5 steps, and the trajectory logs were split into segments which produced reliable motion, by discarding any portions for which $S_{\text{forward}} < 0$. For the segments which produced a reliable forward motion, a more detailed reward function was used:

$$S_{\text{vertical}} = -K_{\text{vertical}} \sum_{i=n}^{n+N} |\dot{\alpha}_{y,i}| \tag{86}$$

$$S_{\text{torque}} = -K_{\text{torque}} \sum_{i=n}^{n+N} \sum_{j=1}^{\text{numjoints}} T_{j,i}^2 \tag{87}$$

$$S = S_{\text{forward}} + S_{\text{vertical}} + S_{\text{torque}} \tag{88}$$

where $\dot{\alpha}_y$ is the derivative of the pitch angle of the boom arm, i.e. represents the vertical motion of the pelvis of the robot, $T_{j,i}$ is the torque on joint $j$ at point $i$, and $K_{vertical}$ and $K_{torque}$ are gains.

The two gains, $K_{vertical}$ and $K_{torque}$ are set to provide scores on a similar scale to $S_{forward}$, so that any trajectory that produces consistent forward motion will receive a positive score, unless it uses excessive torque, or has extremely exaggerated vertical pelvis motion. The gain for the torque was first set, to a level which selected for torques the robot could produce, and then the vertical gain was set so that a normal amount of vertical movement did not result in any great deal of penalisation. In our case we used $K_{torque} = 0.0005$ and $K_{vertical} = 50$.

This function therefore rewards forward motion, while penalising excess vertical motion and torque. The resultant score, $S$, represents a trade off between how well a motion produces locomotion, and how energy expensive it is.

Individual steps were extracted from these multi-step logs, based on areas with high scores according to the above function. The ends of these step trajectories were merged to ensure cyclicity.

The scoring does not take into account the clearance of the leg on swing through, and in some of the selected trajectories the swing leg does brush the floor, although not enough to greatly impede the walk. In the future, the scoring could also be improved to consider the movement of the pelvis relative to that produced during human motion, rather than on general principles of producing consistent forward motion without excessive vertical motion.

### 5.2.2 *Generation from Gait Determinants*

Walking trajectories gathered directly from human data are unlikely to produce an ideal walk for the robot, even if cherry picked from a large amount of data. This is in part due to the differing kinematics of the robot - lacking adduction/abduction joints, as well as its different dynamics. Furthermore, selecting trajectories in this manner does not allow the easy selection of parameters such as the step length and speed.

A method of generating trajectories based on specifying gait transition poses and timings was therefore developed and evaluated. This method allows the specification of step length and clearance height, and generates a trajectory which meets those

criteria. In the future, such trajectories could be used as seed trajectories for dynamic stabilisation. On BLUE, which is a sagittal plane biped without a torso, the trajectories alone can produce locomotion.

#### 5.2.2.1 *Parameterisation of human walking*

As explained previously, the four gait transition events FA, TV, IC, TO can be used to describe a walking trajectory. Each of the transition events has well defined requirements, for example the requirement in FA for the two feet to be adjacent (i.e. have the same x-coordinate). If the step length, clearance height, and hip motion is specified, the joint angles for each transition event can be calculated with few additional parameters required.

With a very small set of free parameters, the poses for BLUE at each gait transition point can be fully specified. The remaining free parameters relate to the timing of the transition events, and to set these we utilise the information gathered from analysis of human data, as shown in Fig. 82

An example of the transitions used in the walk generator is given in Fig. 85. The input to the walk generator is nominally only the *step length* and *clearance height*, but optionally the transition timings can also be specified. For each transition, the horizontal motion of the pelvis is set according to a linear forward motion with the specified transition timings. A brief outline of the procedure for solving each transition pose is given below:

##### INITIAL CONTACT

At initial contact, the hip angles are roughly equal and opposite, and the amplitude of the angles can be calculated according to the specified *step length*. The forward leg, which is just contacting the ground, has a basically straight knee ($\approx 2°$) and a very slight dorsiflexion on the ankle. With the forward leg angles specified, the coordinates of the impacting heel are calculated, and the step length used to calculate the position of the toe and ball of the back leg. During initial contact, the knee joint of the back leg is slowly starting to bend, and is given a nominal flexion ($\approx 5°$). The angles here are extracted from the earlier analysis of human walking, where many transition events were extracted from human walking data. With this specified, it is simple to solve for the back leg hip and ankle angles.

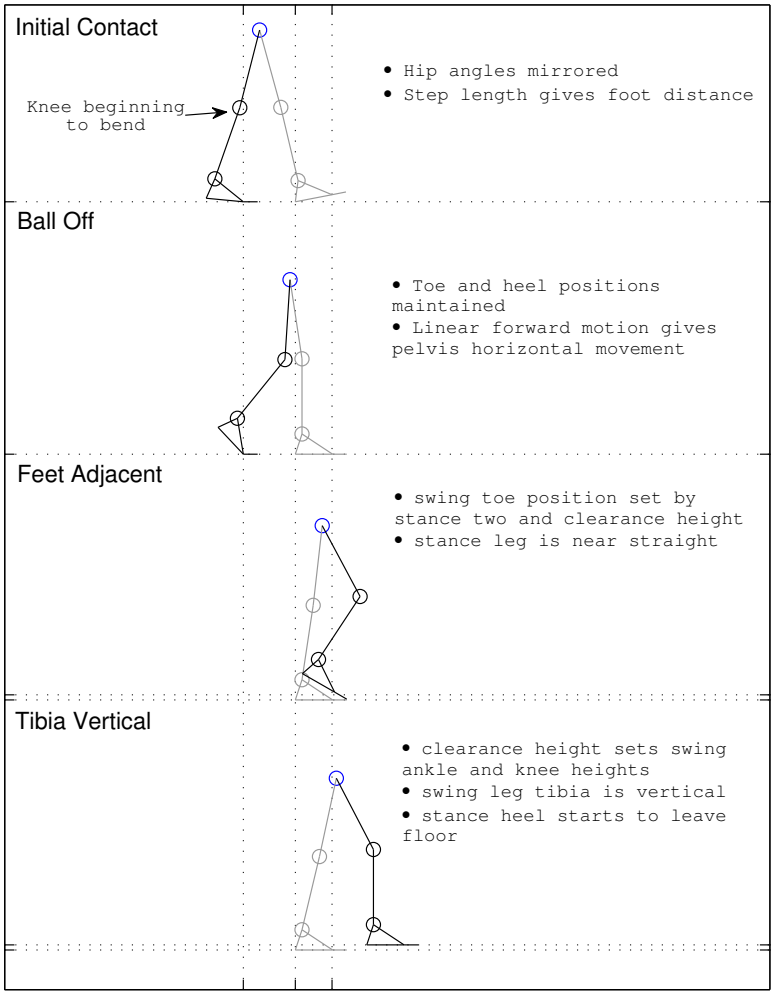



Figure 85: Transitions used by walk generator. The timing of each transition is set according to realistic timing from human data, but can be specified as additional parameters if desired.

#### BALL OFF

It is assumed that there is no slip of the feet on the ground, and therefore the co-ordinates of the forward heel and back ball are unchanged from the IC transition. The pelvis position is calculated according to a nominal linear forward motion, using the timing of the ball-off transition. The forward foot is flat on the floor and the knee directly above it. This allows the vertical position of the pelvis to be calculated. With this known, the angles of the back leg can be calculated, although it is necessary to specify either one joint angle, or the angle of foot lift off, in order to reduce the number of potential solutions. Based on human walking data, we have found that setting a nominal knee angle of 35° gives good performance.

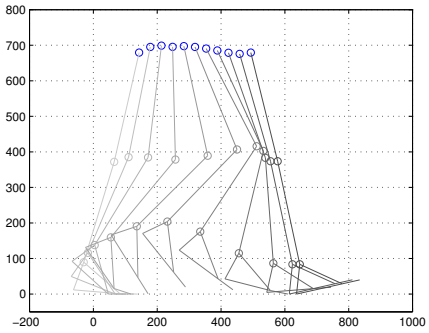
#### FEET ADJACENT

The position of the stance foot is unchanged, and the horizontal co-ordinate of the pelvis is again calculated based on linear forward motion. The stance leg is near straight, and this allows the calculation of the vertical pelvis co-ordinate. The coordinates of the swing leg toe are specified such that they match the horizontal position of the stance leg toe, with a vertical position given by the *clearance height*. A slight dorsiflexion is applied to the swing ankle to improve swing through clearance, and this allows for the remaining angles of that leg to be solved.

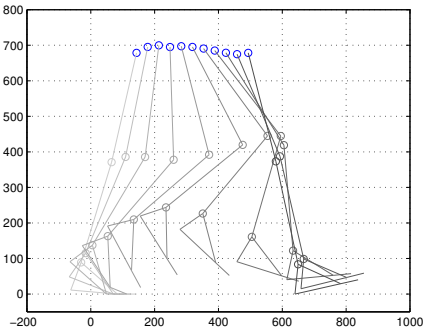
#### TIBIA VERTICAL

Stance foot position is once again unchanged, and pelvis horizontal position is again determined according to the timing of the transition. The position of the stance foot is unchanged, except the heel has begun to lift, and the leg is near straight, allowing the calculation of the stance leg angles and pelvis vertical coordinate. The swing foot is near flat, and the tibia is by definition vertical, allowing the calculation of the vertical co-ordinates of the ankle and knee from the specified *clearance height*. The joint angles of the swing leg can then be solved.

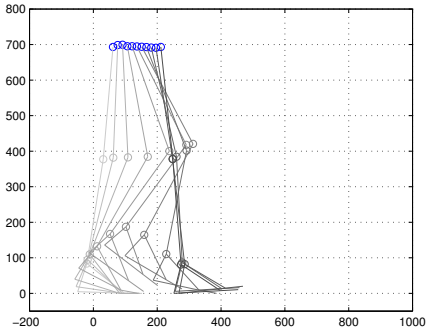
To ensure that the generated trajectory is cyclic, we generate a spline through three entire walking cycles, and extract the middle cycle. After applying the cubic spline through the points and timings specified by the transition data, any knee an-



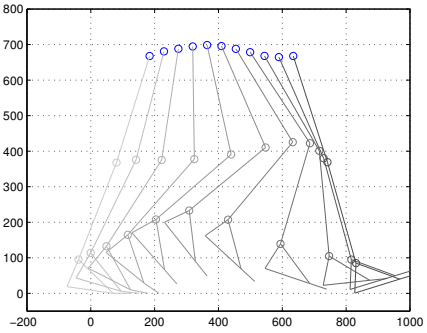

(a) Normal Step, SL=350 CH=10




(b) High Clearance, SL=350 CH=50




(c) Short Step, SL=150 CH=10




(d) Long Step, SL=450 CH=30

Figure 86: Example swing leg motions from walk generator, for a variety of step length (SL) and clearance height (CH) parameters

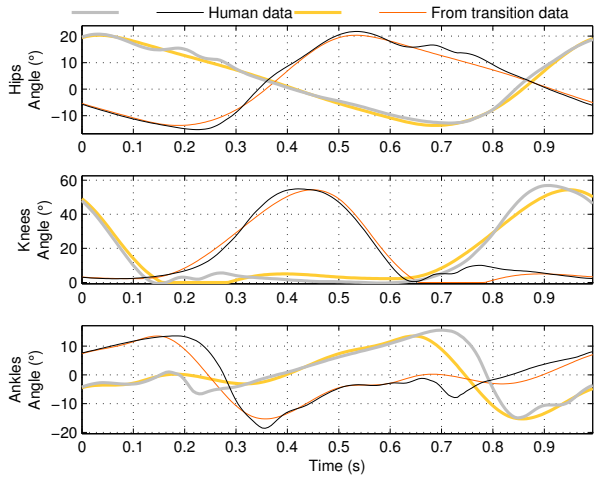



Figure 87: Comparison of human walking data to trajectory generated from transition poses

gles which would lead to hyperextension (i.e. less than zero), are set to zero as a sanity check.

Example swing leg angles from the walk generator are shown in Fig. 86, where it can be seen that the generator allows the use of just two parameters - step length and clearance height - to produce a wide variety of trajectories. Optionally the transition timing data can be explicitly provided to further tune the walk.

Kinematically, these trajectories should produce motion with the specified step length, but as seen earlier, humans utilise smaller steps at lower walking speeds and longer steps at faster walking speeds, with the period of the walk also decreasing slightly. It was postulated that each generated trajectory would have a range of step periods at which it would produce stable motion - producing a range of stable walking speeds for that trajectory. This was evaluated in simulation as shown in section 5.3.2

#### 5.2.2.2 *Comparison to human data*

With the poses and timings of the gait transition events fully specified, a cubic spline is used to interpolate the joint angles to produce a full walking trajectory.

To validate that using a cubic spline to interpolate between transition poses can produce angle data similar to that of actual human walking data, the technique was tested using transition poses and timings taken from actual human data. Fig. 87 shows this comparison, where transition points were automatically detected from a

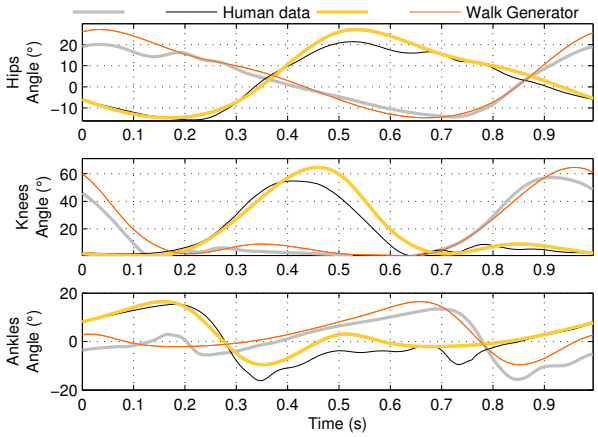



Figure 88: Comparison of trajectory from walk generator to joint angles taken from human data

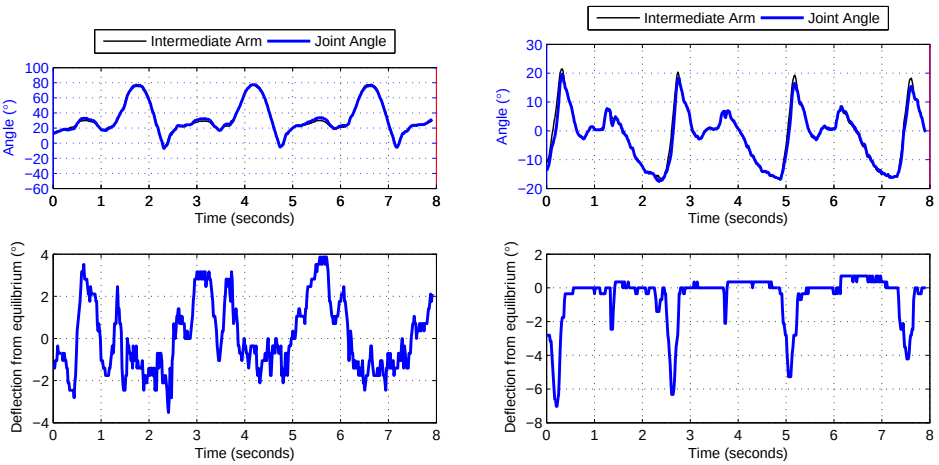
step of human walking data, and a smoothed spline generated between them. Note that in the graphs, two steps are shown, and whilst the generated data repeats, the human data is slightly different for the second step. The figure shows that the generated data closely matches the hip and knee, but has a greater difference from the human ankle angles, particularly with regard to the sudden ankle movements directly following initial contact.

Fig. 88 shows a trajectory taken from the walking generator alongside one taken directly from human walking data. There is a strong qualitative match between the real data and the angles generated from gait determinants - in this case for BLUE with a specified step width of 350mm and clearance height of 25mm. Notably, the hip and knee angles reach greater magnitudes than the human data, and this is primarily as a result of the need to compensate for the lack of adduction/abduction in BLUE, and ensure that the swing foot does not contact the floor during swing through.

## 5.3 PRODUCING LOCOMOTION

The previously described sources of data provided trajectories which produced walking behaviour in the V-REP simulated robot. The most successful trajectories were then run on the real robot.

The graphs in Fig. 89 show that the robot is capable of driving the intermediate arms to reach the complex trajectories extracted from human data. The lower graphs




(a) Knee (b) Ankle

Figure 89: A human walking trajectory played back on BLUE at high stiffness. Lower graphs show a detailed view of the compliant deflection

of each subfigure, which show the deflection of the compliant element, illustrate the cyclic loading during several walking steps.

However, backlash in the joints, which is not present in the simulation, as well as additional real-world disturbances and differences in damping and dynamics, meant that the output trajectory was not the commanded trajectory. Even at higher stiffness levels, the compliance in the robot means that the actual output position is deflected from commanded equilibrium position.

### 5.3.1 *Trajectory learning*

There are several options for working with the generated trajectories on the compliant robot BLUE. Firstly, the behaviour of the robot can be adjusted until the output links reach the specified trajectories. This means compensating for the compliant deflection, backlash, and other disturbances. Secondly, if the walking trajectory is attempting to produce a desired behaviour, e.g. specified step length and ground clearance, the behaviour of the robot can be adjusted to try and meet these parameters. Thirdly, the compliance can be allowed to deflect the joint angles, and the robot's behaviour can be adjusted only to try and ensure foot clearance and a stable walking trajectory.

#### 5.3.1.1 *Achieving a desired trajectory*

To obtain a desired output behaviour, we adopted a model-free approach which iteratively improves a cyclical command trajectory to try and match a given reference output trajectory. A model free approach is useful for several reasons: the same algorithms can be used on the simulation and on the hardware, or on different robots; modelling the full dynamics of the real word robot, including static friction and backlash, is non-trivial; modelling impact dynamics is non-trivial; the real-world robot may degrade over time, the model free approach can adjust itself without relying on the recalibration of an underlying model of the robot.

In our approach, the robot is given a command trajectory which initially follows the desired output reference trajectory. A number of steps are taken to allow the robot to settle, before the actual output trajectory is evaluated. The produced velocity of each joint is compared to the reference trajectory velocity, and the command trajectory velocity is adjusted to push the output velocity closer to the reference velocity. A similar calculation for position is carried out, so that both the position and velocity of the output trajectory are adjusted towards the reference output trajectory. The algorithm is outlined in Algorithm 1.

Note that in Algorithm 1, the low pass filter, differentiation, and integration operations are all based on cyclical trajectories. The low pass filter has no phase lag, and smooths between the end of one trajectory cycle and the beginning of the next. Similarly, the differentiation and integration operations do not change the size of the trajectories - differentiating an N point position trajectory will yield a N point velocity trajectory. The phase lag compensation step takes in two cycles of walking (i.e. four steps), and moves the reference trajectory along the measured trajectory, evaluating where the distance (we use the Normalised Mean Square Error) between the reference trajectory and output trajectory is at a minimum. This offset is then taken as the phase lag between command and behaviour.

The method does not impose any restrictions on the form of the output, and thus can adapt to a wide variety of conditions and models. It copes naturally with physical compliance, but can also adapt to backlash in joints. Initially, a trajectory may fail, with the robot hitting its foot during swing through, for example. On a successful run, across several iterations more of the trajectory will be successfully produced,

**Algorithm 1** Iterative Trajectory Learning

**Input:** reference output trajectory $x_{ref}$, measured output trajectory $x_{out}$, previous command trajectory $u_{in}$

**Output:** updated command trajectory $u_{out}$, log files from each iteration

**function** PUSHTRAJECTORY($u_{in}$, $x_{measured}$, $x_{ref}$)
    $x_{error} \leftarrow x_{ref} - x_{measured}$
    LOWPASSFILTER($x_{error}$)
    $u_{out} \leftarrow u_{in} + Kx_{error}$
    **return** $u_{out}$

**function** DOITERATATION($u$, $x_{measured}$, $x_{ref}$)
    PHASELAGCOMPENSATION($x_{measured}$, $x_{ref}$)
    $u_{pos} \leftarrow$ PUSHTRAJECTORY($u_{in}$, $x_{measured}$, $x_{ref}$)
    $u_{vel} \leftarrow$ PUSHTRAJECTORY($\dot{u}_{in}$, $\dot{x}_{measured}$, $\dot{x}_{ref}$)
    $u_{out} \leftarrow \int u_{vel} dt + u_{pos,0}$

**procedure** ITERATETRAJECTORY($x_{ref}$, $N_{iterations}$)
    $u \leftarrow x_{ref}$
    **for** $i = 1 : N_{iterations}$ **do**
        $X \leftarrow$ RUNTRAJECTORY($u$, $N_{cycles}$)
        $x_{measured} \leftarrow (X_{motors} + X_{deflection})|_{\text{(last two cycles)}}$
        $u \leftarrow$ DOITERATION($u$, $x_{measured}$, $x_{ref}$)
    $u_{out} \leftarrow u$

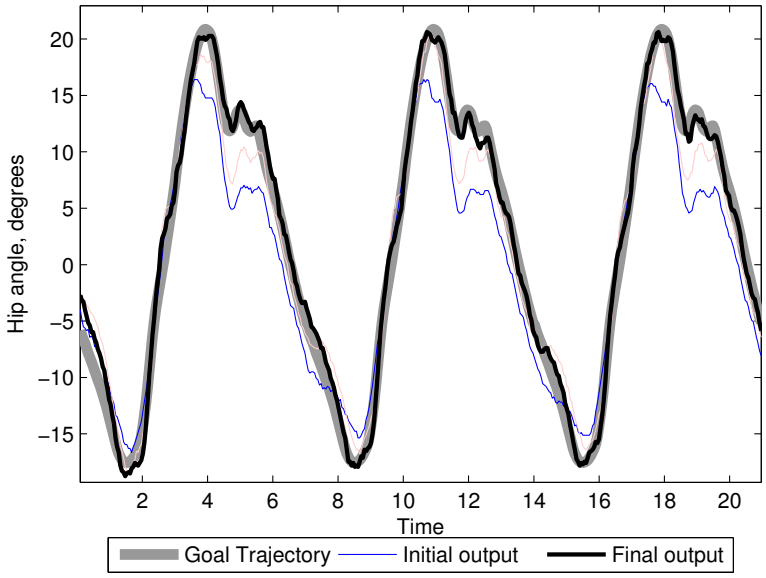



Figure 90: Results from iterative proportional gain controller for learning trajectories

until the robot is walking with the prescribed motions, and the controller continues to make minor adjustments to the command trajectory.

Four cycles were performed per iteration. The first two to settle into a new trajectory, the third to measure, and the fourth to allow for the output lagging behind the motor commands. We utilised a proportional gain, K, of 0.4.

This approach is illustrated in Fig. 90, which shows the joint angles for one of the hips of BLUE, attempting to reproduce a trajectory taken from human walking data, with irregularities. This graph shows the behaviour of the actual robot, at a medium stiffness level. The gain of the trajectory controller was set to 0.5, and three iterations were performed. As can be seen from the graph, the initial output trajectory of the hip joint is far removed from the desired output trajectory, but after three iterations it conforms fairly closely. In general, it was possible to achieve a Normalised Mean Square Error (NMSE) between the desired trajectory and the actual trajectory of around 0.02 for high stiffnesses, and 0.03-0.04 for lower stiffnesses.

### 5.3.2 *Evaluation in Simulation*

A variety of trajectories were generated and evaluated in simulation. To begin with, nominal step lengths 250, 300, 350, 400, and 450mm were generated, with clearance heights for each of 25, 35, 50 and 75mm, to give a total of 20 evaluation trajectories. These trajectories were then run in V-REP, with the stiffness of the joints set to maximum. In order to make the output motion of the robot as close as possible to the desired output trajectory, a proportional gain trajectory iterator of the type described in section 5.3.1.1 was utilised.

The best performing trajectories for a range of walking speeds were then selected, and simulated with varying stiffness levels. In this evaluation, the stiffness of each joint of the robot was set to the same percentage of its maximum. The control trajectories were again iterated to produce output motion as close as possible to the desired trajectory. Fig. 91 shows a flowchart of the general procedure for generating and evaluating trajectories.

The performance of this is shown in Fig. 92, which shows 6 iterations of the trajectory, and the score of each iteration according to eq. 88.

The behaviour in simulation is shown in more detail in Fig. 93, which shows the performance of an iteration after learning has plateaued. The joint angle graphs show the eventual intermediate arm angle, and the output position (intermediate arm angle + compliant deflection). The output position closely reflects the specified trajectory, and the motion of the robot results in continued forward motion, as is shown in the graph of boom arm angles. The evolution of the score function (eq. 88) is also shown in Fig. 93, and for this walk results in a positive score.

Each trajectory was evaluated at a range of different speeds, and it was found that all of the trajectories were capable of producing forward motion at a range of speeds. For each speed, we calculated the energy cost per distance travelled:

$$C_{EPD} = \frac{1}{D} \sum_{n=0}^{N} \delta_{t,n} \sum_{j=0}^{J} T_{j,n}^2 \tag{89}$$

Where $D$ is the forward distance travelled, $N$ is the number of data points, $\delta_{t,n}$ is the time step for point $n$, $J$ is the number of joints, and $T_{j,n}$ is the torque on joint $j$ at data point $n$. This equation is based around the fact that the nominal current

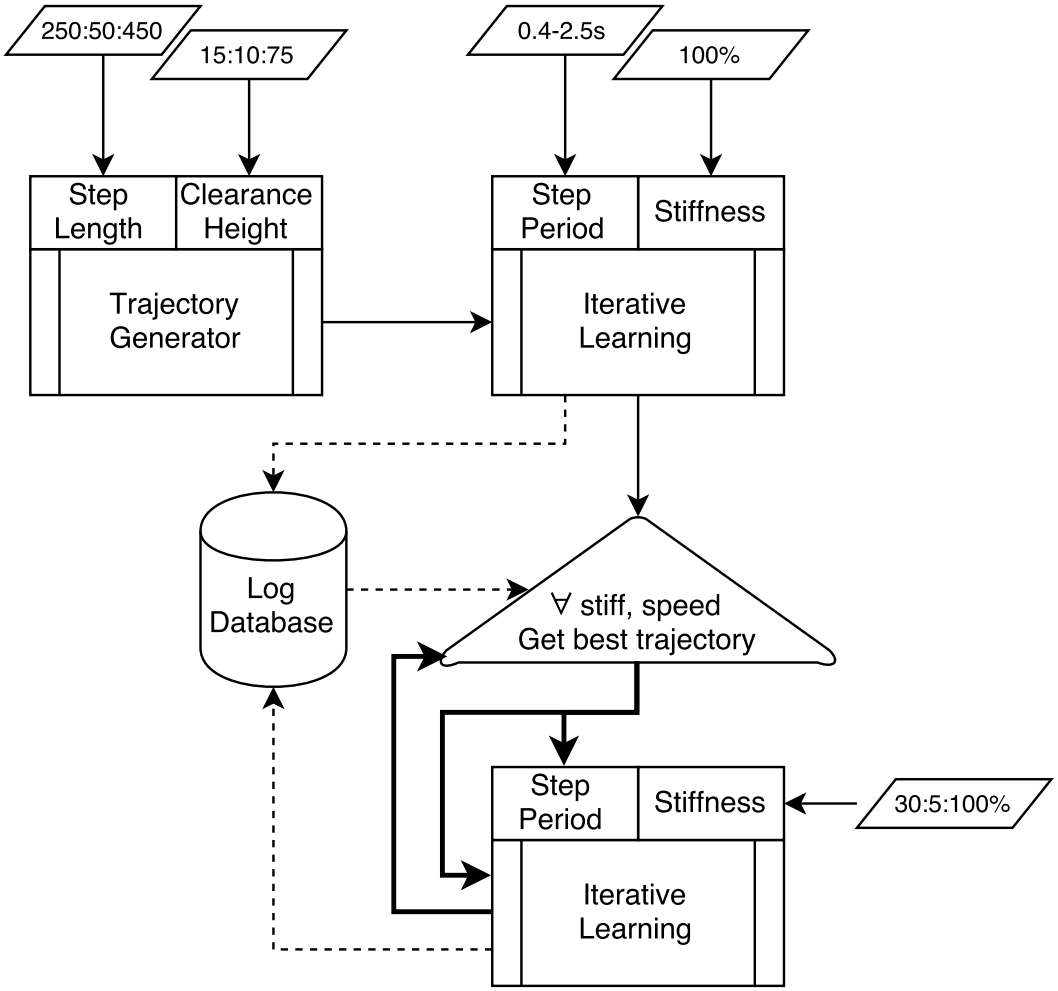



Figure 91: Flowchart of walk generation and evaluation procedure. A broad set of initial trajectories are generated and evaluated in simulation at a range of speeds. The most promising trajectories are then evaluated at a range of stiffnesses. The compliance of the robot provides a natural exploration of the space around the reference trajectories, and the best output trajectories are extracted and used as new reference trajectories at a range of stiffness levels.

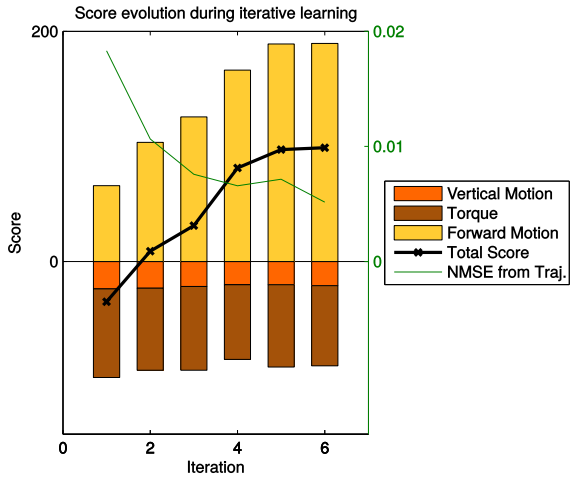



Figure 92: Score function change during trajectory learning in simulation. Normalised Mean Square Error between the specified and actual output trajectory decreases during learning, and is usually accompanied by an increase in score.

through a joint motor will be proportional to the torque it is exerting, and $E = I^2Rt$. The figure therefore nominally represents a scaled indicator of the electrical energy usage, divided by the distance travelled.

From the simulation results, trajectories were selected which produced a total score $S$ greater than zero, and therefore produced reliable forward motion. The walking speed and stiffness level are plotted against the energy cost per distance travelled, $C_{EPD}$ in Fig. 94. Each cell of the grid in this figure shows the mean of the lowest 10% of results by $C_{EPD}$, with a minimum of 3 sample points being averaged. If a cell contained less than 3 samples, results are not shown. This figure shows that there are certain speeds at which the robot walks more efficiently, but also that changing the stiffness level can have a very significant effect on walking efficiency.

Fig. 94 shows that in the simulations, the effect of changing stiffness varies depending on the walking speed. At low speeds, decreasing the stiffness has little, or even a negative effect on energy efficiency. However, at faster, more dynamic speeds, decreasing the stiffness can significantly reduce the energy expenditure. Indeed, the most efficient locomotion is produced at low stiffness values.

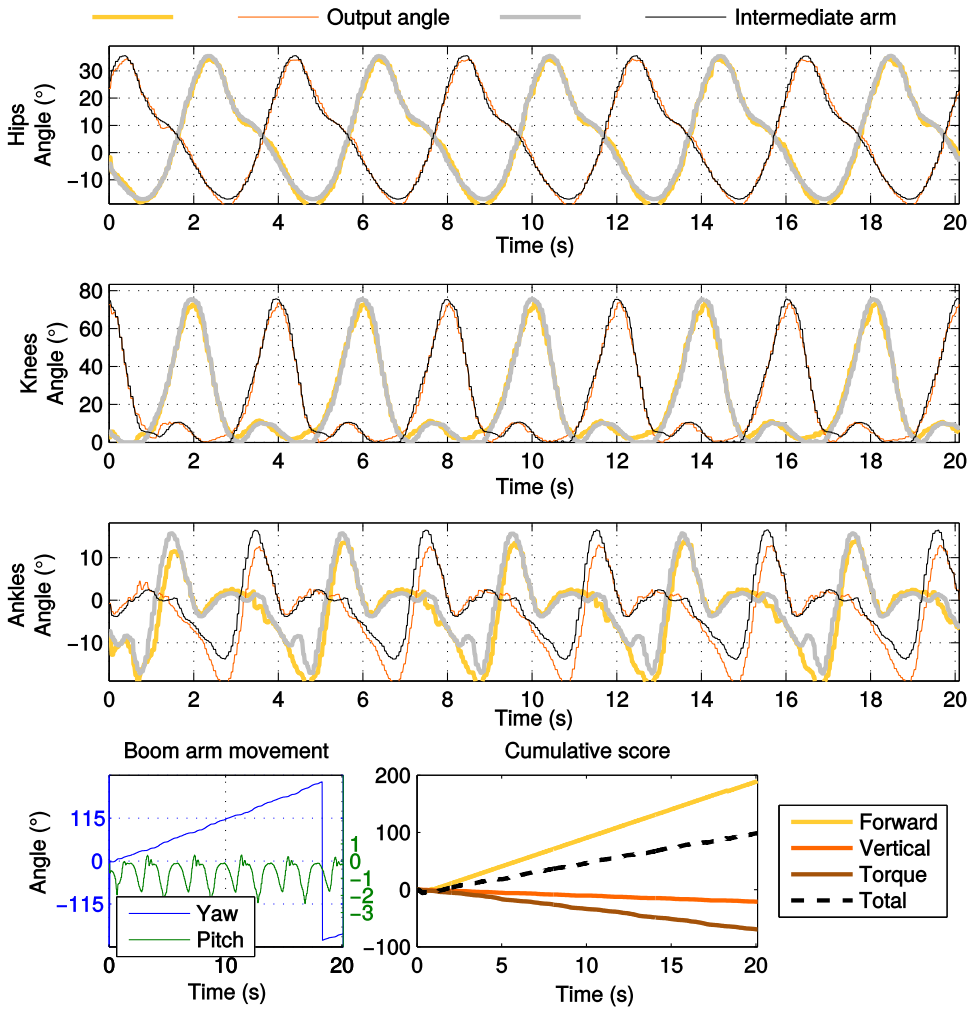



Figure 93: Generated trajectory in simulation, after trajectory learning. The movement of the boom arm indicates a continued forward motion, reflected in the evolution of the score for the walk.

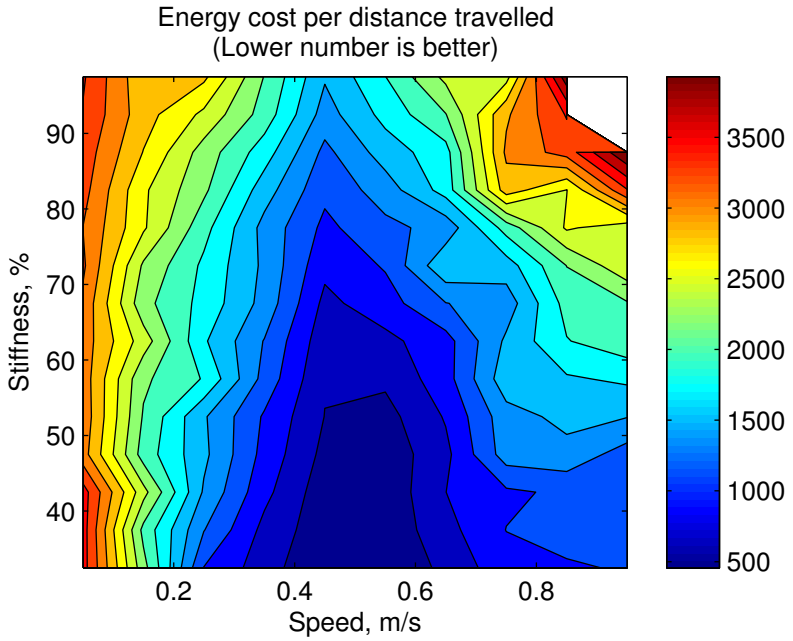


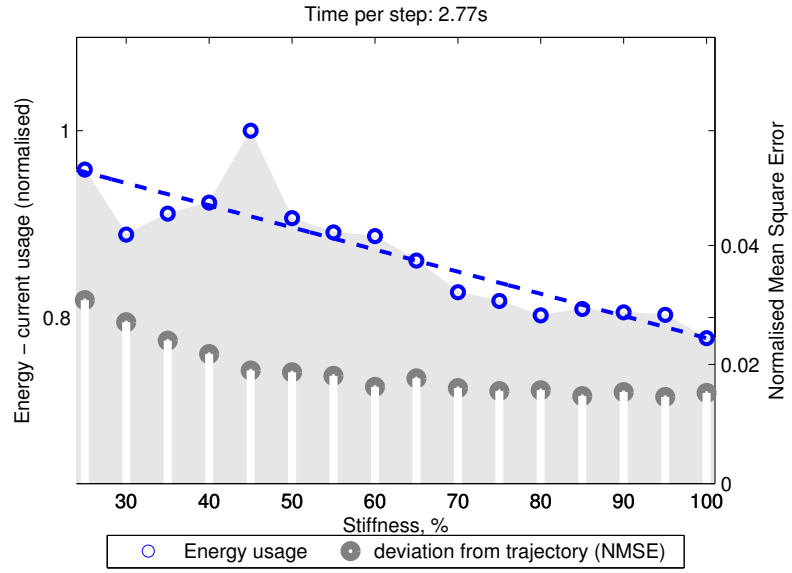

Figure 94: Evaluation of generated trajectories at different speeds and stiffness levels in simulation. The contour plot shows the energy used per distance travelled (according to Eq. 89), and shows that generally, lowering stiffness leads to a decrease in energy usage.

### 5.3.3 *Evaluation in Hardware*

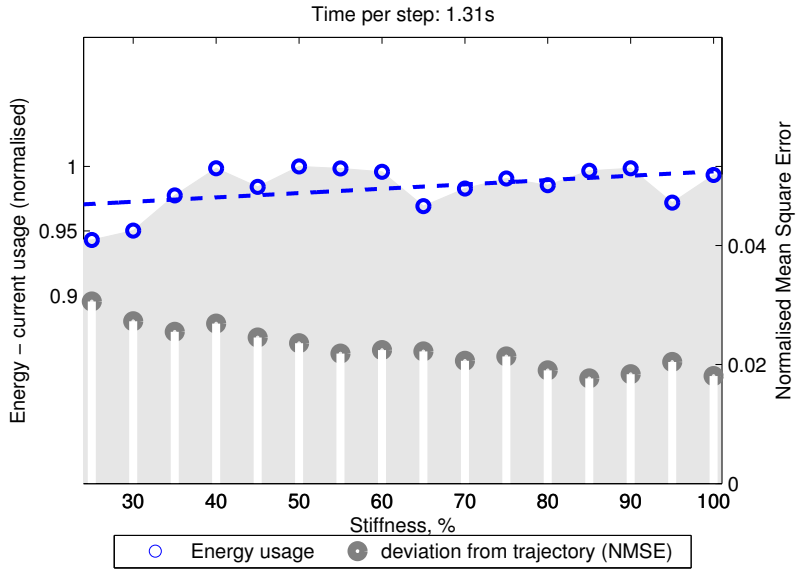
#### 5.3.3.1 *Replicating a trajectory at different stiffnesses*

To investigate the effect of changing stiffness on the electrical energy used by the robot, the same trajectory was produced at different stiffness levels. While the robot was performing each trajectory, the digitally filtered current reading for each drive motor was read continuously, and integrated to produce an overall energy usage for the robot. This energy usage represents a linear scaling of the electrical energy used by the robot for that trajectory at that speed and stiffness level.

Fig. 95 shows the results from performing the same nominal walking trajectory at stiffness values from 25% to 100% of maximum stiffness. The stiffness values for each joint are set to the same percentage of their maximum stiffness - with the hips therefore being stiffer than the knees and ankles. The graphs show the total energy usage, normalised to 1, and also show the error in the output trajectory (from the desired output trajectory) at each stiffness. In general, the error is greater at lower




(a) Very slow speed




(b) Faster speed

Figure 95: Electrical energy usage Vs. stiffness for the same nominal walking trajectory at different stiffnesses. The energy usage is shown, as is the error between the desired trajectory and output trajectory, for each stiffness.

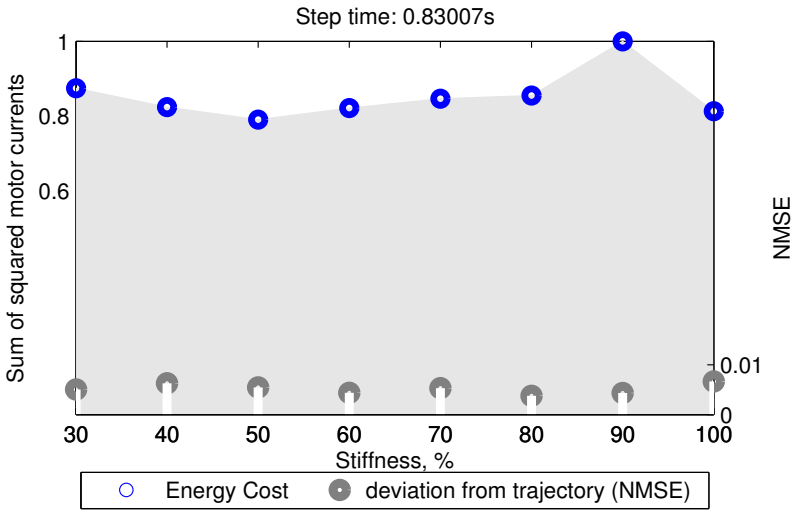



Figure 96: Actual robot walking at a faster speed (0.83s/step) with different stiffness values. The overall electrical energy usage, however, does not vary as predicted with stiffness value.

stiffness values. For these graphs, each data point represents three runs on the robot, with the results averaged.

Fig. 95a shows the results for the trajectory being performed at a very slow speed, which basically represents a point-to-point movement task for the robot. The trajectory was linearly interpolated to ensure smooth commands were sent to the robot, and make sure that the effect was not just down to jerky motion on the robot. As can be seen from this graph, at this very slow speed, decreasing the stiffness generally results in an increase in the overall energy consumption for performing that motion.

However, as can be seen from Fig. 95b, at faster speeds this effect is not seen. In actuality, decreasing the stiffness is, at this speed, linked to a slight decrease in overall energy usage. This is beneficial not only because it decreases the power consumption of the robot, but also because it makes the robot less susceptible to shocks and other disturbances at this more dynamic speed.

Fig. 96 shows walking with a step period of around 0.8 seconds. The graph shows the mean of the two iterations for each stiffness level which most closely matched the reference trajectory. The electrical energy used by the motors does not vary as significantly as stiffness is varied.

The decrease in energy usage expected from the simulation results did not occur, and this could be for a variety of reasons. We speculate that the additional movement of the compliant mechanism could be causing additional frictional losses. At lower stiffness levels, the drive motors must move significantly more, and so a more detailed model of motor electrical usage may also ameliorate some of the disparity.

Similarly for very slow tasks, on the robot we see that reducing the stiffness appears to be associated with an increase in motor current consumption. At these speeds, there is no dynamic benefit from decreasing the stiffness of the robot, and in fact it may be harder to assume precise positions and reduce oscillations in the robot. At lower stiffness values, the motors must move much more, and additional frictional losses will also be incurred in the variable stiffness mechanisms themselves, which deflect much more.

These results indicate that at realistic walking speeds, there is no electrical energy penalty associated with decreasing stiffness. This is useful as during these highly dynamic motions, a decreased stiffness can help the robot conform to unexpected disturbances, and reduces shock loading when impacts occur.

## 5.4 CONCLUSIONS

In this chapter, the ability of BLUE to recreate human like heel-toe locomotion has been demonstrated, both in simulation and on the real robot. A parameterisation of the walking cycle was developed, utilising gait transition events as well specified key poses, which can define a walking trajectory when accompanied by timing information. This method was utilised to create a walk generator for BLUE, which requires only step length and clearance height to generate a kinematically sound walking trajectory, which closely resembles a human walking trajectory but also takes into account the kinematics of BLUE.

The behaviour of the robot was iteratively adjusted automatically until it produced these specified trajectories, and it was confirmed that the trajectories does indeed produce locomotion at a range of speeds, with some speeds being more energy efficient than other speeds.

With a system that can produce trajectories with to specified parameters, and which can adjust the behaviour of the compliant robot to produce those trajectories, the

effect of varying stiffness was explored in simulation and on the hardware. In simulation, the results showed that it is possible to produce the walking behaviour at a variety of fixed stiffness levels, and that at more dynamic speeds, reducing the stiffness should be associated with a decrease in the motor torque levels, and the theoretical energy cost using a very basic motor model. In simulation the results also showed that the robot was most efficient when walking at a speed of around 0.5m/s.

Selected results from the simulation were then repeated on the actual robot. Where it was shown that the iterative trajectory adjustment can reproduce walking trajectories at a range of stiffness levels.

However, in our setup, a penalty was associated with reducing the stiffness level - the motor current usage increased. This meant that at slow speeds, where there was predicted to be little difference in energy usage, there was actually a considerable increase. At faster speeds, the predicted improvement in energy was not observed, although there was no negative effect of reducing the stiffness, and in some cases a slight positive benefit.

The discrepancy between the simulation and the robot could be caused by additional frictional losses observed at lower stiffnesses, and also due to the simplicity of the motor model used for the simulations. Additionally, the controller may be struggling to reproduce the trajectories exactly at low stiffness levels, and may not be generating the most efficient control trajectory.

In future work, higher quality variable stiffness mechanisms and higher efficiency motors and gearing could be used to ameliorate some of these issues. Additionally, better torque sensing in the joints would give a clearer idea of the discrepancy between simulation and real life.

While further experimentation is needed in this area, our initial results show that there is an effect on the energy consumption of the robot as stiffness is varied. The effect on the hardware is not as useful as that seen in simulation, but for dynamic walking we showed that lower stiffness levels can be used without any energy penalty. At the same time, these lower stiffness levels would better allow the robot to naturally adjust to disturbances in the terrain, and let it absorb shocks better, reducing the risk of damage to the robot and potentially also the environment.

For a multifunctional bipedal robot, there may therefore be a benefit to including variable stiffness mechanisms.

# 6

# PARAMETER BASED TELEOPERATION OF BLUE

From the work presented in the last chapter, we have a system which can generate trajectories with a specified step length and clearance height, and can then evaluate them at a range of stiffness settings. The output from these experiments is stored, so that it is possible to associate a "best" stiffness setting with a particular trajectory at a particular speed, i.e. one that results in a lower energy cost. For a given walking speed, there will be several trajectories which can produce this speed, i.e. with more, shorter steps or fewer, longer steps. Again, we can learn the energy cost associated with each of these trajectories.

Teleoperation - in this case the remote control of the robot by a user wearing a motion sensing suit - is a useful task to consider for BLUE. As humans are still much better than robots at planning in complex environments, we envisage a scenario when a user will utilise a telepresence system to control a humanoid robot - controlling the motion of the robot by moving in a motion sensing suit, and "seeing" through the eyes of the robot via a virtual reality headset.

In other teleoperation tasks, the goals are generally clear - for example the user can control directly the output positions of the end effectors in order to perform a task. For locomotion, the goal is to produce the same movement as the user, but the specific joint angles used are less important. We should aim to match the speed of the user as well as their trajectory. It may be beneficial to also copy the step height of the user, as they may adapt to different terrains - for example grass vs. tarmac.

In a telepresence system, it is advantageous to make the motion of the robot as close to the motion of the user as possible, in order to reduce the motion sickness that the user feels. For this reason, and the additional reason that the user may be selecting a step length on purpose based on the terrain, it may also be desirable to copy the step length of the user.

A system which extracted parameters from the walk of a user in a motion sensing suit, in order to effect a behaviour of the robot which reproduced these parameters,

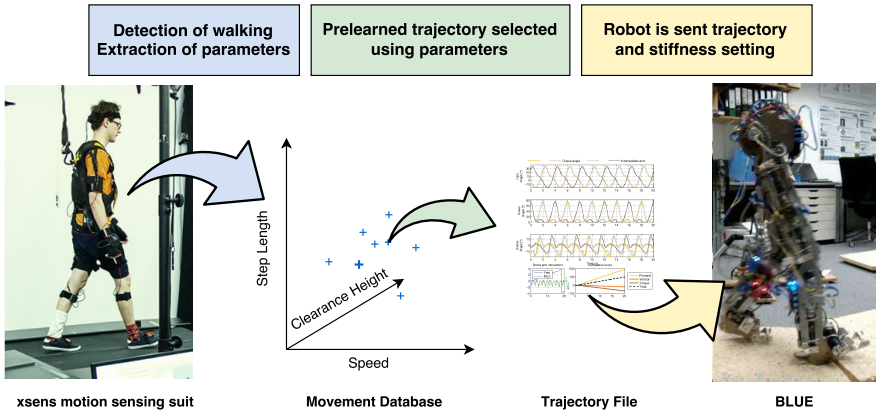



Figure 97: Overview of the parameter based teleoperation system for BLUE. Salient parameters are extracted from the walk of the user, and copied by the robot, without copying the specific joint angles. The robot will also assume the best stiffness value from prior evaluation of the selected trajectory.

was therefore developed. The system detects when the user is walking, calculates the speed, step length and phase, and searches for a pre-learned, pre-optimised behaviour for the robot to perform, which will reproduce these parameters. The result is that, from the user's point of view, the robot performs the same motion as them, however the robot is adopting trajectories and stiffness settings which are more suitable to it, without requiring further input from the user.

A schematic of the system operation is shown in Fig. 97, illustrating how high level parameters are extracted from the user's walking movements, and then matched to learned trajectories according to one or more of step length, speed, and clearance height. Once a trajectory has been matched it is sent to the robot, along with the prelearned best stiffness value for this trajectory. If the user changes their motion, the system will match to a new trajectory, and instruct the robot accordingly.

In this system, there is a delay associated with the identification of the user action, and our system lags one step behind the user's actions. In the future, additional work could be done to see how early in the gait cycle it is possible to predict the step length of the user, and attempt to match their walk in real time.

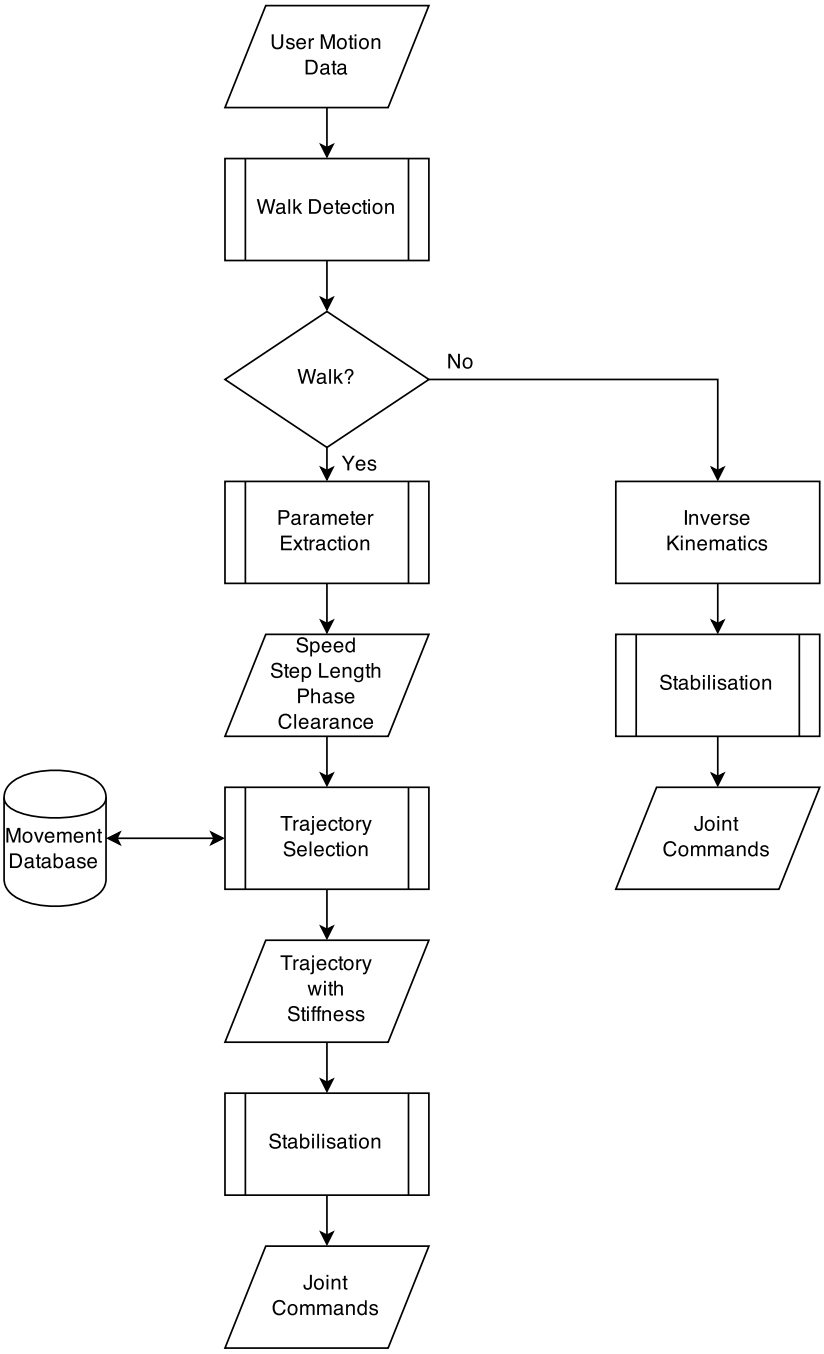



Figure 98: Architecture of the teleoperation system

## 6.1 SYSTEM ARCHITECTURE

Fig. 98 shows the basic operation of the teleoperation system. Fundamentally, the system has two distinct operational modes - discrete or rhythmical movements. For the moment, we limit the rhythmical movements to walking, although in the future the framework could be expanded to other rhythmical movements - e.g. running, skipping, or moonwalking.

For the teleoperation, the user wears the xsens motion tracking suit, as shown in Fig. 81. The data representing the user's motions is transmitted in real time over a network link, from the Windows PC running the MVN Studio software. Once the data is received, it is parsed by a walk detection algorithm, which determines whether or not the user's recent movements constitute a walking behaviour.

By default, and if the walking detection function does not detect a walking motion, the system attempts to copy the user's behaviour by aiming to achieve the same foot positions. This is done by inverse kinematic analysis from the user's foot positions and orientations, giving joint angles to command BLUE. The flowchart in Fig. 98 shows a stabilisation step next. Presently, this just constitutes a smoothing function to ensure that the robot's behaviour is not too jerky. However, in the future, and especially if the teleoperation system were to be utilised on a full three dimensional robot, this stabilisation function could include balancing control, to ensure the robot does not fall.

If a walking motion is detected, the system next aims to extract the parameters from the walk. These parameters are what will be copied by the robot, and consist of step length, walking speed, clearance height and phase. With these parameters, a selection algorithm selects the closest matching trajectory from the movement database, in order to try and provide the closest recreation of the user's movements. The movement database contains data from walking at a variety of stiffness levels, and so the selected trajectory is associated with a particular stiffness level, and the robot assumes this stiffness level as it begins the selected trajectory. In the future, this stiffness could be explicitly specified as a parameter to be used when selecting a trajectory. A stabilisation step occurs, although as with the discrete movements, for the time being this simply reduces the jerk of the robot's joints.

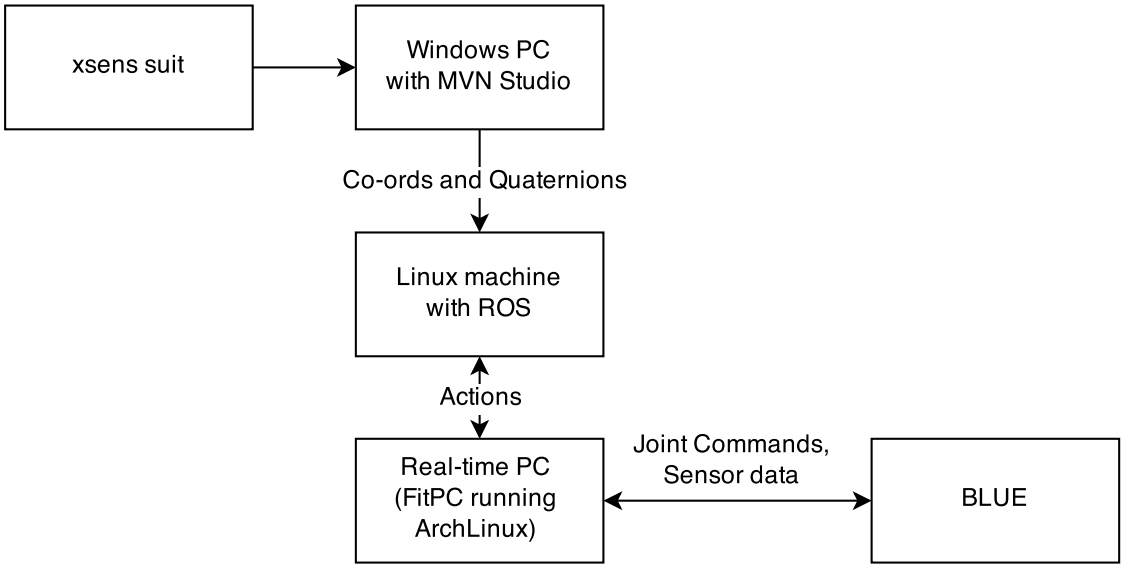



Figure 99: Teleoperation system hardware

The movement database itself contains a number of pre-evaluated trajectories for the robot, each of which produces reliable walking movements with a specified speed and step length. The trajectories are evaluated offline, in a similar way to that detailed in Section 5.3.3, in order to determine the best stiffness level to use for that trajectory. Since the parameter space is quite large, we concentrated on the area of likely human walking speeds and step lengths, as shown in Fig. 83. We are still in the process of assembling the movement database, as we test trajectories on the hardware.

The walk detection and parameter extraction routines are run on each new data point which arrives at the control PC, and the robot will attempt to match changes in the user's walking parameters. Similarly, if the user stops walking and begins discrete movements, the robot will transition back to discrete motion mode.

## 6.2 TELEOPERATION CONTROL FRAMEWORK

The teleoperation control framework involves the co-ordination of several PCs, in part due to the necessity to use a Windows PC to run the xsens software and collect the user's motion data. The FitPC is dedicated to running real-time control of the robot, and we therefore utilise a Linux machine, in between the Windows PC and the

| Code | Command | Parameters |
|---|---|---|
| 0xF0 | STOP | |
| 0xF1 | SET POS | flags, time, 6 x joint angle |
| 0xF2 | SET STIFF | flags, time, 6 x stiffness |
| 0xF3 | RUN TRAJECTORY | flags, time, trajectory ID, number of steps, polarity |
| 0xF4 | WALK | flags, time, step length, speed, polarity |
| 0xF5 | GET STATUS | |
| 0xFF | PING | |

Table 11: Command set for real-time PC

real-time PC, to extract walking parameters, and perform high level control of the robot.

### 6.2.1 *Network control scripts*

As can be seen from the block diagram in Fig. 99, a socket connection is utilised to allow the real-time PC and the Linux PC to communicate. To facilitate this, a special client script was written, to allow the real-time PC to operate as a server, issuing joint trajectories to the robot in real time, whilst accepting high level commands from a remote PC.

#### 6.2.1.1 *High level robot control commands*

To define exactly the behaviours which the real-time PC will have to execute, a high level command set was developed. This included commands for setting whole-robot positions and stiffnesses, as well as instructions to run specific trajectories, and get information from the robot. This command set is shown in Table 11, along with the parameters which the real-time PC expects to receive with each command.

Many of the commands have a *time* parameter, which gives a delay in milliseconds until the real-time PC should execute that command.

##### STOP

The *stop* command will cause the real-time PC to immediately issue a command turning off all motors.

#### SET POS

The *set pos* command is used to instruct the robot to assume a discrete position. The parameters chiefly consist of a set of joint angles, representing a pose for the robot. The real-time PC will automatically act to smooth the trajectory between the robot's current state and the desired pose, to minimise jerk. A flag, set in the *flags* parameter, allows this smoothing behaviour to be turned off, for example when the robot is under discrete teleoperative control, and is receiving a continuous stream of joint angles.

#### SET STIFF

The *set stiff* command is similar to the set pos command, except it commands a set of stiffness settings, one for each joint of the robot.

#### RUN TRAJECTORY

If the Linux PC has selected a particular trajectory from the movement database to be performed on the robot, it can simply send the ID of that trajectory to the real-time PC, to action that trajectory on the robot. The *polarity* parameter allows the Linux PC to specify whether the trajectory should start with the left foot swinging or the right foot swinging, and can be used, for example to get the robot to match human behaviour.

#### WALK

If a particular trajectory has not yet been selected, but the walk parameters are known, these can be sent to the real-time PC, to allow it to select the most appropriate walk. This increases the modularity of the system, since it allows the real-time PC to be issued high level commands from a device which does not have a copy of the movement database.

#### GET STATUS

The *get status* command instructs the real-time PC to return information on the status of the robot and the control system. This information includes the present positions of the robots joints, their deflections from equilibrium, their commanded positions, and electrical current readings.

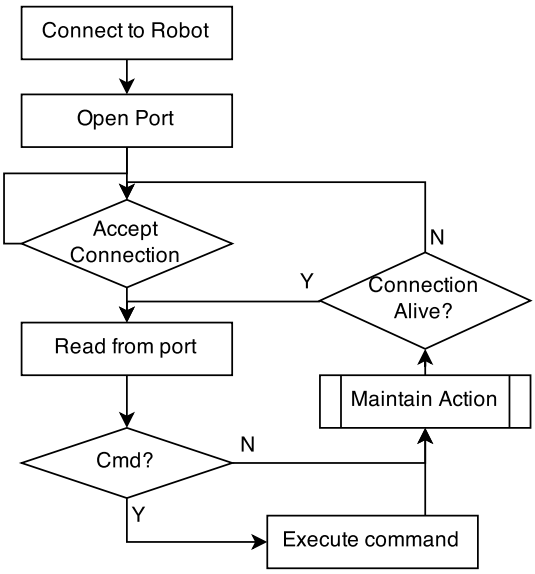



Figure 100: Block diagram of network control server script

PING

The *ping* command allows the Linux PC and the real-time PC to check communication with each other, and check latency. The real-time PC will respond immediately to a *ping* command.

#### 6.2.1.2 *Network control server script*

With the desired commands specified, the special client script was written for the real-time PC, to allow it to operate as a server, receiving high level commands, constructing trajectories, and commanding these in real time. A block diagram of the operation of this program is shown in Fig. 100.

The server script opens a non-blocking socket connection, allowing it to check for incoming commands and data without blocking a main control loop. The program has structs for storing command information, and maintains an action queue, as well as storing the previous command. When an action is interpreted, the script generates the trajectory steps into a deque, with each step including a nominal time to instruct that trajectory step. On each iteration through the control loop, the program will check if there are instructions in the trajectory queue waiting to be executed, and will evaluate the present time against the nominal step execution time, to determine whether that command should be issued.

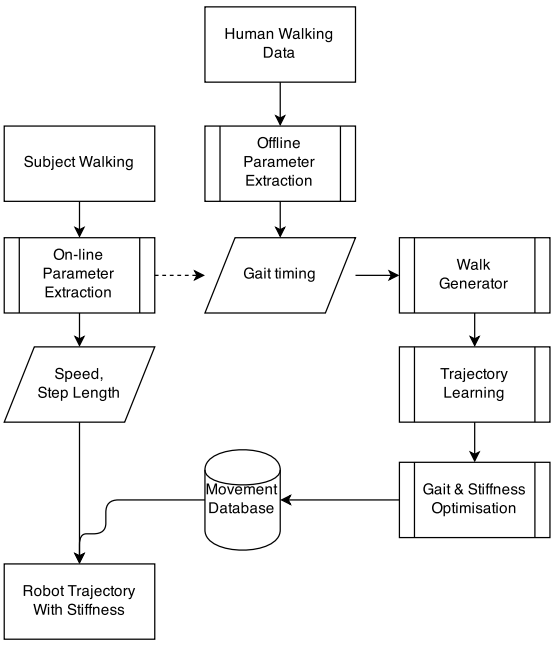



Figure 101: Flowchart of movement database utilisation

The Maintain Action step of Fig. 100 is responsible for ensuring that the instructed trajectories are performed, with exact timings as specified in the trajectory files. Longer actions, such as walks with a specified number of steps, are broken down into individual steps. Specified trajectories are loaded step by step, ensuring that the polarity of the walk is appropriately set - and the assignment of stance/swing legs is correct. The use of time information allows actions to be queued, or to allow newer actions to subsume older actions at an appropriate point (usually the feet adjacent transition), or to tune the timing of an operation, for example to attempt to walk in synchrony with a user.

### 6.2.2 *Movement database*

The movement database is still being constructed, but when we perform an experiment on BLUE, the log files are automatically saved, and can be evaluated for the relevant parameters and costs or scores, and added to the database.

When searching for a match in the database, it is necessary to consider which of the three parameters are most important - speed, step length, and clearance height. If only speed is important, then the system will just have several trajectories to choose

from, and will select the one with the best score (energy cost per distance travelled + high torque penalisation). In general, selecting one, two or three parameters will lead to the description of a plane, line, or point in parameter space.

Searching in a local neighbourhood, the particular trajectory is selected based on minimisation of the function $\delta^2 C$, where $C$ is the cost associated with that trajectory, and $\delta = K_{speed}\delta_{speed} + K_{step\_length}\delta_{step\_length} + K_{clearance\_height}\delta_{clearance\_height}$ is a measure of the distance from the specified plane, line, or point in parameter space. The $K$ terms relate to a weighting of the relative importance of matching each parameter.

### 6.2.3 *ROS control and interfacing*

The Linux machine shown in Fig. 99 serves as the control centre for the teleoperation system. It is responsible for receiving and interpreting the xsens data, selecting a matching trajectory, and instructing the real-time PC. The machine runs a ROS (Robot Operating System) core, to aid the integration of the various components. A ROS node opens a socket connection to the Windows PC and receives the xsens data, before processing it into joint angles and publishing it.

A second ROS node maintains a socket connection to the real-time PC, and is responsible for sending the commands specified in section 6.2.1.1. Presently this node also processes the human joint angle data, performs the walk detection and parameter matching, and selects a trajectory to perform. In the future, the modularity of the system will be increased, to reduce the number of tasks that this second node performs.

## 6.3 PARAMETER EXTRACTION AND WALK DETECTION

Presently, a simple state machine is used for detecting when a user is walking. As the joint angle data comes in, the co-ordinates of the legs are analysed to look for the gait transition events as specified in section 5.1.2. If the system detects a sequence of transitions that describes a walk (i.e. Feet Adjacent $\rightarrow$ Tibia Vertical $\rightarrow$ Initial Contact $\rightarrow$ Toe/Ball off), then it determines that a walking step has taken place, and the parameters of the step (step length, speed, clearance height) are determined from the

kinematics. It is necessary to set the system with the segment lengths of the user, otherwise default values are assumed.

This system has the issue that it always lags one step behind the user, due to the time taken to identify and parameterise the step. It would be far better to be in sync with the user, and hence in the future we will undertake analysis to determine how quickly into a gait cycle it is possible to determine that a stepping motion is taking place, and predict the parameters from it.

## 6.4 SUMMARY: TELEOPERATION WITH OPTIMAL STIFFNESS

While it is still being fine tuned, the system outlined here provides a method for allowing a user to control the movement and locomotion of a variable impedance bipedal robot in a very natural way. When a rhythmical walking motion is detected, the system automatically switches from giving discrete instructions, to recruiting an appropriate pre-learned walking trajectory for the robot. This is similar to the way in which humans can control voluntary motions with the motor cortex, or can recruit CPG networks to execute learned movements including locomotion.

The system also has the benefit that when locomotion is occurring, the robot can assume joint trajectories and stiffness values which provide the best known energy efficiency for it. Such trajectories are likely to be more stable and more energy efficient than attempting to copy joint angles or foot movements directly from the user. Whilst the stiffness of the user is not measured, this does not affect the ability of the robot to select an appropriate stiffness value for the selected trajectory.

# Part IV

# A NOVEL MORPHOLOGY

A diminutive digression in a familiar field

# 7

# A BIPEDAL ROBOT WITH 3DOF LEGS

## 7.1 MOTIVATION

I am a great fan of walking robots, and I believe that they are a great way to inspire and engage with people who are interested in learning about robotics, programming, mechanism design, or any related field. During the course of this PhD, emphasis was placed on designing robots which can reproduce the amazingly efficient human walking cycle as closely as possible. The robots produced are ideal for research, and are comparatively inexpensive as research robots, however they are far too large and expensive to be general educational tools.

In this chapter, a design is presented for an ultra low cost miniature bipedal robot, which is free standing and capable of three dimensional movement. The chief criterion for this design was to try and produce a highly functional walking robot - i.e. one which can perform a variety of behaviours - utilising as few motors as possible, and as cost effectively as possible.

## 7.2 INTRODUCTION

Many designs for robots attempt to mimic the human leg, usually with 3 DoF hips, 1 DoF knees and 2 DoF ankles (e.g [77], [88]). This results in six actuators per leg, and the need to design an arrangement of these actuators which can reproduce spherical joints as closely as possible. Such designs are the most capable, but are more generally more expensive than designs which require fewer actuators, and pose a significant challenge to the designers who must fit all the required actuators and electronics into the legs.

Alternatively, some designs for legged robots seek to reduce the number of actuators used to produce locomotion. For example, certain passive-dynamic walkers are capable of producing locomotion at a given speed and with a predefined gait, using

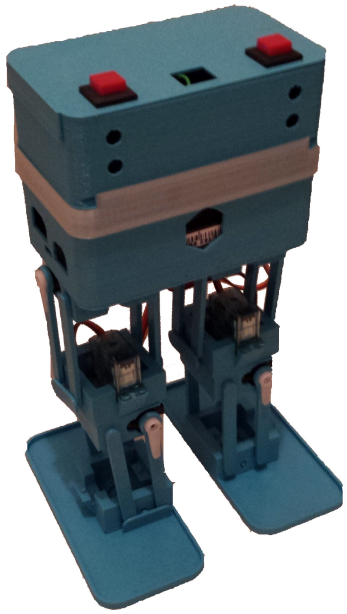


Figure 102: A small bipedal robot comprising two 3 DoF legs and a prismatic joint in the torso. The robot is capable of three dimensional locomotion, and uses the weight of the battery in the upper box in lieu of an upper body to aid balance.

only one motor [13]. Other robots use specially designed linkages (for example an eight bar linkage [1]) to force each foot of the robot to follow a path which resembles a bipedal walking pattern. However, these robots are capable of fewer behaviours than a full DoF robot.

Other designs utilise four bar linkages to allow flexible motion while retaining a parallel relationship between components [32][10][64]. For example, the robot CHARLI-L reduces the number of motors required by one, having 5 DoF per leg[33].

In this chapter, a design is presented for a leg which utilises only three actuators, in tandem with a pair of four-bar linkages. We present the kinematics for the design, which are simplified by the fact that the four-bar linkage based design keeps the feet of the robot parallel to the body. A design for a bipedal robot is introduced, built on two of the 3 DoF legs, and with a prismatic joint in the body to help balancing.

We show that such a robot is not only capable of producing a walking gait, but can produce a wide variety of behaviours, from walking with different step widths, to turning around one leg or turning while walking. A small implementation of such a robot has been built - a 20cm high self contained three dimensional biped constructed from 3D printed parts.

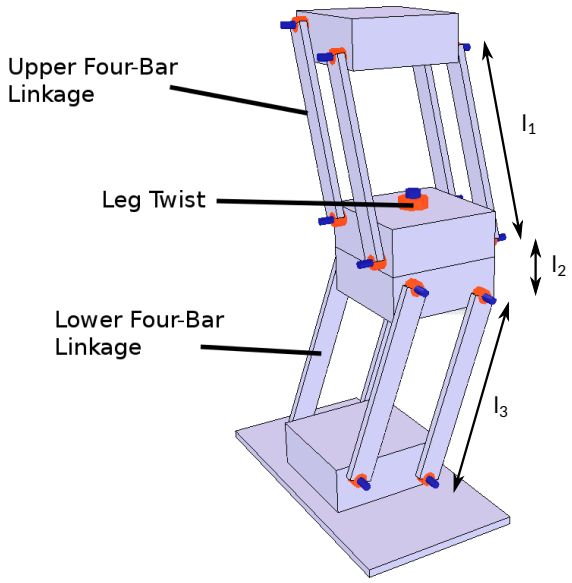



Figure 103: Kinematic design of the 3 DoF leg, comprising two four bar linkages with a nominal 90 degree twist at the knee. Leg rotation is enabled by a "twist" motor between the two four bar linkages at the knee.

## 7.3 THREE DOF LEG DESIGN

In order to enable a wide range of behaviours, each leg must be capable of moving in three dimensions, and must be able to produce a rotation in order to let the robot turn easily. In our design, we also seek to minimise the complexity of the construction by separating the motors as much as possible. A schematic of the kinematic design is shown in Fig. 103.

As can be seen from the schematic, a key feature of the design is that the two four-bar linkages have a nominal 90 degree rotation between them. Thus, in this example schematic, the upper four-bar linkage nominally produces motion in the sagittal plane (flexion/extension), and the lower four-bar linkage nominally produces motion in the coronal plane (adduction/abduction). Due to the nature of the four-bar linkages, both joints produce vertical motion.

Motion in the horizontal plane (internal/external rotation) is enabled by the twist motor which sits at the knee, between the two four bar linkages. In this way, a large amount of motion is possible, albeit with the caveats that there is a coupling between vertical and horizontal motions, and that the foot is maintained parallel to the body.

The kinematics for this design are as follows, starting from the hip, the first four-bar linkage operates as a rotation about the y-axis. The linkage ensures that the orientation frame remains unchanged, and can be considered as a rotation about the y-axis, a translation of the link length, and then a rotation back about the y-axis by the opposite amount:

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & -l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -l_1\sin\theta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_1\cos\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{90}$$

The upper knee position is therefore $\{-l_1 \sin\theta, 0, -l_1 \cos\theta\}$, where $\theta$ is the rotation of the upper four-bar linkage, and $l_1$ is the link length of this linkage.

There is then a translation by the distance between the upper and lower knee segments, a 90 degree rotation about the z-axis at the knee, and the twist joint modulates this angle:

$$\begin{bmatrix} 1 & 0 & 0 & -l_1\sin\theta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -l_1\cos\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(90+\psi) & -\sin(90+\psi) & 0 & 0 \\ \sin(90+\psi) & \cos(90+\psi) & 0 & 0 \\ 0 & 0 & 1 & -l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(90+\psi) & -\sin(90+\psi) & 0 & -l_1\sin\theta \\ \sin(90+\psi) & \cos(90+\psi) & 0 & 0 \\ 0 & 0 & 1 & -l_2 - l_1\cos\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{91}$$

The lower knee position is therefore $\{l_1 \sin\theta, 0, l_2 + l_1 \cos\theta\}$, where $\psi$ is the twist angle at the knee, and $l_2$ is the distance from the upper to the lower knee. The lower

four-bar linkage then produces further arc motion. In the same manner as for the upper four-bar linkage, this produces translation, but not rotation. The transformation from the hip to the foot is therefore:

$$\begin{bmatrix} \cos\Psi & -\sin\Psi & 0 & -l_1\sin\theta \\ \sin\Psi & \cos\Psi & 0 & 0 \\ 0 & 0 & 1 & -l_2 - l_1\cos\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & l_3\sin\phi \\ 0 & 0 & 1 & -l_3\cos\phi \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{92}$$
$$= \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 & -l_3\sin\Psi\sin\phi - l_1\sin\theta \\ \sin\Psi & \cos\Psi & 0 & l_3\cos\Psi\sin\phi \\ 0 & 0 & 1 & -l_3\cos\phi - l_2 - l_1\cos\theta \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where $\Psi = 90 + \psi$. And so, the position of the foot relative to the hip is $\{-l_3\sin\Psi\sin\phi - l_1\sin\theta, l_3\cos\Psi\sin\phi, -l_3\cos\phi - l_2 - l_1\cos\theta\}$, where $\phi$ is the rotation of the lower four-bar linkage, and $l_3$ is its link length.

The robots simulated and implement in this chapter utilise two legs of the above design, along with one additional DoF in the torso, which helps with counterbalancing the robot. This joint is a prismatic joint operating in the coronal plane, and moves the upper body (which contains the battery in the real robot) above the stance leg during statically stable walking.

The specific leg implementation illustrated above is conducive to easing the mechanical design of the legs. The motor for the upper four-bar linkage can be either in the hip or the upper knee. The motor for the leg twist can be in the upper or lower knee, and the motor for the lower four-bar linkage can be positioned in the lower knee or in the ankle. In our constructed robot, there is one motor in the hip, the twist motor in the upper knee, and the lower four-bar linkage motor in the lower knee.

Alternatively, a similar principle can be used to produce different, but still viable, leg designs with 3 DoF. For example, the twist motor could be positioned in the hip or ankle, and the 90 degree rotation between the two four bar linkages fixed. If the twist motor was positioned in the hip, it would be possible to use one motor to rotate both legs, thereby using a total of five motors for two legs. Such layouts are illustrated in Fig. 104.

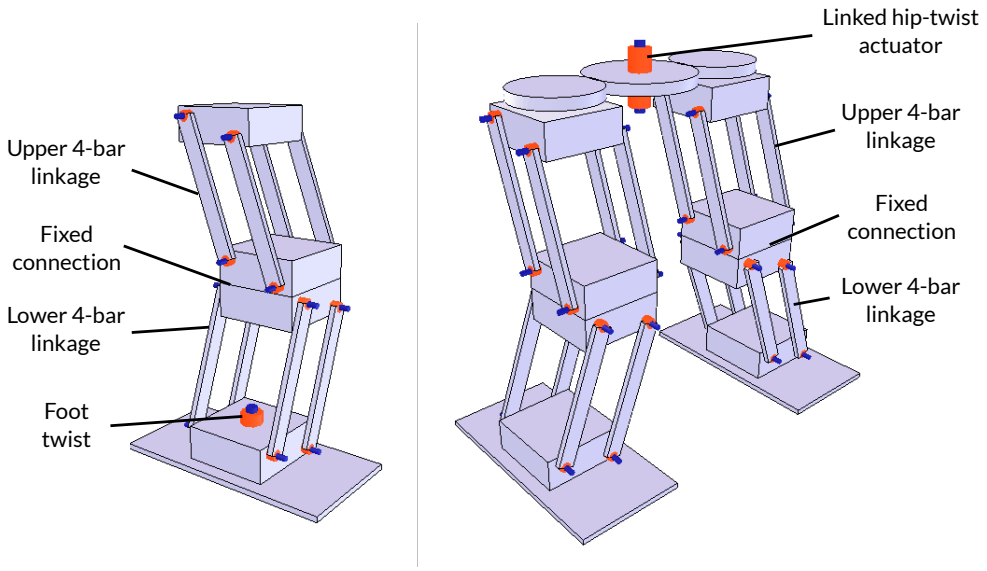



Figure 104: Alternative kinematic designs based around the principle of two four-bar linkages with a nominal 90 degree twist. One design uses only five motors between both legs.

| **Link / Segment** | |
|---|---|
| $l_1$ | 250 |
| $l_2$ | 60 |
| $l_3$ | 250 |
| Leg Spacing | 350 |

Table 12: Link lengths for the simulated robot, in mm

## 7.4 ROBOT DESIGN AND SIMULATION

V-REP was utilised to produce dynamic simulations of a robot based around two 3 DoF legs and prismatic torso joint. The link lengths and masses of the simulated robot are given in Tables 12 and 13.

V-REP allows the simulation of closed loop kinematic chains - the dummy-dummy link functionality was used to simulate the four-bar linkages. The simulation was configured to utilise an external controller script, which in turn could set the joint angles, or read a trajectory file and play it back on the simulation. We also embedded

| Segment | Mass (kg) |
|---|---|
| Upper body | 3.0 |
| Lower body | 1.0 |
| Box | 1.0 |
| Link | 0.2 |
| Foot | 0.8 |

Table 13: Segment masses for the simulated robot

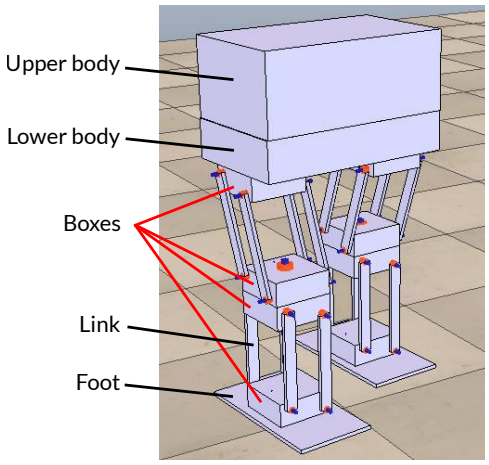



Figure 105: Segments of the simulated robot

force sensors at the connection points between the body and each leg, and at each ankle.

To prove that the robot is capable of walking and producing other motions, we first simulate statically stable trajectories. Snapshots of walking straight forward are shown in Fig. 106. As can be seen, the robot shifts its weight above the stance leg, before using the lower four bar linkage to lift the swing leg outwards and upwards. The upper four bar linkages are then used to produce flexion/extension of the legs, and move the whole robot forwards. The upper four-bar linkages are moved to equal and opposing angles, such that the step length is given by:

$$L = 2l_1 \cos\theta \tag{93}$$

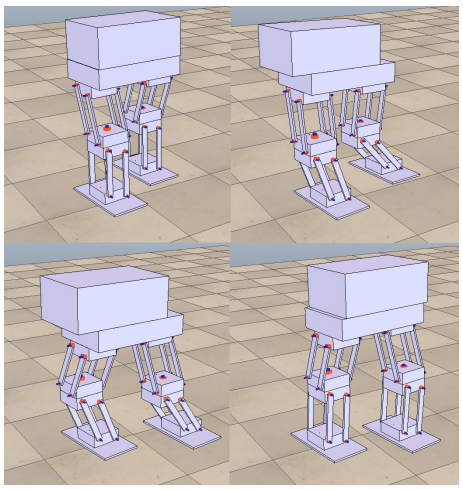


Figure 106: Snapshots of a statically stable forward walk, during dynamic simulation in V-REP

Where θ is the angle of the upper four-bar linkage. The simulated robot can produce a variety of step sizes from zero to beyond 350mm, although the vertical motion of the body does increase with step size.

The robot is also capable of turning about one leg, as is illustrated in Fig. 108, where the robot rotates 45 degrees around its right leg. These motions can be combined to produced curved walking paths, for example as shown in Fig. 107.

The robot is capable of lifting its legs quite high, and so, provided balance can be achieved, should be capable actions such as climbing stairs.

## 7.5 MINIATURE BIPEDAL ROBOT

Whilst the dynamic simulations in V-REP show that the design of leg proposed above should be capable of producing locomotion and discrete movements, giving such a robot a great deal of behavioural flexibility, such simulations are never 100% accurate. We therefore constructed a small robot with the same design, to test movement in the real world.

The robot was designed to be as simple and quick to manufacture as possible, and it utilises off the shelf parts alongside a 3D printed chassis. The parts were designed to be modular, allowing for easy reconfiguration of the design if required. The parts for the robot are shown laid out in Fig. 109. Each four bar linkage consists of two box parts, which are designed to fit around standard micro-servos. The "driving" box part

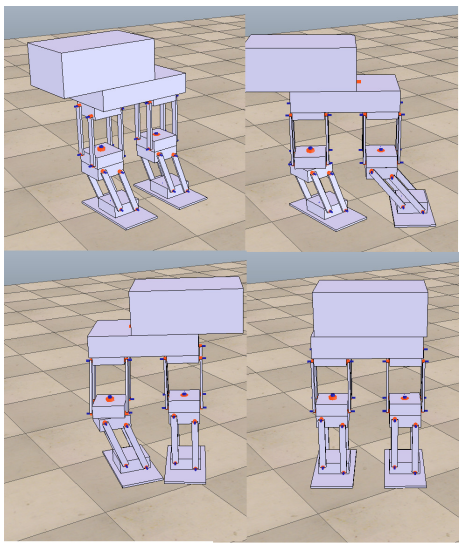


Figure 107: Snapshots of a statically stable turn about one leg. In this case the robot rotates 45 degrees about the leg.

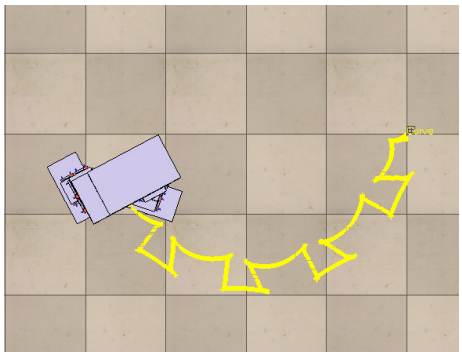


Figure 108: Simulation of walking while turning. Twisting movement was applied to the right leg during its stance phase, to make the robot gradually turn right as it walked.

Figure 109: Most of the parts required to build the robot, including BeagleBone Black computer and micro servos. The plastic parts are 3D printed and designed to be modular.

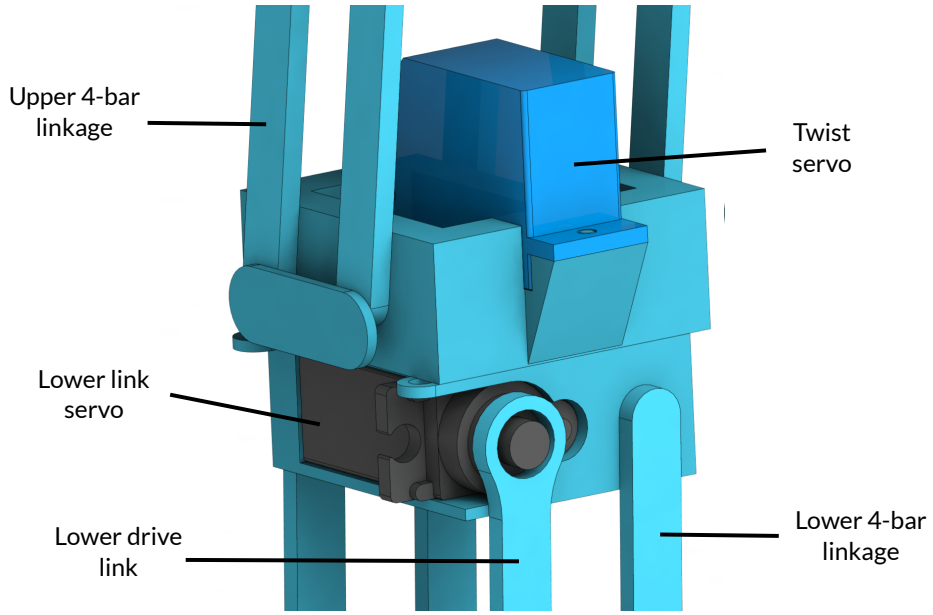



Figure 110: Close up render of the knee joint of the robot. The upper box part holds the four bars of the upper linkage in place, and serves as a mount for the twist servo. The lower box part tightly encases the drive servo for the lower four-bar linkage. Since the resolution of the 3D printer does not allow an accurate servo spline to be produced, the drive link has a large bore which is designed to fit around a standard servo horn.

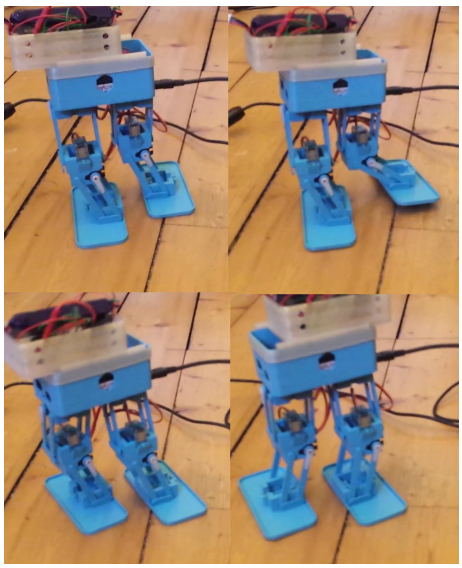


Figure 111: Robot Walking with a statically stable walking gait.

exposes the output spline of the servo and has three bores for rotating links, while the "slave" box simply has four bores for rotating links. Each segment of the robot comprises four links and two box parts - three of the links are identical, whilst the fourth is designed to communicate with the servo output.

As shown in Fig. 110, a modified box part carries the leg twist motor, with a thrust bearing surface taking the loading when the robot is standing on that leg. The output from the leg twist motor connects directly to the top of the box which forms the upper part of the lower four bar linkage. A foot part attaches in a similar way to the bottom of the lower box of the lower four-bar linkage.

The hips of both legs attach to the lower body part, which is a box sized to snugly fit a BeagleBone Black embedded computer. Two guide rails and a rack and pinion gearing mechanism for a prismatic joint between this lower body part and an upper body part, which holds the battery and voltage regulation circuitry. A LiPo battery is used, with RC BECs to step down and regulate the power for the servos and for the BeagleBone Black.

Currently, we utilise an $I^2C$ PWM breakout board to produce the servo pulses, however since the robot only comprises seven motors, it would be possible to use only the PWM outputs of the BeagleBone Black. Regardless, including all the electronics, motors, and plastic, the total cost of the robot is less than £100. The most expensive component of the robot is the BeagleBone Black, but using this means that the robot

has a 1GHz Arm Cortex A8 computer running linux, and this allows the robot to have a high degree of autonomy and expandability. For example, the robot would support the addition of a webcam, giving this low cost robot a vision system. Additionally, there are a large number of available GPIO ports, which allow the addition of various sensors, LEDs, etc.

The constructed robot stands 200mm tall, to the top of the upper body part. The lengths of the links in the four-bar linkages is 55mm.

To validate that the physical robot is indeed capable of reproducing the kinds of movements demonstrated in simulation, we produced a similar walking trajectory to that shown above in simulation, on the robot. The result can be seen in Fig. 111, which shows the robot taking one step. The robot is able to walk successfully, in this case taking a step every two seconds.

The servos in the robot are capable of moving much more quickly than this, and in the future we will generate walking trajectories which produce dynamically stable behaviour, allowing the robot to move much more quickly. We have also produced locomotion on the robot without the upper part of the body attached i.e, without the moving torso as counterbalance - however for this the robot had to be externally powered since the battery was no longer on-board.

## 7.6 SUMMARY

In this chapter, we have introduced a design for a robot leg which uses only three motors, and yet can produce distinct motion in the sagittal, coronal, and horizontal planes. This allows a robot constructed with such legs to retain a high degree of behavioural flexibility, even with the reduced number of motors.

A key feature of this design is that each leg consists of two four-bar linkages, with a nominal 90 degree twist between them at the knee of the robot. This permits motion in the sagittal and coronal planes, whist keeping the foot parallel with the torso, and thereby eliminating the need for motors in the ankle joint. The relocation of the leg rotation motor to the knee simplifies the mechanics of the robot, allowing more space in around the hips and body of the robot to be utilised for electronics, etc. Although, as we have shown, alternative designs are possible, for example where the twist motors are located at the hips, or where one twist motor rotates both legs.

The novel mini-bipedal robot, constructed using the 3DoF leg design, and using a chassis built entirely from 3D printed parts, represents a very cost effective, but highly capable robotic platform. We demonstrated that whilst this is an extremely low cost robot, it is capable of producing locomotion, and therefore show that this leg design is a way to enable flexible legged locomotion using only half the number of motors often employed in bipedal robot legs. Furthermore, it carries on it a full computer running Linux, and thus is not just a toy, but a highly functional and expandable platform.

# Part V

# CONCLUSIONS AND FUTURE WORK

# 8

# CONCLUSIONS & FUTURE WORK

This thesis has concentrated on the design and construction of novel bipedal robots, primarily ones which have the ability to vary the impedance of their joints. The robots BLUE and miniBLUE were designed specifically for producing human-like locomotion, and have a foot design heavily based on human foot morphology, in order to let them produce and utilise the heel-toe walking gait which humans use.

There are not many bipedal robots which have three part compliant feet (e.g. [69][34]), with the majority using flat, stiff feet to simplify balancing and control. For passive dynamic walkers, rounded feet are often used to provide a rolling motion[13], but such feet can make it difficult to perform other tasks - especially slower or static ones. Other robots such as MABEL[30] use point feet to simplify ground contact - again not good for slow or static behaviours, although this is simplified on a planar biped. In this work we have produced a robot with a three part compliant foot, and shown that it can be used to produce human-like heel-toe walking at dynamic speeds. In addition to this it is also capable of supporting the robot while it performs slow, statically stable movements such as very slow, small steps.

Variable impedance has become a very large field in recent years[98], and when constructing a robot which will utilise variable impedance joints, it is difficult to know where to start. Before beginning construction of BLUE or miniBLUE, a full survey of the field was undertaken, and each published mechanism evaluated and compared. The modified AwAS[44] mechanism utilised in BLUE and miniBLUE was judged to give the best combination of energy efficiency, stiffness range, and the ability to cope with high torques. When compared to pre-existing variable stiffness bipeds[97][93], our robot is capable of producing faster and more human-like movements.

Working at a lab which has previously primarily dealt with software forced some creative thinking on the manufacture of the robots. Techniques such as waterjet cutting and 3D printing were utilised in order to reduce costs and reduce the time taken to produce the robots. A method for combining these two manufacturing techniques

to create hybrid parts has also been detailed. The use of 3D printing for creating structural parts with a tailored compliance was also investigated. In miniBLUE, the 3D printed foot arch performs exactly as predicted by FEA analysis, deforming just enough to hold the weight of the entire robot before contacting a hard stop. The foot arch of BLUE, which is under significantly higher loading, exhibited a softer profile than predicted, but was still capable of dealing with very large loads. We have therefore demonstrated how 3D printing can be used for structural components even in larger robots.

Of the robots themselves, BLUE was perhaps the first bipedal robot to be designed to be able to independently vary the stiffness and damping of its joints. Due to a lack of time we have not explored the variable damping functionality during this work, and it is hoped that a future student will develop this further.

miniBLUE was designed to be a highly modular platform for walking experiments. As well as starting to introduce three dimensional motion - the robot has variable impedance hip adduction and abduction - it was designed to allow different types of variable stiffness actuators to be evaluated. As mentioned earlier, a printing defect in the chassis has limited the usefulness of the robot so far, but it is hoped that the necessary parts can be reprinted soon to enable the robot to be fully functional.

In addition to creating novel mechanical platforms, this work has also resulted in the development of a low cost, high performance, modular control system for the robots. The system shows that Ethernet connected control boards can be used to produce low latency communications with refresh rates in excess of 1kHz. Furthermore, in contrast to communication systems like EtherCAT, our system uses only standard Ethernet hardware, and is capable of multi-master communications. This greatly expands the functionality of the system, allowing control boards to initiate corrective action if an error condition is encountered, or for example to allow the implementation of local reflexive loops, in an architecture similar to the human peripheral nervous system.

To produce human-like walking on BLUE, the heel-toe gait cycle was analysed in some detail, and a parameterisation developed which utilises four transition points in the gait cycle. It was shown that just from these poses and the associated timing information, angle data for a whole walking cycle can be generated. To expand this to BLUE, a parameterised representation of each transition pose was developed, based

on the kinematics of BLUE, and requiring very little tuning data apart from the timing of the transitions, and a specified step length and clearance height. It was shown, first in simulation and then on the hardware itself, that gait trajectories generated by this system produce consistent locomotive behaviour on BLUE, across a range of speeds. Although we have empirically evaluated these trajectories, in the future, they could be used as the input to a dynamic stabilisation and movement generation algorithm (e.g. [62]), and improved further.

Preliminary work has also begun on characterising the effect of varying stiffness during these walking motions. Using dynamic simulation in V-REP, we found that the robot should be capable of using the generated trajectories to produce locomotion, at a variety of speeds and stiffness levels. Our simulation results indicated that at very slow speeds, there would be no real benefit to changing the stiffness level, but that at faster and more human-like walking speeds, decreasing stiffness could be associated with a reduction in motor torques and energy usage.

In our initial experiments to try to replicate these results on the hardware, where we can actually measure the current through the motors to calculate their energy expenditure, we found that in our implementation reducing the stiffness seemed to incur an energy penalty. For slow movements this means that decreasing stiffness actually increased energy usage, while for faster movements no great change in energy levels was observed. We speculate that one reason for this is increased frictional losses in the variable stiffness mechanisms, and because the drive motors have to move more.

Nevertheless, our initial results show that it is possible to have a robot that can be stiff at slow speeds and soft at faster speeds. While the stiffness reduction at dynamic speeds did not have the predicted energy benefit, it does still make the robot less sensitive to external disturbances, and less likely to be damaged in the event of a collision.

In contrast to the majority of methods for generating movement trajectories and providing dynamic stability for bipedal robots (e.g. [101] [23], our control method does not require a detailed and accurate model of the robot - instead needing only a kinematic model and a platform on which to experiment. Using dynamic simulation the many potential trajectories were evaluated at different speeds to ascertain which were most likely to be viable on the robot. On the robot itself, we were able to

produce locomotion without needing to explicitly model factors such as backlash or unexpected joint friction. This is especially useful on a real robot, where performance may change or degrade over time.

From these experiments, we are amassing a collection of validated walking trajectories on BLUE, for each of which we know the energy expenditure, the step length, the speed, and the nominal clearance height. We also have previously utilised an xsens motion sensing system for capturing human walking data, and developed techniques for extracting walking cycle parameters from this data.

Combining these features together, we have built a teleoperation system for BLUE, whereby a user can walk naturally in the motion sensing suit, and BLUE will initiate a trajectory that best matches the parameters of that walk, for example speed and step length. Since the specific trajectory has already been evaluated on the robot, the robot can also assume the best stiffness value for that trajectory, and our teleoperation system therefore provides a simple mechanism for movement and stiffness control of the robot.

In addition to this a novel design for reducing the number of motors required in a leg was detailed. The design allowed the production of a bipedal robot which is capable of three dimensional motion, but which has only three motors in each leg. The constructed robot is also designed to be very low cost, i.e. less than $100 including the onboard Linux based PC, in order to make it an ideal tool for education. In the future, I would like to investigate using variable impedance joints with such a design, perhaps on a larger version of the robot designed for use in an office or workplace environment.

A large part of this work has concentrated on the production of systems which are modular and expandable platforms for research. The robots built and the control infrastructure developed allows inexperienced users to quickly get started on the robots. Preliminary work has shown that variable impedance can have a significant effect on energy efficiency, as well as affecting how robust the robot is to shocks and disturbances. In the future, it is hoped that work on the robots will continue, perhaps to include stabilisation and further trajectory optimisation on BLUE, and to expand this work into three dimensions and onto miniBLUE.

## 8.1 FUTURE WORK

In the future, there are many avenues which could be explored to take this work further. In the first instance, running additional experiments on the hardware and gathering a larger data set would be useful. The disparity between simulation and hardware should be investigated, initially by developing a more detailed model of motor energy usage on the simulation side. This would allow a more useful direct comparison between the energy usage of each of the joints of the robot compared to the predicted usage.

An expected source of at least part of the disparity is additional frictional losses in the variable stiffness mechanisms. In future robots, higher precision manufacturing of these mechanisms may allow such losses to be reduced. Additionally, the levels of backlash on the real robot may be introducing complications, as when changing direction the drive motors must first take up slack before even beginning to drive the joint. Future robots should use locking collars or other non-keyed transmissions.

The variable damping functionality also remains to be investigated, but could potentially be of great use to help reduce oscillations[58] without requiring action from the drive motor through the variable stiffness element.

The work in this thesis has focussed on using fixed stiffness levels through the joint cycle, but in future more dynamic stiffness trajectories should be investigated. Varying stiffness during movements can confer dynamic benefit (e.g. [7]), and if the platform supports it should be investigated. Note that changing stiffness under loading will almost certainly require additional energy input from the stiffness adjustment actuator, and this should be factored in when setting the specifications for a stiffness adjustment motor. Optimisation methods[21] could be used to investigate more dynamic stiffness profiles.

Another avenue of research would be to combine the cyclical trajectories and control methods currently used with reflexive control. [27][52] This could improve the ability of the robot to adapt to disturbances .

Regarding the low-cost, low-DoF bipedal robot detailed in chapter 7, a proof of concept robot has been made, but the concept could be taken much further. In particular, it would be of interest to add variable stiffness mechanisms, or at least compliance to the robot. The design of the robot - using four bar linkages - means that it would be

very simple to integrate a variable stiffness mechanism, as well as potentially adding assistive springs in parallel with the joints. The robot would then become a very easy to make variable impedance 3D biped.

This thesis has shown that it is possible to produce human-like walking on a variable impedance biped at a variety of stiffness levels and speeds. The use of human-like morphology with trajectories designed heavily based on human ones allows for very dynamic motions. It is hoped that this proof of concept can be expanded to further the exciting field of variable impedance robotics.

Part VI

# APPENDIX

A

# MEASURING MULTIPLE ROTATIONS

In certain situations it can be necessary to measure the absolute position of a rotating shaft over several revolutions. For example, if the shaft drives a leadscrew, such a system would allow the determination of the output of the linear stage without the need for a linear position sensor or a calibration stop. Alternatively, it may be impractical to have a position sensor at the final output link of a mechanism, and the position of a shaft earlier up the gear chain may need to be measured instead - for example if a packaged servo-motor were to be used in a situation where it is connected to an external stage of gearing.

In this system, two rotary position sensors are used, nominally on the shaft itself, and on the shaft directly before it in the gear train. The gear ratio between them should be $n : o$, where $n$ is not an integer multiple of $o$, and $n$ is greater than $o$. This arrangement means that two encoders produce a combined reading which repeats only every $m/n$ revolutions, where $m$ is the lowest common multiple of $n$ and $o$.

For example, Fig. 112 shows the rotational positions of two shafts which are connected to each other through a 35:10 gear ratio. As can be seen, the combined positions of the two shafts only loops every two revolutions of the first shaft.

This concept is further illustrated in Fig. 112, which shows the readings from encoders on shafts with a 35:10 gear ratio plotted against each other. The readings repeat every second revolution. Furthermore, the system allows for noise rejection, since we know that the combined reading must be constrained to one of the data points shown on the graph.

Fig. 114 shows the same graph for encoders with a 32:10 gear ratio, which results in a combined reading which repeats every five revolutions of the output shaft.

Commonly, sensors which measure absolute positions over several rotations either require a calibration step on power on, or operate by gearing down the input to an absolute position sensor, resulting in a drop in resolution, and accuracy loss due to

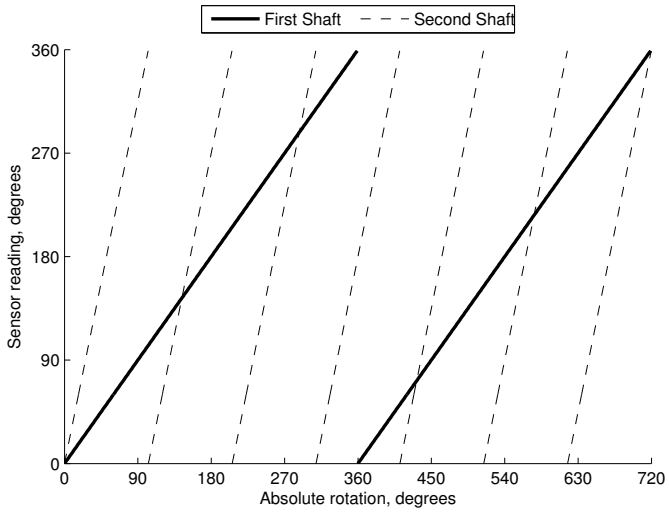



Figure 112: Rotational positions of first and second shafts geared through 35:10 ratio

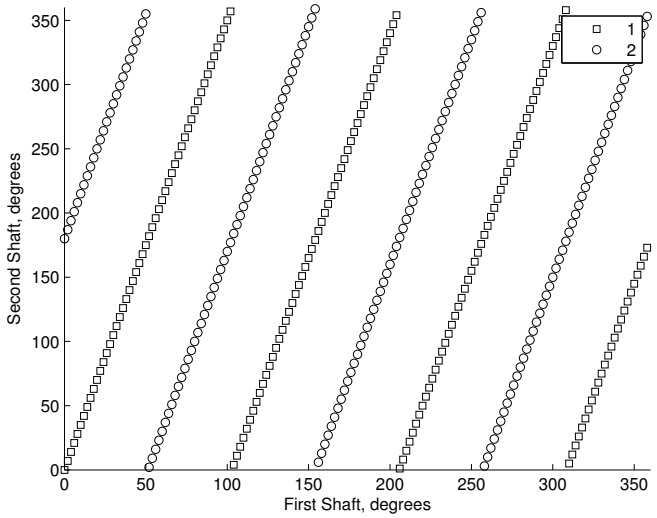



Figure 113: Plot of the reading from the first encoder vs. the reading from the second encoder, for a 35:10 gear ratio

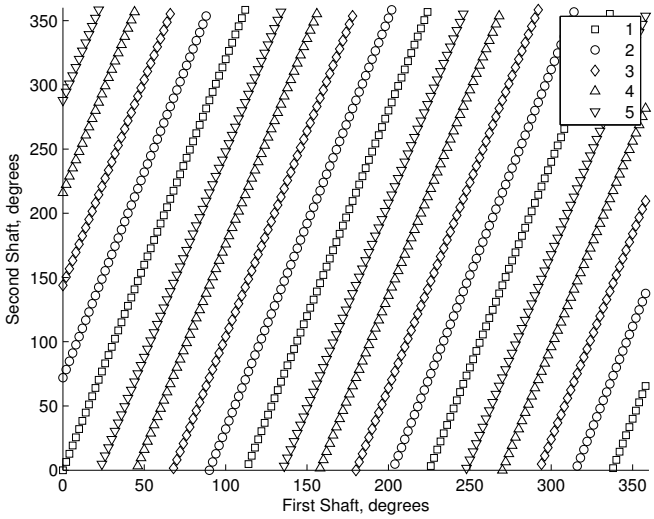



Figure 114: Reading from the first encoder vs. reading from second encoder, for a 32:10 gear ratio.

backlash etc. The system presented here allows the instantaneous determination of an absolute position over several rotations, with a high resolution.

# B

## CONTROL BOARD COMMANDS

As specified earlier, the request frame of the robot communication system takes the following format:

| Byte Offset | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | Destination Address | | | | | |
| 6 | Source Address | | | | | |
| 12 | Length | | Subsystem | Command | Flags | Sequence Number |
| 18 | Sequence Number | Subsystem and command specific data | | | | |
| ... | | | | | | |

Each subsystem has a number of commands associated with it, and a brief overview of the commands implemented so far is given below. The system is designed to be very extensible, to allow the addition of additional functionality later, if required. For the commands below, the flags and sequence number are left blank, as they will be set according to task requirements. The flags parameter can be used to set whether or not an acknowledgement is required for a command, and the sequence number from a request frame is returned in the reply frame, to allow matching of requests and responses.

*Subsystem 0x01: ADC*

| | Cmd | Flags | Sequence Number | | Command Specific Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6 + 2n** | **7 + 2n** | ... |
| **Read Sensors** | 0x80 | - | - | | No. of channels | | Channel ID | | ... |
| | Relays data back from the sensors, for example ADC or magnetic rotation sensors.<br>Command specific data is:<br>number of channels (N), and then a list of N channel IDs, specifying which sensors to read.<br>Response will return the readings from these sensors in the same order.<br>• (uint16) Number of channels<br>• (uint16) Channel *n* ID | | | | | | | | |

## *Subsystem 0x02: Motor*

| | Cmd | Flags | Sequence Number | Command Specific Data | |
|---|---|---|---|---|---|
| **Byte** | 0 | 1 | 2 | 4 | - |
| **Motor Enable** | 0x01 | - | - | Motor flags | - |
| | Enables one or more motors<br>• (uint8) Motor flags - Motors to be enabled are indicated by setting bits<br>e.g. 0x03 would enable motors 0 and 1 | | | | |
| **Byte** | 0 | 1 | 2 | 4 | - |
| **Motor Disable** | 0x02 | - | - | Motor flags | - |
| | Disables one or more motors<br>• (uint8) Motor flags - Motors to be disabled are indicated by setting bits<br>e.g. 0x03 would disable motors 0 and 1 | | | | |
| **Byte** | 0 | 1 | 2 | 4 | - |
| **Motor Brake** | 0x04 | - | - | Motor flags | - |
| | Brakes one or more motors. This will also disable affected motors, if they are enabled.<br>• (uint8) Motor flags - Motors to be braked are indicated by setting bits<br>e.g. 0x03 would brake motors 0 and 1 | | | | |
| **Byte** | 0 | 1 | 2 | 4 | - |
| **Motor Unbrake** | 0x08 | - | - | Motor flags | - |
| | Unbrakes one or more motors<br>• (uint8) Motor flags - Motors to be unbraked are indicated by setting bits<br>e.g. 0x03 would unbrake motors 0 and 1 | | | | |

| **Byte** | 0 | 1 | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Set Parameters** | 0x10 | - | - | Motor ID | pos_min | pos_max | $I_{max}$ | $I_{linear}$ | def_offset | def_sign | flags |
| | Sets the parameters for a given. Command data is follows:<br>• (uint16) Motor ID<br>• (uint16) pos_min - Lower position limit<br>• (uint16) pos_max - Upper position limit<br>• (uint16) $I_{max}$ - Current limit - linear counter increased if current is over this limit<br>• (uint16) $I_{linear}$ - Current linear limit - error raised if count of linear filter over this limit<br>• (uint16) def_offset - Deflection sensor offset<br>• (int8) def_sign - Deflection sensor sign (polarity)<br>• (uint8) flags - Various flags - space for future expansion<br>• 0 (LSB): Polarity - setting to 1 flips the motor direction | | | | | | | | | | |

## *Subsystem 0x03: Damping*

| | Cmd | Flags | Sequence Number | Command Specific Data | |
|---|---|---|---|---|---|
| **Byte** | 0 | 1 | 2 | 4 | 5 |
| **Set damping** | 0x01 | - | - | cmd 1 | cmd 2 |
| | Sets damping commands of motors<br>• (uint8) Cmdn damping value from 0-100% | | | | |

## *Subsystem ox80: PID*

| | Cmd | Flags | Sequence Number | Command Specific Data | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Byte** | **0** | **1** | **2** | **4** | **6** | **10** | **14** | **18** | **20** |
| **Set Parameters** | 0x01 | - | - | PID ID | $K_p$ | $K_i$ | $K_d$ | set_min | set_max |
| | sets the parameters for a given PID loop. Command data is:<br>• (uint16) PID ID<br>• (float) $K_p$ - Proportional Gain<br>• (float) $K_i$ - Integral Gain<br>• (float) $K_d$ - Derivative Gain<br>• (uint16) set_min - minimum acceptable set position<br>• (uint16) set_max - maximum acceptable set position | | | | | | | | |
| **Byte** | **0** | **1** | **2** | **4** | **6 + 4n** | **8 + 4n** | ... | | |
| **Set Position** | 0x02 | - | - | No. of channels | $PID_n$ ID | $setpos_n$ | ... | | |
| | Changes the set positions for a number of PID loops.<br>Command data consists of a number of channels (N), followed by N PID ID/setpos pairs<br>• (uint16) Number of channels<br>  • (uint16) $PID_n$ ID<br>  • (uint16) $setpos_n$ - New target position for PID n | | | | | | | | |

## *Subsystem ox81: Error*

| | Cmd | Flags | Sequence Number | Command Specific Data |
|---|---|---|---|---|
| **Byte** | **0** | **1** | **2** | **4** |
| **Error Read** | 0x80 | - | - | - |
| | Reads the error report from the control board<br>Will retrieve a list of any errors that have occurred since the last time this function was called.<br>No command specific data is required for this command, but the reply will be as follows:<br>• (uint16) Num Errors<br>  • (uint16) Error code n | | | |

## *Subsystem ox82: Filter*

| | Cmd | Flags | Sequence Number | | Command Specific Data | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Byte** | **0** | **1** | **2** | **3** | **4** | **5** | **6 + 2n** | **7 + 2n** | **...** |
| **Read Filters** | 0x80 | - | - | | No. of filters | | Filter ID | | ... |
| | Relays data back from the filter outputs.<br>Command specific data is a number of filters (N), followed by N filter IDs.<br>The reply will consist of the output from each filter in the specified order<br>• (uint16) Number of filters<br>  • (uint16) Filter n ID | | | | | | | | |

# BIBLIOGRAPHY


[1] O Al-Araidah, W Batayneh, T Darabseh, and SM BaniHani. Conceptual design of a single dof human-like eight-bar leg mechanism. *JJMIE*, 5(4), 2011.

[2] R McN Alexander. Three Uses for Springs in Legged Locomotion. *The International Journal of Robotics Research1*, 9(2):53–61, 1990.

[3] D.J. Bennett, J.M. Hollerbach, Y. Xu, and I.W. Hunter. Time-varying stiffness of human elbow joint during cyclic voluntary movement. *Experimental Brain Research*, 88(2):433–442, February 1992. ISSN 0014-4819. doi: 10.1007/BF02259118. URL http://www.springerlink.com/index/10.1007/BF02259118.

[4] Pranav A Bhounsule, Jason Cortell, Anoop Grewal, Bram Hendriksen, JG Daniël Karssen, Chandana Paul, and Andy Ruina. Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge. *The International Journal of Robotics Research*, 33(10):1305–1321, 2014.

[5] Antonio Bicchi, Giovanni Tonietti, Michele Bavaro, and Marco Piccigallo. Variable stiffness actuators for fast and safe motion control. In *Robotics research. The eleventh international symposium*, pages 527–536. Springer, 2005.

[6] D Braun, M Howard, and S Vijayakumar. Optimal variable stiffness control: formulation and application to explosive movement tasks. *Autonomous Robots*, 2012.

[7] D. Braun, F. Petit, F. Huber, S. Haddadin, P. van der Smagt, A. Albu-Schaffer, and S Vijayakumar. Robots driven by compliant actuators: Optimal control under actuation constraints. *IEEE TRANSACTIONS ON ROBOTICS*, 29(5):1085–1101, 2013.

[8] David J Braun, Jason E Mitchell, and Michael Goldfarb. Actuated dynamic walking in biped robots: Control approach, robot design and experimental validation. *2009 9th IEEERAS International Conference on Humanoid Robots*, pages

237–242, 2009. doi: 10.1109/ICHR.2009.5379573. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5379573.

[9] DJ Braun and JE Mitchell. Actuated Dynamic Walking in a Seven-Link Biped Robot. *Mechatronics, IEEE/ASME*, PP(99):1–10, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5676222.

[10] Hyeung-Sik Choi and Yong-Heon Park. Development of a biped walking robot actuated by a closed-chain mechanism. *Robotica*, 24(01):31–37, 2006.

[11] Junho Choi, Sunchul Park, Woosub Lee, and Sung-Chul Kang. Design of a robot joint with variable stiffness. In *2008 IEEE International Conference on Robotics and Automation*, pages 1760–1765. Ieee, May 2008. ISBN 978-1-4244-1646-2. doi: 10.1109/ROBOT.2008.4543455. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4543455.

[12] Junho Choi, Seonghun Hong, Woosub Lee, Sungchul Kang, and Munsang Kim. A Robot Joint With Variable Stiffness Using Leaf Springs. *Robotics, IEEE Transactions on*, 27(99):1–10, 2011. ISSN 1552-3098. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5692845.

[13] Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. Efficient bipedal robots based on passive-dynamic walkers. *Science (New York, N.Y.)*, 307(5712): 1082–5, February 2005. ISSN 1095-9203. doi: 10.1126/science.1107799. URL http://www.ncbi.nlm.nih.gov/pubmed/15718465.

[14] F. Daerden. *Conception and realization of pleated pneumatic artificial muscles and their use as compliant actuation elements*. PhD thesis, Vrije Universiteit Brussel, Belgium, 1999. URL http://wwwtw.vub.ac.be/werk/RG/MM/Publications/thesis_a4_print.pdf.

[15] S Davis and D.G. Caldwell. The design of an anthropomorphic dexterous humanoid foot. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2200–2205. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5649756.

[16] Sarah M Donelson and Claire C Gordon. *1995 Matched Anthropometric Database of U.S. Marine Corps Personnel: Summary Statistics*. United States Amry Soldier Systems Command, Natick, 1995.

[17] Oliver Eiberger, Sami Haddadin, Michael Weis, A. Albu-Schaffer, and Gerd Hirzinger. On joint design with intrinsic variable compliance: Derivation of the DLR QA-Joint. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1687–1694. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509662.

[18] Michael F Eilenberg, Hartmut Geyer, and Hugh Herr. Control of a powered ankle-foot prosthesis based on a neuromuscular model. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 18(2):164–73, April 2010. ISSN 1558-0210. doi: 10.1109/TNSRE.2009.2039620. URL http://www.ncbi.nlm.nih.gov/pubmed/20071268.

[19] C English and D Russell. Mechanics and stiffness limitations of a variable stiffness actuator for use in prosthetic limbs. *Mechanism and Machine Theory*, 34(1):7–25, January 1999. ISSN 0094114X. doi: 10.1016/S0094-114X(98)00026-3. URL http://linkinghub.elsevier.com/retrieve/pii/S0094114X98000263.

[20] C English and D Russell. Implementation of variable joint stiffness through antagonistic actuation using rolamite springs. *Mechanism and Machine Theory*, 34(1):27–40, January 1999. ISSN 0094114X. doi: 10.1016/S0094-114X(97)00103-1. URL http://linkinghub.elsevier.com/retrieve/pii/S0094114X97001031.

[21] Tom Erez and Emanuel Todorov. Trajectory optimization for domains with contacts using inverse dynamics. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4914–4919. IEEE, 2012.

[22] Dominic James Farris and Gregory S. Sawicki. The mechanics and energetics of human walking and running: A joint level perspective. *Journal of the Royal Society Interface*, 9(66):110–118, 1 2012. ISSN 1742-5689. doi: 10.1098/rsif.2011.0182.

[23] Siyuan Feng, X Xinjilefu, Christopher G Atkeson, and Joohyung Kim. Optimization based controller design and implementation for the atlas robot in the

darpa robotics challenge finals. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 1028–1035. IEEE, 2015.

[24] Deanna J. Fish and Cheryl S. Kosta. Genu Recurvatum: Identification of Three Distinct Mechanical Profiles. *JPO Journal of Prosthetics and Orthotics*, 10(2):26–32, January 1998. ISSN 1040-8800. doi: 10.1097/00008526-199801020-00003. URL http://content.wkhealth.com/linkback/openurl?sid=WKPTLP:landingpage&an=00008526-199801020-00003.

[25] E. Garcia, JC Arevalo, G. Muñoz, and P. Gonzalez-de Santos. Combining series elastic actuation and magneto-rheological damping for the control of agile locomotion. *Robotics and Autonomous Systems*, 59(10):827–839, 2011.

[26] Amit Gefen. The in vivo elastic properties of the plantar fascia during the contact phase of walking. *Foot & ankle international / American Orthopaedic Foot and Ankle Society [and] Swiss Foot and Ankle Society*, 24(3):238–44, March 2003. ISSN 1071-1007. URL http://www.ncbi.nlm.nih.gov/pubmed/12793487.

[27] Hartmut Geyer and Hugh Herr. A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 18(3):263–273, 2010.

[28] Hartmut Geyer, Andre Seyfarth, and Reinhard Blickhan. Compliant leg behaviour explains basic dynamics of walking and running. *Proceedings. Biological sciences / The Royal Society*, 273(1603):2861–7, November 2006. ISSN 0962-8452. doi: 10.1098/rspb.2006.3637. URL http://www.ncbi.nlm.nih.gov/pubmed/17015312.

[29] Giorgio Grioli, Sebastian Wolf, Manolo Garabini, Manuel Catalano, Etienne Burdet, Darwin Caldwell, Raffaella Carloni, Werner Friedl, Markus Grebenstein, Matteo Laffranchi, et al. Variable stiffness actuators: The user's point of view. *The International Journal of Robotics Research*, 34(6):727–743, 2015.

[30] J.W. Grizzle, Jonathan Hurst, Benjamin Morris, Hae-won Park, and Koushil Sreenath. MABEL, a new robotic bipedal walker and runner. In *2009 American Control Conference*, pages 2030–2036. IEEE, June 2009. ISBN 978-1-4244-4523-3.

doi: 10.1109/ACC.2009.5160550. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5160550.

[31] Erico Guizzo and Evan Ackerman. The rise of the robot worker. *Spectrum, IEEE*, 49(10):34–41, 2012.

[32] Arnaud Hamon and Yannick Aoustin. Cross four-bar linkage for the knees of a planar bipedal robot. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 379–384. IEEE, 2010.

[33] Jeakweon Han and Dennis Hong. Development of a full-sized bipedal humanoid robot utilizing spring assisted parallel four-bar linkages with synchronized actuation. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 799–806. American Society of Mechanical Engineers, 2011.

[34] Kenji Hashimoto, Yuki Takezaki, Kentaro Hattori, Hideki Kondo, Takamichi Takashima, Hun-ok Lim, and Atsuo Takanishi. A study of function of foot's medial longitudinal arch using biped humanoid robot. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2206–2211. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5650414.

[35] J. Hass, JM Herrmann, and T Geisel. Optimal mass distribution for passivity-based bipedal robots. *The International Journal of Robotics Research*, 25(11):1087, 2006. doi: 10.1177/0278364906072449. URL http://ijr.sagepub.com/content/25/11/1087.short.

[36] Hugh Herr and Ari Wilkenfeld. User-adaptive control of a magnetorheological prosthetic knee. *Industrial Robot: An International Journal*, 30(1):42–55, 2003. ISSN 0143-991X. doi: 10.1108/01439910310457706. URL http://www.emeraldinsight.com/10.1108/01439910310457706.

[37] J H Hicks. The mechanics of the foot. IV. The action of muscles on the foot in standing. *Acta anatomica*, 27(3):180–92, January 1956. ISSN 0001-5180. URL http://www.ncbi.nlm.nih.gov/pubmed/13338973.

[38] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of Honda humanoid robot. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, pages 1321–1326, 1998. doi: 10.1109/ROBOT.1998.677288. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=677288.

[39] Koh Hosoda, Yuki Sakaguchi, Hitoshi Takayama, and Takashi Takuma. Pneumatic-driven jumping robot with anthropomorphic muscular skeleton structure. *Autonomous Robots*, 28(3):307–316, December 2009. ISSN 0929-5593. doi: 10.1007/s10514-009-9171-6. URL http://www.springerlink.com/index/10.1007/s10514-009-9171-6.

[40] Tsz-Ching Hsu, Chung-Li Wang, Wen-Chung Tsai, Jung-Kun Kuo, and Fuk-Tan Tang. Comparison of the mechanical properties of the heel pad between young and elderly adults. *Archives of physical medicine and rehabilitation*, 79(9): 1101–1104, 1998.

[41] Sang-ho Hyon and Gordon Chengtt. Gravity Compensation and Balancing for Humanoid Robots. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, volume 00, pages 214–221, 2006.

[42] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653, 2008.

[43] A. Jafari, N.G. Tsagarakis, and D.G. Caldwell. Awas-ii: A new actuator with adjustable stiffness based on the novel principle of adaptable pivot point and variable lever ratio. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4638–4643. IEEE, 2011.

[44] Amir Jafari, Nikos G Tsagarakis, Bram Vanderborght, and Darwin G Caldwell. A Novel Actuator with Adjustable Stiffness ( AwAS ). In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4201–4206, 2010.

[45] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1620–1626. IEEE, 2003.

[46] Kenji Kaneko, Fumio Kanehiro, Mitsuharu Morisawa, Kanako Miura, Shin'ichiro Nakaoka, and Shuuji Kajita. Cybernetic human HRP-4C. *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 7–14, December 2009. doi: 10.1109/ICHR.2009.5379537. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5379537.

[47] I.A. Kapandji, R. Tubiana, and LH Honoré. *The physiology of the joints, volume two: lower limb*. Churchill Livingstone Edinburgh, 1970.

[48] Alexandria Kappel-Bargas, Richard D Woolf, Mark W Cornwall, and Thomas G McPoil. The windlass mechanism during normal walking and passive first metatarsalphalangeal joint extension. *Clinical biomechanics (Bristol, Avon)*, 13 (3):190–194, April 1998. ISSN 1879-1271. URL http://www.ncbi.nlm.nih.gov/pubmed/11415787.

[49] Nikos Karavas, Arash Ajoudani, Nikos Tsagarakis, Jody Saglia, Antonio Bicchi, and Darwin Caldwell. Tele-impedance based assistive control for a compliant knee exoskeleton. *Robotics and Autonomous Systems*, 2014.

[50] RF Ker, MB Bennett, S.R. Bibby, and R.C Kester. The spring in the arch of the human foot. *Nature*, 325:147–149, 1987. URL http://www.nature.com/nature/journal/v325/n6100/abs/325147a0.html.

[51] B.S. Kim and J.B. Song. Hybrid dual actuator unit: A design of a variable stiffness actuator based on an adjustable moment arm mechanism. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1655–1660. IEEE, 2010.

[52] Theresa J Klein and M Anthony Lewis. A physical model of sensorimotor interactions during locomotion. *Journal of neural engineering*, 9(4):046011, 2012.

[53] L. Klenerman and B.A. Wood. *The human foot: a companion to clinical studies*. Springer Verlag, 2005.

[54] K. Koganezawa, T. Inaba, and T. Nakazawa. Stiffness and Angle Control of Antagonistially driven joint. *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pages 1007–1013,

2006. doi: 10.1109/BIOROB.2006.1639223. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1639223.

[55] Pavitra Krishnaswamy, Emery N. Brown, and Hugh M. Herr. Human Leg Model Predicts Ankle Muscle-Tendon Morphology, State, Roles and Energetics in Walking. *PLoS Computational Biology*, 7(3):e1001107, March 2011. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1001107. URL http://dx.plos.org/10.1371/journal.pcbi.1001107.

[56] M. Laffranchi, N. Tsagarakis, and DG Caldwell. A compact compliant actuator (compact (tm)) with variable physical damping. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4644–4650. IEEE, 2011.

[57] Matteo Laffranchi, Nikos G Tsagarakis, Darwin G Caldwell, and Western Bank Sheffield. A Variable Physical Damping Actuator (VPDA) for Compliant Robotic Joints. In *2010 IEEE International Conference on Robotics and Automation*, pages 1668–1674, 2010.

[58] Matteo Laffranchi, Lisha Chen, Nikos G Tsagarakis, and Darwin G Caldwell. The role of physical damping in compliant actuation systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3079–3085. IEEE, 2012.

[59] M Latash. Joint stiffness: Myth or reality? *Human Movement Science*, 12(6):653–692, December 1993. ISSN 01679457. doi: 10.1016/0167-9457(93)90010-M. URL http://linkinghub.elsevier.com/retrieve/pii/016794579390010M.

[60] K.F. Laurin-Kovitz, J.E. Colgate, and S.D.R. Carnes. Design of components for programmable passive impedance. *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, (April):1476–1481, 1991. doi: 10.1109/ROBOT.1991.131824. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=131824.

[61] C.N. Maganaris and J.P. Paul. In vivo human tendon mechanical properties. *The Journal of Physiology*, 521(1):307, 1999. URL http://jp.physoc.org/content/521/1/307.short.

[62] Ian R Manchester, Uwe Mettin, Fumiya Iida, and Russ Tedrake. Stable dynamic walking over uneven terrain. *The International Journal of Robotics Research*, page 0278364910395339, 2011.

[63] Yong Mao, Jiaxin Wang, Peifa Jia, Shi Li, Zhen Qiu, and Zhuo Han. A reinforcement learning based dynamic walking control. In *Robotics and Automation, 2007 IEEE International Conference on*, number April, pages 3609–3614. IEEE, 2007. ISBN 1424406013. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4209649.

[64] Kazuhisa Mitobe, Genci Capi, and Yasuo Nasu. Control of walking robots based on manipulation of the zero moment point. *Robotica*, 18(06):651–657, 2000.

[65] D. Mitrovic, S. Klanke, M. Howard, and S. Vijayakumar. Exploiting sensorimotor stochasticity for learning control of variable impedance actuators. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 536–541. IEEE, 2010.

[66] T. Morita and S. Sugano. Design and development of a new robot joint using a mechanical impedance adjuster. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 3, pages 2469 –2475 vol.3, may 1995. doi: 10.1109/ROBOT.1995.525630.

[67] T. Morita and S. Sugano. Development of one-dof robot arm equipped with mechanical impedance adjuster. In *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, volume 1, pages 407 –412 vol.1, aug 1995. doi: 10.1109/IROS.1995.525828.

[68] T. Morita and Shigeki Sugano. Development of an anthropomorphic force-controlled manipulator WAM-10. In *Advanced Robotics, 1997. ICAR'97. Proceedings., 8th International Conference on*, pages 701–706. IEEE, 1997. ISBN 0780341600. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=620258.

[69] Kenichi Narioka, Toshiyuki Homma, and Koh Hosoda. Humanlike ankle-foot complex for a biped robot. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 15–20. IEEE, 2012.

[70] D a Nawoczenski, J F Baumhauer, and B R Umberger. Relationship between clinical measurements and motion of the first metatarsophalangeal joint during gait. *The Journal of bone and joint surgery. American volume*, 81(3):370–6, March 1999. ISSN 0021-9355. URL http://www.ncbi.nlm.nih.gov/pubmed/10199275.

[71] Tf Novacheck. The biomechanics of running. *Gait & posture*, 7(1):77–95, January 1998. ISSN 1879-2219. URL http://www.ncbi.nlm.nih.gov/pubmed/10200378.

[72] Christian Ott, Christoph Baumg, Johannes Mayr, Matthias Fuchs, Robert Burger, Dongheui Lee, Oliver Eiberger, Alin Albu-sch, Markus Grebenstein, and Gerd Hirzinger. Development of a Biped Robot with Torque Controlled Joints. In *2010 IEEE-RAS International Conference on Humanoid Robotics*, pages 167–173, 2010.

[73] Jonghoon Park and Youngil Youm. General zmp preview control for bipedal walking. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2682–2687. IEEE, 2007.

[74] Florian Petit, Maxime Chalon, Werner Friedl, Markus Grebenstein, AA Schaffer, and G. Hirzinger. Bidirectional antagonistic variable stiffness actuation: Analysis, design & implementation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4189–4196. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509267.

[75] G.A. Pratt and M.M. Williamson. Series elastic actuators. In *Proc IEEE Int. Workshop on Intelligent Robots and Systems (IROS'95)*, pages 399–406. Published by the IEEE Computer Society, 1995. URL http://www.computer.org/portal/web/csdl/doi/10.1109/IROS.1995.525827.

[76] Jerry Pratt and Ben Krupp. Design of a bipedal walking robot. *Proceedings of SPIE*, pages 69621F–69621F–13, 2008. ISSN 0277786X. doi: 10.1117/12.777973. URL http://link.aip.org/link/PSISDG/v6962/i1/p69621F/s1&Agg=doi.

[77] Nicolaus A Radford, Philip Strawser, Kimberly Hambuchen, Joshua S Mehling, William K Verdeyen, A Stuart Donnan, James Holley, Jairo Sanchez, Vienny Nguyen, Lyndon Bridgwater, et al. Valkyrie: Nasa's first bipedal humanoid robot. *Journal of Field Robotics*, 2015.

[78] A. Radulescu, M. Howard, D.J. Braun, and S. Vijayakumar. Exploiting variable physical damping in rapid movement tasks. In *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, pages 141–148. IEEE, 2012.

[79] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, Rob Playter, and TB Team. Bigdog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress*, volume 17, pages 10822–10825, 2008.

[80] Marc H Raibert. *Legged robots that balance*. MIT press, 1986.

[81] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent asimo: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 3, pages 2478–2483. IEEE, 2002.

[82] R. Schiavi, G. Grioli, S. Sen, and a. Bicchi. VSA-II: a novel prototype of variable stiffness actuator for safe and performing robots interacting with humans. *2008 IEEE International Conference on Robotics and Automation*, pages 2171–2176, May 2008. doi: 10.1109/ROBOT.2008.4543528. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4543528.

[83] S H Scott and D a Winter. Biomechanical model of the human foot: kinematics and kinetics during the stance phase of walking. *Journal of biomechanics*, 26(9): 1091–1104, September 1993. ISSN 0021-9290. URL http://www.ncbi.nlm.nih.gov/pubmed/8408091.

[84] Hiroaki Seki, Yoshitsugu Kamiya, and Masatoshi Hikizu. SCARA Type Robot Arm with Mechanically Adjustable Compliant Joints. *2006 IEEE Conference on Emerging Technologies and Factory Automation*, pages 1175–1181, September 2006. doi: 10.1109/ETFA.2006.355192. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4178307.

[85] Koushil Sreenath, Hae-Won Park, Ioannis Poulakakis, and Jessy W Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *The International Journal of Robotics Research*, 30(9):1170–1193, 2011.

[86] G. Tonietti, R. Schiavi, and A. Bicchi. Design and Control of a Variable Stiffness Actuator for Safe and Fast Physical Human/Robot Interaction. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 526–531, 2005. doi: 10.1109/ROBOT.2005.1570172. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1570172.

[87] N.G. Tsagarakis, I. Sardellitti, and D.G. Caldwell. A new variable stiffness actuator (compact-vsa): Design and modelling. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 378–383. IEEE, 2011.

[88] Nikos G Tsagarakis, Zhibin Li, Jody Saglia, and Darwin G Caldwell. The Design of the Lower Body of the Compliant Humanoid Robot cCub. *Technology*, pages 2035–2040, 2011.

[89] Nikos G Tsagarakis, Stephen Morfey, Gustavo Medrano Cerda, Li Zhibin, and Darwin G Caldwell. Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 673–678. IEEE, 2013.

[90] Mitsunori Uemura and Sadao Kawamura. A new mechanical structure for adjustable stiffness devices with lightweight and small size. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2364–2369. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5649072.

[91] Brian R Umberger and Philip E Martin. Mechanical power and efficiency of level walking with different stride rates. *The Journal of experimental biology*, 210 (Pt 18):3255–65, September 2007. ISSN 0022-0949. doi: 10.1242/jeb.000950. URL http://www.ncbi.nlm.nih.gov/pubmed/17766303.

[92] Takuya Umedachi and Akio Ishiguro. A Development of a Fully Self-contained Real-time Tunable Spring. *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2):1662–1667, October 2006. doi: 10.

1109/IROS.2006.282121. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4058614.

[93] Ronald Van Ham, Bram Vanderborght, M Van Damme, B Verrelse, and Dirk Lefeber. MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot. *Robotics and Autonomous Systems*, 55(10):761–768, October 2007. ISSN 09218890. doi: 10.1016/j.robot.2007.03.001. URL http://dx.doi.org/10.1016/j.robot.2007.03.001.

[94] Ronald Van Ham, Thomas G Sugar, Bram Vanderborght, Kevin W Hollander, and Dirk Lefeber. Compliant actuator designs. *IEEE Robotics Automation Magazine*, 16(3):81–94, 2009. ISSN 10709932. doi: 10.1109/MRA.2009.933629. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5233419.

[95] B Vanderborght, N G Tsagarakis, Claudio Semini, Ronald Van Ham, and Darwin G Caldwell. MACCEPA 2.0: Adjustable compliant actuator with stiffening characteristic for energy efficient hopping. *ICRA09*, pages 544–549, 2009. doi: 10.1109/ROBOT.2009.5152204. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5152204.

[96] Bram Vanderborght. *Dynamic stabilisation of the biped Lucy powered by actuators with controllable stiffness*. PhD thesis, Vrije Universiteit Brussel, 2007. URL http://books.google.com/books?hl=en&lr=&id=Q0yKIla_RNQC&oi=fnd&pg=PP2&dq=Dynamic+stabilisation+of+the+biped+Lucy+powered+by+actuators+with+controllable+stiffness&ots=5DZiNXrwSE&sig=ioBUw05mCPDhtPqsanHHhEAT9t4.

[97] Bram Vanderborght, Ronald Van Ham, Björn Verrelst, Michaël Van Damme, and Dirk Lefeber. Overview of the Lucy Project: Dynamic Stabilization of a Biped Powered by Pneumatic Artificial Muscles. *Advanced Robotics*, 22(10):1027–1051, June 2008. ISSN 01691864. doi: 10.1163/156855308X324749. URL http://openurl.ingenta.com/content/xref?genre=article&issn=0169-1864&volume=22&issue=10&spage=1027.

[98] Bram Vanderborght, Alin Albu-Schäffer, Antonio Bicchi, Etienne Burdet, Darwin G Caldwell, Raffaella Carloni, M Catalano, Oliver Eiberger, Werner Friedl, Ganesh Ganesh, et al. Variable impedance actuators: A review. *Robotics and Autonomous Systems*, 61(12):1601–1614, 2013.

[99] Bj Verrelst, Ronald Van Ham, Bram Vanderborght, Frank Daerden, Dirk Lefeber, and Jimmy Vermeulen. The Pneumatic Biped "Lucy" Actuated with Pleated Pneumatic Artificial Muscles. *Autonomous Robots*, 21(2):45–213, March 2005. doi: 10.1007/s10514-005-0726-x.

[100] Ludo C Visser, Raffaella Carloni, and Stefano Stramigioli. Energy-efficient variable stiffness actuators. *Robotics, IEEE Transactions on*, 27(5):865–875, 2011.

[101] Miomir Vukobratović and Branislav Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 1(01):157–173, 2004.

[102] Dan Wagner, Joseph A Birt, Michael Snyder, and James P Duncanson. *Human Factors Design Guide*, chapter 14. FAA Technical Center, 1996.

[103] Michael W Whittle. *Gait Analysis: An Introduction, Fourth Edition*. Butterworth, Heinemann, Elsevier, 2007.

[104] S Wolf and G Hirzinger. A new variable stiffness design: Matching requirements of the next robot generation. *2008 IEEE International Conference on Robotics and Automation*, pages 1741–1746, 2008. doi: 10.1109/ROBOT.2008.4543452. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4543452.

[105] J. Yamaguchi, D. Nishino, and a. Takanishi. Realization of dynamic biped walking varying joint stiffness using antagonistic driven joints. *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, (May):2022–2029, 1998. doi: 10.1109/ROBOT.1998.680612. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=680612.

[106] Jin'ichi Yamaguchi, Eiji Soga, Sadatoshi Inoue, and Atsuo Takanishi. Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 368–374. IEEE, 1999.

## COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

http://code.google.com/p/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/