Mechatronic Simulation & Exploration of a Mechanical Context relevant to Quadrupedal Neuromorphic Walking employing Nervous Networks for Control

A thesis presented in partial fulfillment of the

requirements for the degree of

Master of Engineering

Mechatronics

at

Massey University,

Albany, New Zealand.

Matthew Read

2008

Abstract

Neuromorphic engineering is the study and emulation of neural sensory and control structures found in the natural world. Currently a significant research focus in this field, and indeed, in engineering at large, is the research of robotic walking platforms – an ideal application for artificial neural controllers.

To design such neuromorphic controllers, significant knowledge is needed of the robotic context to which they will be applied. The focus of this research is to explore the relationship between the mechanical design of a robot, and its resultant walking proficiency.

A neuromorphic controller utilizing Nervous networks was constructed, and embedded into a typical & useful mechatronic context. This consists of a simple walking platform, of a type commonly used in Nervous network research. This robot was used to provide intuition and a reference point for development of a simulation for empirical testing.

A physical simulation of the mechanical context was developed, allowing for the exploration of its behaviour, particularly with regard to the type of walking "caused" by the integration of an appropriate Nervous network controller.

To evaluate the behavioural fitness of this context in various configurations, empirical simulations were run using the developed simulation, and heuristic results derived to develop optimized parameters for causing walking behaviours in the studied context. Further simulations were then run to evaluation the efficacy of these developed heuristics.

From these simulations & explorations, the presence of an identifiable "critical point phenomenon" in the interaction between the robot's legs was demonstrated. This critical point was then used for parameter extraction; further simulation demonstrated that parameters extracted from this critical point provided near-optimal walking behaviour from the robot in a variety of leg topologies.

These results provide significant knowledge and intuition for designers of quadrupedal walking platforms, particularly those driven from Nervous network derived neuromorphic controllers. Implementation of these results in such a robotic platform will provide useful new "real world" data, allowing the developed models & heuristics to be further refined.

Acknowledgements

Though completed now this thesis is,
But by long labour it would not have been;
So many hours of solder, sweat, and tears,
Were given, and never shall be seen again.

But not just by my merit, not just by my toil,
Did this research happen or this thesis be wrote;
A vast host of helpers, motivating, encouraging,
Were oft all that gave me inspiration and hope.

At home, ever steadfast, with wisdom and patience
Ideas, support, and good food gave my parents to me;
At my first home away, my flatmates, my brothers,
Buoyed up my spirits in such times I had need.

My second home away, the house of the Creator,
With prayer and counsel, my friends and pastors e'er were there;
My third home, the Institute, the ceaseless office-moving,
Yet never-the-less from my supervisor, support always I found here.

Johan, Peter and Olaf, Lim, Baden, Gordon and Jonty,
Colleagues many and varied there be;
Sound advice, handy components, and friends to commiserate,
Though I'm sure some will be glad to see the back of me.

To Amber, my dear friend, I owe ever so much,
The ear to listen, and the heart to care;
And last but not least, to the God of the Universe
I owe my life, my sanity, my freedom, my gifts and my mind.

Without you all, this thesis would never have existed.

Thank you.

Table of Contents

List of Figures

List of Equations

List of Tables

Abbreviations

Nv-neuron	Nervous Neuron
Nu-neuron	Neural Neuron
Nv-net	Nervous Network
CPG	Central Pattern Generator
ITE	Institute of Technology & Engineering, Massey University
aVLSI	Analogue Very Large Scale Integration
BEAM	Biology, Electronics, Aesthetics, Mechanics
LANL	Los Alamos National Laboratory
IEEE	Institute of Electronic & Electrical Engineers
V-leg	Vertical Stepping leg, i.e. the leg with a rotational axis along the Z-axis (along the "spine" of the robot)
H-leg	Horizontal Stepping leg, i.e. the leg with a rotational axis along the Y-axis.

1 General Introduction

1.1 Aim

This Masters thesis will explore and investigate, in simulation, the influence of various factors in the design of a neuromorphic walker's mechanical context on the robot's walking behaviour, through a fusion of mechatronic and neuromorphic engineering techniques.

1.2 Introduction

Robotic walking has been a much pursued field of study for many years now, for a wide variety of research aims, and from the perspective of a broad spectrum of disciplines, both within engineering, and as part of the wider study of science (Paulsen, 2004). This project will focus on a particular subset of the possibilities for robotic walking offered by the fields of mechatronics and neuromorphics, applied through the perspective of BEAM design philosophy.

The particular robot design to be used, to be detailed in section 2: Robotic Platform, is a simple quadrupedal walker driven by two motors, with the control provided by a nervous network, or Nv-net. Nervous networks are a type of artificial neuronal system that has not yet been explored widely. Despite their successful implementation across dozens of robotic designs, academic investigation into their properties and behaviour has been scarce, due in part to the underdeveloped theory currently underlying their operation (Tilden, 1997).

Many of the design philosophies applicable to the design and research of systems using this type of neuronal network (to be outlined in section 1.4: Design Approaches) espouse the usage of tightly integrated multi-disciplinary techniques. And indeed, due to the "bounded chaos" nature of Nv-nets (Tilden, 1997), it is very difficult to abstractly

understand how a robot designed with such a controller will react and behave in the "fractal environment" of the real world (Rietman et al., 2003).

As such, the author proposes that much meaningful research into the behaviour of Nv-nets will require them to be embedded into "environmental interfaces" – robots or other similar systems that provide the network with an environment to interact with, be stimulated by, and respond/react to. One of the simplest such situations developed is a small, two-motor, quadrupedal walker. This design has been used extensively by both researchers and amateur roboticists, but also has a poorly developed theoretical underpinning.

This can significantly complicate the study of the Nv-net's behaviour in this context, and thus it is necessary to facilitate better causal isolation between the influence of the controller upon the robot's behaviour, and the influence of the mechanical geometry on the robot's behaviour. This study intends to explore the influence of the mechanical platform of such robots, from a geometrical perspective, in the hopes of providing a better defined platform and context within which more dedicated Nv-net research can take place.

1.3 Robotic Walking

In recent years, a major focus of study within the field of mobile robotics has been the study and development of perambulatory robots – robots that employ legs as their primary means of locomotion (Wahde & Petterson, 2002). Many of these robots draw at least minimal inspiration from the natural world, and are developed for a wide variety of applications and motivations.

Current research into walking and legged robots is driven by a wide variety of motivations, but some generalizations can be drawn. One very common such "high level" goal is to produce robots that are able to operate within the same environment and "task domain" as humans. An excellent example of this is Honda's ASIMO robot, which

appears very human-like, and has demonstrated the ability to climb up and down stairs, as well as natural-looking walking behaviour (Hirai et al., 1998).

Such an approach aims for several advantages, in particular the ability of the robot to use equipment and facilities designed originally for human use. Although this also extends significantly to the manipulators etc... or the robot, the legged locomotion system is also a very important consideration when developing robots towards this aim. Any robot expected to work in such environments will need to be capable of handling stairs, avoidance of moving obstacles (i.e. humans) in narrow passages, and ideally be able to deal with sitting and standing at workstations designed for human use, all while maintaining its balance.

Liu et. al (2007) note that, in addition to this, legged robots designed to operate in outdoor environments hold significant advantages over equivalent wheeled/tracked designs. Particularly, a walking robot requires only to make contact with discrete points of the environment in order to move, whereas a wheeled robot requires a safe, continuous path to achieve motion between two points. In rough terrain, this can significantly limit the number of potential paths for a wheeled robot, or may present no traversable path at all; a legged robot would however have the ability to evaluate a path as a discrete series of safe "stepping stones" that were within the capabilities of its walking mechanism and controller.

Further to this, another major rationale for the research of humanoid walking robots, is to learn and understand more about how the human body deals with walking. Most humans learn to walk very early in their life, and with the exception of severe neural or muscular damage, rarely struggle with or lose this ability until well into old age (Guralnik, et. Al., 2001). By conducting research into what allows a robot to walk, we thus gain further insight into the internals and control techniques that allow humans to walk so intuitively; this in turn will result in new techniques and viewpoints for the development of robotic walking systems (Paluska, 2000).

At a somewhat simpler level, and perhaps more useful in the short term, is the similar pursuit of walking systems modeled of insects, arachnids, and simple quadrupeds such as lizards or salamanders (Long Jr., J. H., 2007). As the neuromuscular complexity of these organisms is much less than that of a human, these studies offer the opportunity for more immediately accessible insight, and thus provide clues and inspiration for the pursuit of understanding regarding biological walking movement.

1.4 Design Approaches

As robotics is both a diverse and rapidly-evolving field of engineering, a wide variety of approaches and philosophies have been applied to robotic design.

1.4.1 Classical

Classical robotics is used here as a rather broad term, but in essence refers to the prevailing design philosophy for a significant proportion of the "Age of Robotics" – the process of mathematically modeling the robotic mechanism in question, and using this model to produce analytical control expressions that can be used to determine the joint angles required of a mechanism to reach a certain position or stance (Goldenberg, 1985).

This approach has been used successfully in many areas of robotics, and particularly the field of industrial robotics, where robotic manipulators are required to perform a consistent series of algorithmic steps many times over.
In this situation, the ability to apply rapidly calculable mathematical equations to compute the robot actuations required to fulfill a particular algorithmic step provides a simple and reliable system. The most common analytical method applied to such tasks is that of Inverse Kinematics. Kinematics is the study of motion, and by applying its methods to a robot, it is trivial to develop an equation that can express the position and orientation of an end effector as a function of its actuator angles. The Inverse Kinematic

model of a robot, however, attempts to express the actuator angles of a robot as a function of the position and orientation of the end-effector (Goldenberg et. al., 1985).

Additionally, many robots operate in situations where, despite the repeatability of the overall task, small variances may occur between iterations of a task, such as picking fruit from a conveyor belt (Corke, 1994). This necessitates additional computational capability in the robot to calculate, with the aid of relevant input data provided from sensors (such as video cameras, for the fruit picking example), the modifications to the algorithm's defined motion paths and operations to allow the robot to adapt.

The generation of dynamic paths, however, rapidly becomes more complex as DOFs are introduced to the system, or if constraints or motion paths are applied to the individual links composing the robotic system, rather than just the end effector (Shiller & Dubowsky, 1989). The primary cause for this increase in computational burden is because the system can no longer choose any available path to move the end-effector in a particular way – it must now evaluate a motion path that is subject to strict constraints, and may even require backtracking of some joints. To express this another way – although fewer choices are now available to the system, the computational burden is increased simply because a choice must now be made.

Veering away somewhat from industrial robotics, the above principle particularly comes into effect in the development of robots to interact or work in environments where humans are significantly present. Robots must be able to accurately sense and rapidly adapt to human presence within their operating domain, as many robots are capable of harming a human being within their normal operating cycle (Ikuta et. al., 2003).

While these problems require serious design consideration in fixed robots, the difficulties they can cause increase by an order of magnitude when factored into the development of mobile walking robots. As a walking robot moves, joints must not only be actuated such as to effect the robot's motion, they must also be managed to keep the robot balanced (Pratt, 2000).

To this end, extensive onboard processing must be done to evaluate the robot's position and motion relative to its environment, and to thus produce a series of real-time movements that will allow the robot to remain upright in the environment, while achieving the movement goals required of it. McGeer (1990) refers to the differentiation between static and dynamic walking – static walking operates off a cycle of step-balance-plan, whereas dynamic walking involves continuous movement without pauses between steps for planning and/or balancing. Dynamic walking is particularly employed by robots that attempt to employ the natural physical dynamics of their mechanism to assist them in walking, as pausing with each step would disrupt the flow of their motions.

While some of the most capable walking robots in the world, such as Honda's ASIMO (Advanced Step in Innovative Mobility) robot, are developed and controlled using this model, significant processing power and development time is required to support perambulatory motion in this context (Chestnutt et. al., 2005).

1.4.2 Mechatronic

Mechatronics as an engineering discipline primarily evolved out of Electromechanical engineering during the 1970s, however it exhibits significant philosophical differences (Comerford, 1994). Mechatronics places a higher emphasis on integration of technologies, particularly between the software, mechanical and electronic engineering, and thus attempts to create a field of engineering that is intrinsically multi-disciplinary; a diagram expressing this disciplinary fusion is shown in Figure 1 below.

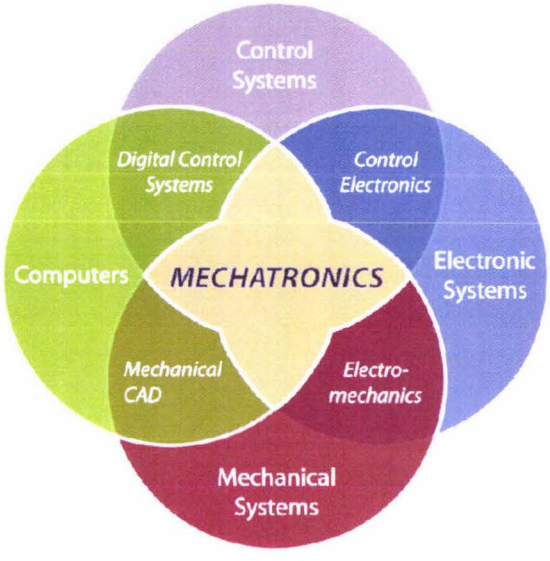

Figure 1: Mechatronics Overview

This emphasis on systems integration by mechatronics lends itself well to the design of all types of robots, including mobile "intelligent" robotic agents. It is thus difficult to define particular examples of Mechatronic robotics project, as almost all robotic research could be classified, to some extent or another, as an exercise in mechatronics (Ferretti, et. al., 2004).

1.4.3 Neuromorphic

Neuromorphic research is primarily categorized by the emulation and imitation of neural and sensory systems from natural organisms. A major theme in recent neuromorphic research has been the development of "neural" sensors that attempt to perceive & process information and stimuli in a similar fashion to equivalent biological sensors (i.e. sensory organs) (Bains, 2004).

A typical example of such research is the work of Netter (2004) in designing and implementing a neuromorphic equivalent of a fly's eye. The developed sensor's

perception is highly pixilated (much like the image received by a fly's compound eye), however it is able to perform high level feature identification in real time, without the aid of a DSP or microprocessor. Figure 2 below shows an example of the output generated by this sensor.

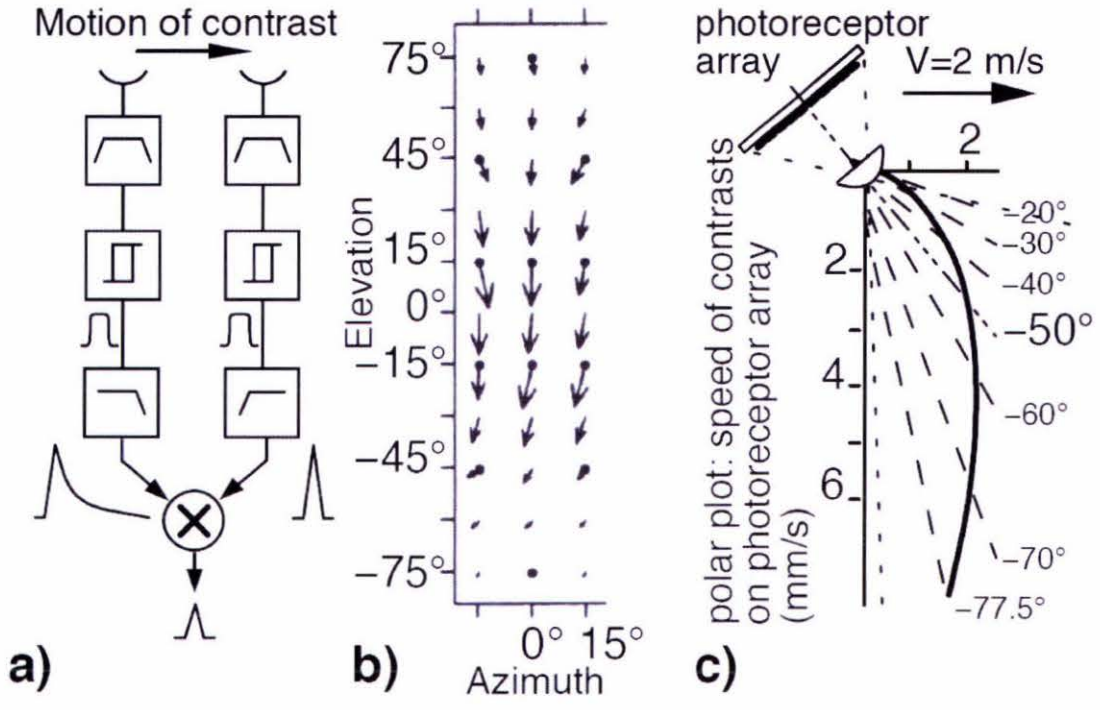

Figure 2: Output from a Neuromorphic aVLSI Implementation of a Fly's eye (Netter, 2004)

In "true" neuromorphic fashion, this sensor not only emulates the stimulus collection method (a low-resolution array of grayscale sensors), but also the stimulus processing technique and the information transmission protocol from the target organism.

While much research has indeed been focused on the study & development of this type of "artificial sensory organs", two other significant avenues of research are observed by the author in current literature. As opposed to the experimental & practical implementation research being pursued towards neural sensor technologies, a significant body of research has been performed to develop the enabling technology of aVLSI (Analogue Very Large Scale Integration) as well; indeed, the field of neuromorphics developed out of research in this area (Mead, 1990)

Although Tilden has demonstrated the potential to mass-produce neuromorphic machines using discrete components (Hyrnkiw & Tilden, 2002), implementing neuromorphic systems in aVLSI has significant advantages, particularly in terms of miniaturisiation (allowing for smaller systems which require less power to operate) and mass production (making such systems viable for everyday application). Additionally, the ability to create abstracted, modular, aVLSI "organs" that can be used as discrete processing components is also popular among neuromorphic researchers (Haefliger, 2005, Vogelstein et. al., 2004).

1.4.4 Biomorphic

Strongly related to neuromorphic engineering, biomorphic engineering involves considering an entire organism, rather than focusing on the neural and sensory elements thereof, as shown in Figure 3 (Smith, 2001). The name is a reference to "biological morphology" – morphology being the study of shape and form, and biology referring to natural organisms – thus the engineering study of biological organisms in their entirety.

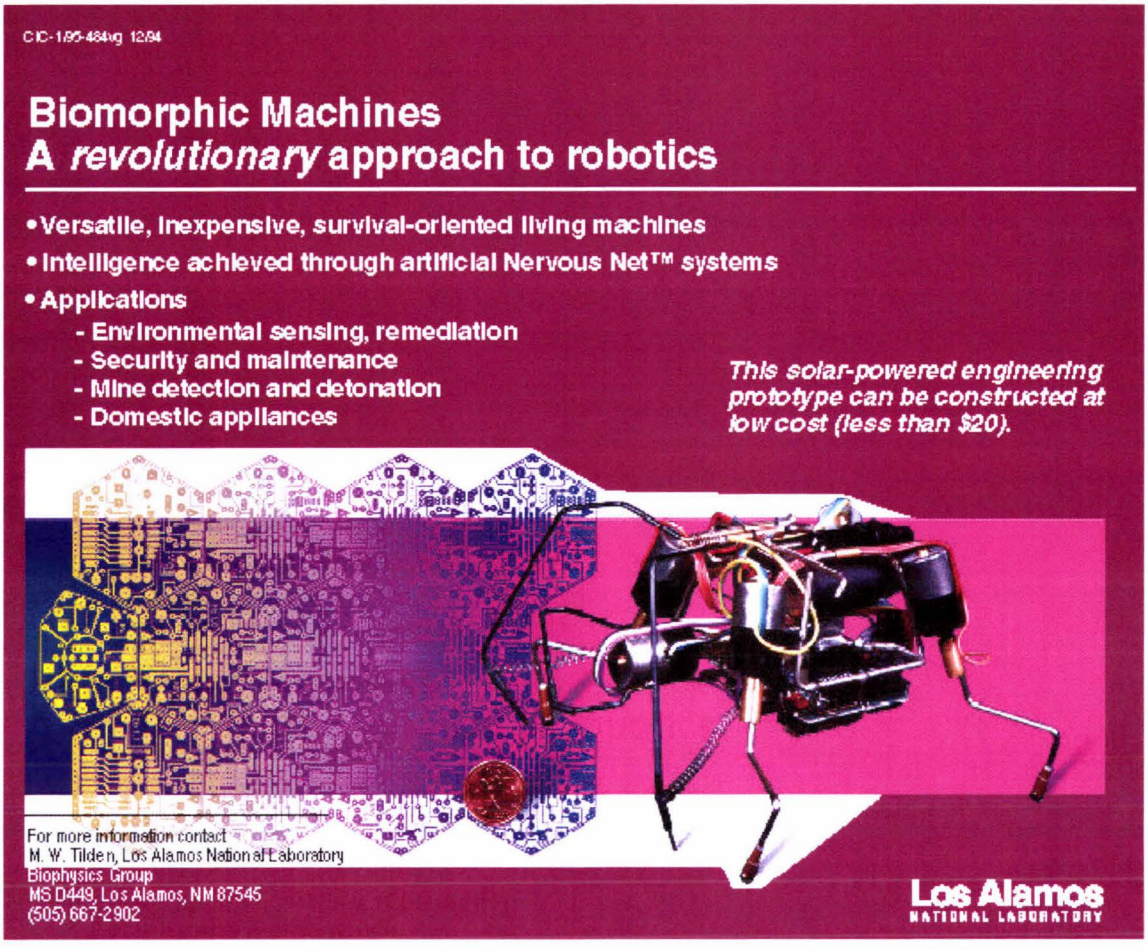

Figure 3: An Overview of Biomorphic Robotics (Smith, 2001)

There exists considerable overlap in neuromorphic & biomorphic research, making it difficult sometimes to distinguish between the two fields. As such, the term "neuromorphic" is used relatively loosely throughout this thesis, referring not only to biomorphism & neuromorphism as engineering disciplines, but also to the concept of biomimetism (engineering imitating nature) as a whole.

This confusion exists partly due to the fact that any biomorphic design will inherently incorporate neuromorphic elements, as any biological organism inherently contains a neural component. An example of a typical biomorphic research project is Rodriguez' (et. al., 2006, 2006) work with deriving biomorphic designs from *Coleoptera Scarabaidae*, a type of scarab beetle.

Rodriguez' team constructed a robot closely imitating the morphology of the scarab beetle, and developed a controller to imitate its walking gaits and behaviour. The were able to use this model to then explore the changes that a scarab's "walking controller" had to implement in order to effect proficient locomotion across a wide variety of terrain types, as well as providing comparative studies of the relative performance of traditional control systems and locomotion platforms.

In simulating such organisms, it is thus possible to gain considerable and valuable insight into the engineering design of the natural world: our ability to imitate these natural sensory & mechanical systems is growing rapidly due to concerted biomimetic research, allowing us to engineer "better" systems from lessons learned from nature. There is still much to learn so we can more closely emulate nature's elegantly designed systems, thus necessitating continued research into this field.

1.4.5 BEAM

BEAM robotics forms a significant sub-discipline of Biomorphic engineering, however it is significant in that a large amount of activity in this area is carried out in non-academic circles. In particular, an extremely active community of amateur roboticists has made significant contributions to the practical implementation and design of a veritable plethora of robots. From a research perspective, however, this can create difficulties due to the low level of academically-recognised published theoretical advancement carried out by such experimenters. Additionally, significant advances have certainly been made by notable "consumer robots" such as the Robosapiens, BIO-Bug, and related robots, however due to the commercial nature of their designs, few results or theoretical research have been made publicly available.

As a design philosophy, however, BEAM is well developed, and provides an interesting "worldview" combining biomorphic, biomimetic and mechatronic approaches. BEAM robotics requires an approach that considers all aspects of a robotic system as influencing

all other aspects – for example, adjusting the shape of the legs on a walker is as much "programming" the behaviour of the resultant robot as altering its controller (Geng, 2007). This mirrors both the mechatronic approach of tight integration with the biomorphic consideration of an organism as an entire system.

Additionally, BEAM also places considerable emphasis on aesthetics as a design tool – as many BEAM designs are biomimetic in nature, this draws off the observation that many natural systems not only work well, but also look good – function follow form. However the inverse is also true, and many BEAM-designed systems are relatively minimalist, allowing form to follow function, with little extraneous "eye candy" or non-functional cosmetic components (Tilden, 1997). As an example of this, the Robosapiens 2 consumer robot has what appear to be purely cosmetic kneecaps on its legs; on closer observation of the walking robot however, it becomes apparent that these kneecaps form an integral part of its walking gait.

Many of these principles are expressed by BEAM's name, a very flexible acronym from which several apt descriptions of this design philosophy have been constructed, including:

- **B**iology, **E**lectronics, **A**esthetics, **M**echanics
- Biotechnology Ethology Analogy Morphology
- Building Evolution Anarchy Modularity

The first of these is the most commonly used acronym to describe the field due to its broad scope, as many of the other "backronyms" describe only particular facets of the philosophy in detail.

Another distinguishing feature typical of BEAM designs is the lack of a microcontroller. The primary reason for this is, in following the philosophy of elegant imitation of nature, it is more suitable to employ neural controllers (Bain, 2004). While in amateur circles it is common to construct the desired artificial neuronal network from discrete components,

a significant level of research into aVLSI has been undertaken by the neuromorphic community, and is relevant for academic research into complex BEAM "organisms".

An important aspect of BEAM design philosophy is the distinction between designing robots as creatures instead of machines – organisms instead of automata. As described by Tilden (2004), robots have been developed that are survival-, as opposed to task-, oriented – and as such are designed to a) require some resource to "survive", and b) employ some strategy to acquire this resource.

A practical example of the possibilities of this approach to robotic design is the BIObug – a commercial toy developed by Tilden. The BIObug was produced in several different "species" (each with an identifying colour), each with different behavioural characteristics. Two of these "species" were coloured blue and red – blue bugs are "right handed" and can only approach "food sources" (i.e. an IR transmitter) by orbiting in a clockwise direction; red bugs lacked this trait, and could directly approach a food source.

The resulting behaviours of this combination bear eerie similarity to the behaviour of natural creatures. Blue robots tended to cluster around the IR "food" sources, jostling for position, while their red counterparts tended to make high-speed passes towards the food source. Amazingly, the blue bugs would then tend to cluster around such an intruder and force it away from the light source, before returning to the feeding circle. As Tilden (2006) describes it as, "... just one of a billion experiments I've done where the behavior is obvious to a child, but alas, mathematically intractable despite the minimality of the control and sensor networks."

1.5 Neuromorphic Walking

As outlined earlier, neuromorphic engineering focuses on technologically emulating neuronal & sensory systems found in natural organisms. When this approach is applied to the study of legged locomotion, the author proposes that this requires special consideration. Particularly, this approach requires that neural systems responsible for

walking in an organisms must first be studied in a "mechanical context" similar to that of the organism concerned. Only then can meaningful exploration or conclusions be pursued with regard to constructing knowledge relevant to walking as a whole.

This view is supported by Lewis et. al. (2003) and others, and is a corollary of physical computation. As summarised by Lewis, physical computation refers to the common natural theme of simplifying control requirements by employing mechanical structures which effectively provide computational assistance/complexity to simpler neural behaviours. The physical dynamics of the mechanical context in which a controller is embedded thus has the potential to exhibit significant influence on the overall behaviour of the robot, justifying the examination of the mechanical context in this thesis in order to provide a platform for future work with the intended Nv-net controller.

In natural organisms, two predominant walking paradigms exist: reactive walking, and CPG-driven walking (Geng, 2007) Most creatures employ one or the other of these methods, or a combination of the two.

As an example of predominantly reactive walking, Cruse et. al. (1998) point out that a significant body of research has suggested that stick insects (*Carausius morosus*) have no central rhythmic generator to control their walking. Rather, purely reactive models have been created which accurately model the walking behaviours of such insects in slow walking gaits. Cruse models the stick insect walking controller as consisting of three distinct networks: one to perform the forward return "swing" leg movement, one to perform the driving "stance" leg movement, and a "selection" network to switch between the two as the leg reaches the extremes of its motion. These networks exist independently for each leg, and although some synchronisation between legs occurs during stick insect walking, each leg is capable of perform steps without any central driving factor.

Contrast this form of perambulatory control with that discussed by Duysens et. al. (2002). It is pointed out that many "higher" organisms rely not on purely reactive systems, but

rather have some form of central rhythmic controller which "manages" the walking behaviour of the organism - a Central Pattern Generator, or CPG. This may range from relatively simple neuronal networks in organisms such as the Salamander (Aoi & Tsuchiya, 2006), to more complex mammals such as cats (Duysens, 2000). Regardless of the complexity or arrangement of these networks, they all serve the same purpose - to generate and regulate patterns of motor impulses to regulate walking.

However, rather than exhibiting a pure approach, many natural organisms employ some combination of these two control techniques. For example, Geng (2007) summarises results confirming the existence of both CPG and reaction-based contributory systems to walking in cats. In this case, CPGs are located within the spinal column, and provide "high level" motor control to the walking, whereas reactive systems manage some lower level functionality, such as maintaining a walking gait despite a limb encountering an obstacle.

1.6 Nervous Network

1.6.1 Introduction

Nv-networks were developed as a technology by Mark Tilden & Brosl Haslacher during the 1990s, at Los Alamos National Laboratory, in the USA. Stemming from research into non-linear oscillators (Tilden, 1997), they quickly realized the potential of this technology for novel forms of controllers and sensory processing. The BEAM design approach was developed by Mark Tilden during this time, and remains a predominant school of thought in the development of Nv-net based robots.

1.6.2 Nervous Neurons

An Nv-neuron consists of a high-pass RC filter attached to the input of Schmitt trigger inverter logic gate, shown in Figure 4 below:

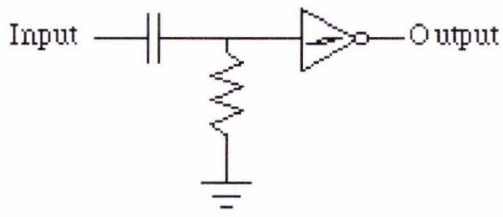

Figure 4: Nervous neuron (Solarbotics.net)

1.6.2.1 High-Pass RC Filter

A high-pass RC filter is a simple passive network that "blocks" low frequency signals by means of a capacitor. It can be analysed as a simple voltage divider, providing that the complex impedance of each component is considered, not just the real resistance (i.e. impedance = resistance + j x reactance). This calculation is shown below as Equation 1:

$$X_C = \frac{1}{\omega \cdot C}$$

$$X_R = R$$

$$\because V_o = \frac{V_i \cdot X_o}{X_i + X_o} \text{ (Voltage divider equation)}$$

$$\Rightarrow V_o = \frac{V_i \cdot X_R}{X_C + X_R}$$

$$\Rightarrow V_o = \frac{V_i \cdot R}{R + \frac{1}{\omega \cdot C}}$$

$$\therefore V_o = \frac{V_i \cdot \omega \cdot R \cdot C}{1 + \omega \cdot R \cdot C}$$

Equation 1: Output voltage of an RC High Pass Filter

1.6.2.2 Schmitt Trigger Inverter

For the Nv-neuron implementation used in this project, the Schmitt trigger inverter was replaced by a Schmitt triggered NAND gate with its inputs connected together – this is computationally & logically equivalent to the inverter component. The characteristic curves as defined by the chip's datasheet (available at: http://library.solarbotics.net/datasheets/MM74HC14.pdf) are shown below in Figure 5:

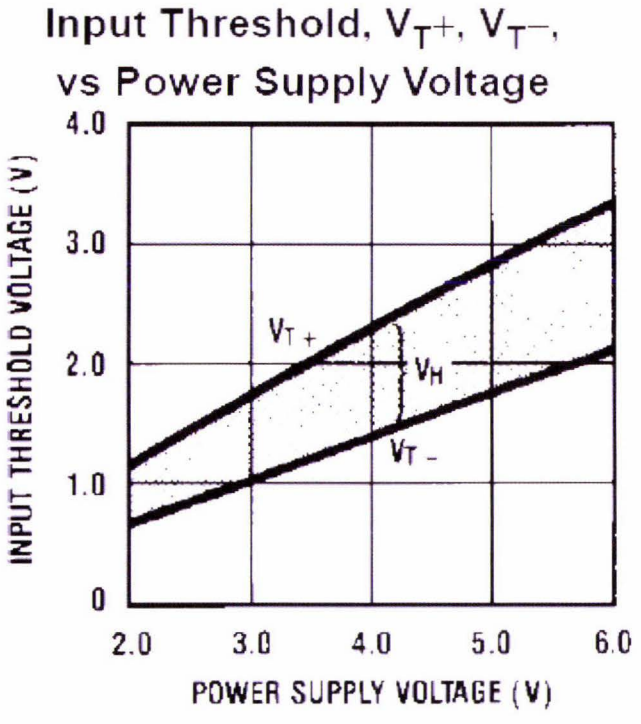

Figure 5: Schmitt-Trigger Inverter Threshold Voltages

A schmitt-triggered circuit implies that, in signal processing terms, it has some degree of "memory", as it exhibits *hysterisis* – the switching thresholds are different depending on whether the input is a rising or falling edge. As can be seen in Figure 5 above, V_{T-} is the threshold voltage for a falling input signal, and V_{T+} is the threshold for a rising input. This behaviour significantly increases the noise tolerance of a circuit, as well as allowing the component to "fuzzily" respond to real input as well.

1.6.2.3 Combined Behaviour

In order to express the behaviour of the entire neuron, it is necessary to combine the individual behaviours of the individual components present in the neuron. The behaviour, assuming a compatible input from an identical neuron "upstream", is shown in the following diagrams:

$$V_{in}(t) = V_n(-u(t-t_0)+u(t-t_1))$$

$$u(t) = \begin{cases} 0, t < 0 \\ 1, t \geq 0 \end{cases}$$

$$V_a(t) = \begin{cases} 0 & ,t < t_0 \\ -V_n\left(\dfrac{-e^{-(t-t_0)}}{RC}\right) & ,t_1 < t < t_0 \\ \dfrac{\left(v_n + V_a\left(t_1^-\right)\right)\cdot e^{-(t-t_1)}}{RC} & ,t > t_1 \end{cases}$$

$$V_{out} = \begin{cases} V_{cc} & ,t < t_1 \\ 0 & ,t_1 < t < t_2 \equiv V_{cc}(-u(t-t_1)+u(t-t_2)) \\ V_{cc} & ,t > t_2 \end{cases}$$

$$t_2 - t_1 = -RC \cdot \ln\left(\frac{V_{tl}}{V_{cc}\left(1-e^{\frac{-(t_1-t_0)}{RC}}\right)}\right)$$

Equation 2: Voltage Response of a Nervous Neuron

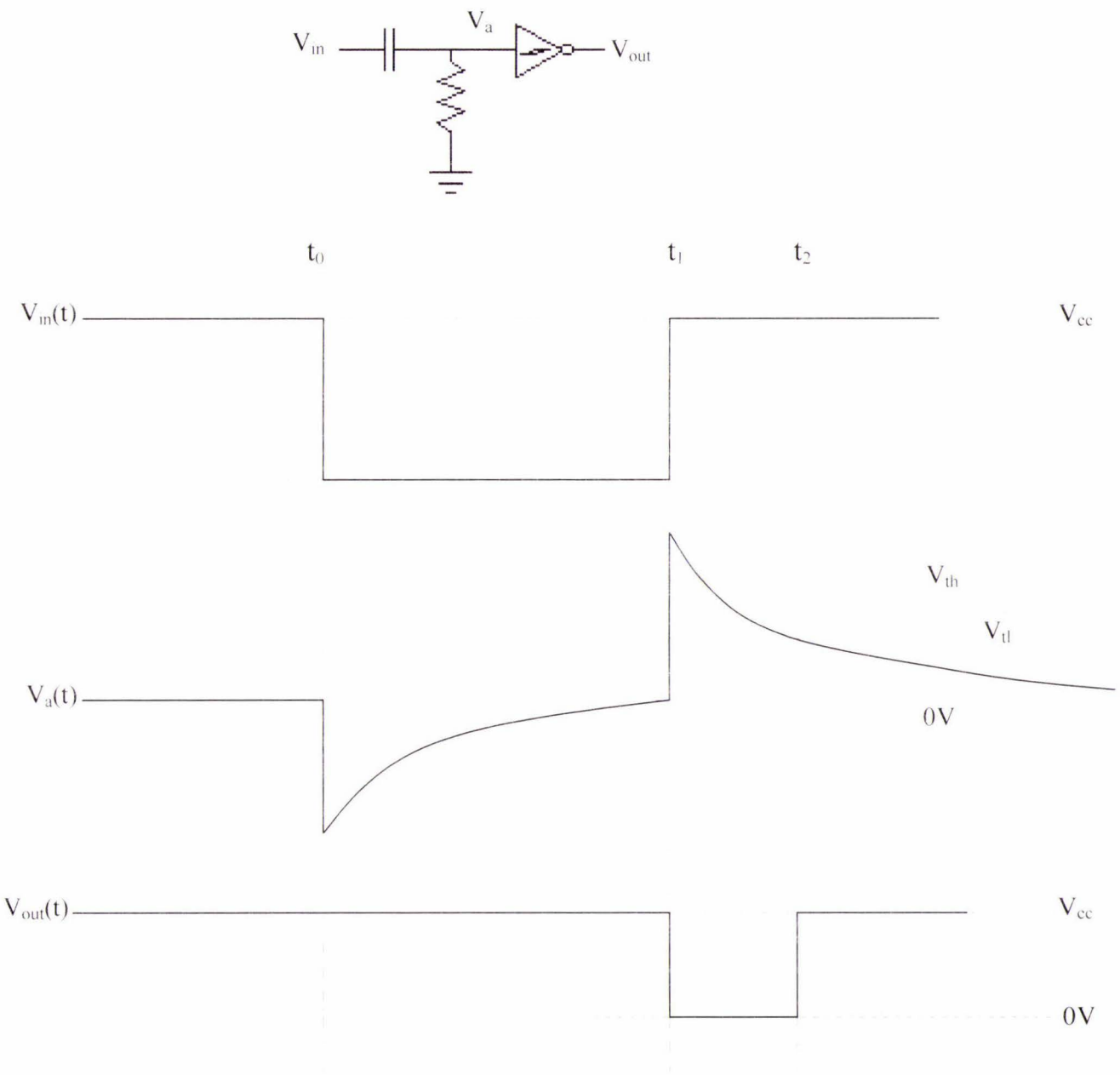

Figure 6: Voltage Response of a Nervous Neuron

The following implications are evident from the preceding equations and figures:

- As expressed by these equations, the neuron will only become "active" when there is a change of states at the neuron's input - more precisely, a falling edge.

- No matter what type of pulse is input to the neuron, the output will only be one of two things: 0, or a "square" pulse with a duration related to the amplitude and "frequency" of the input pulse.

- Likewise, no matter what kind of pulse train, sinusoid, or other periodic signal is input to the neuron, the output will be only one of two possibilities - a pulse train of frequency and duty cycle related to the frequency and amplitude of the input, or 0. It is also likely possible to "saturate" an Nv-neuron with a suitable high frequency input, however this case is not relevant to this thesis, and will not be considered.

Of particular note is the nature of the RC circuit – unlike traditional Nu-neurons (i.e. "neural neurons", the type commonly used in classical neural networks), which predominantly employ integrative & summative techniques to process their inputs, Nv-neurons employ a differentiator in their input path. The result of this is that Nv-neurons are effectively edge triggered, as the neuron's response will be based on the first derivative of the input signal.

One result of this is that Nv-neurons are very tolerant of both static biases and low-level noise, while still remaining sensitive to signals of the "correct" format.

1.6.3 Nervous Networks

1.6.3.1 Bicore

As its name implies, the bicore consists of two Nv-neurons connected in a loop topology. This circuit represents the simplest possible Nv-net topology; however its behavioural patterns can be exploited to control robots at many levels of complexity, particularly if multiple bicores are coupled together. The schematic of a typical bicore network is shown in Figure 7 below:

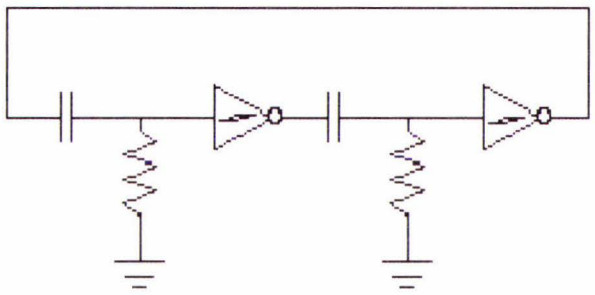

Figure 7: A typical Bicore Nervous Network Circuit (Solarbotics.net)

If each neuron is treated as having a binary state (1 = 'activated' = output low, 0 = 'off' = output high), the bicore network can be shown to only produce the following "binary" network states:

State	Neuron 1	Neuron 2
0	0	0
1	0	1
2	1	0
3	1	1

Table 1: Bicore Network States

The "states" 1 & 2 are equivalent, as they each represent only one "active" neuron on the network. This circuit thus has only one stable mode of oscillation (with respect to neuron excitation) - in order for the bicore circuit to be useful for more complex applications, it must be used in a coupled network topology.

1.6.3.2 Quadcore

The quadcore is an Nv-network that follows the same basic loop topology as the bicore; the quadcore is however made up of four neurons. The schematic for the quadcore circuit is shown in Figure 8 below.

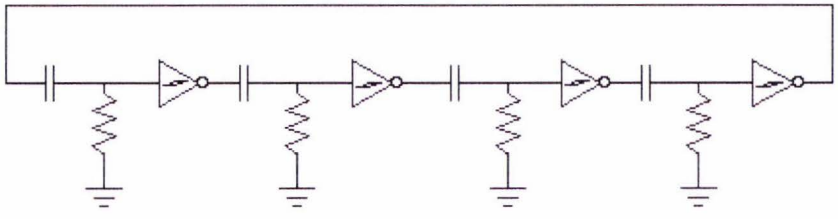

Figure 8: A typical Quadcore Nervous Network Circuit (Solarbotics.net)

The equivalent "binary" analysis of a quadcore network yields Table 2:

State	Neuron 1	Neuron 2	Neuron 3	Neuron 4
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Table 2: Quadcore Network States

Thus there are only six distinct patterns that can be present in the network's operation, only two of which represent a stable network state: 5, and 1. This being the case, the full state diagram for the quadcore network is in the Figure 9 following:

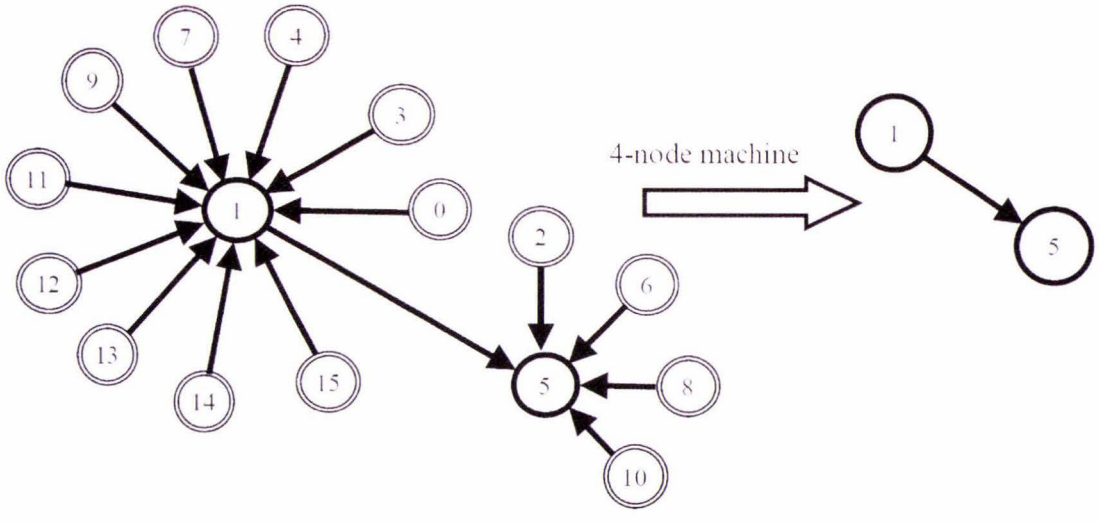

Figure 9: Input/output behaviour (attractor diagram) for a Quadcore Nervous Network (Rietman et. al., 2003)

While a simple robot (such as the context studied by this thesis) may use only one of the "behaviours" represented by a particular pattern/rhythm present on the network, more complex controllers may utilize the additional possible patterns to provide advanced functionality & adaptive behaviours. Research by Tilden & Haslacher (1995), the founding researchers of the Nv-neuron technology, proposed chaos theory as a potential

means of understanding the operation of Nv-nets, and Rietman et. al. (2003) developed a model based on necklace functions; neither of these are complete, however, and thus research is ongoing into the theoretical underpinnings of nervous networks.

1.6.3.3 Other Topologies

Other topologies have been developed to exploit the behaviour of nervous networks, varying in size, complexity and the difficulty of their analysis. A short list of some of the principal alternate topologies not already discussed is shown below:

- Large Rings

  Topologically identical to the bicore and quadcore designs discussed earlier, these networks simply include higher numbers of neurons. For all such networks with an even number of neurons, their behaviour is similar to the smaller networks discussed earlier, however significantly more complex (Rietman et. al., 2003). As the network grows, the number of stable states able to exist on the networks grows super-linearly. Particularly, these networks rapidly become very difficult to analyse, as their phase-space grows exponentially with the number of neurons.

- Odd-numbered networks

  Little work has been done in analyzing rings of Nv-neurons containing an odd number of neurons, however a brief exploration was published by Rietman et. al. (2003). They demonstrated that, while observed network behaviour appeared to be chaotic, there may be some form of underlying order. A statistical analysis (Autocorrelation function; lag represents time-distance for correlation) undertaken to determine whether the neuron states were purely random showed a small, but statistically significant, skew in the correlation distribution. This is shown in Figure 10 below:

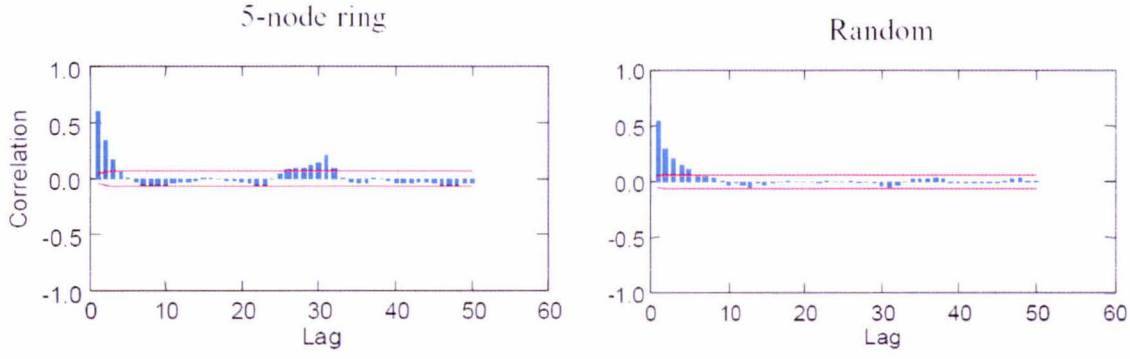

Figure 10: Autocorrelation Function plots for 5-neuron Nv-network compared to Random (Rietman et. al., 20030)

Many current models of Nv-nets do not deal with the behaviour of odd-numbered networks, and thus there is little or no research into their behaviour and application. However considerable work will be needed to understand whether useful work can be extracted from this type of topology.

- Wave Processor

Whereas Nv-nets such as the quad core use a form of one dimensional coupling between neurons (i.e. each neuron is connected to only one other neuron in each direction), the "wave processor" type topology employs a two dimensional "hexagonally beaded" grid of *quadcores*. It is proposed by Haslacher & Tilden (1995) that this topology may be ideal for modeling fluid dynamic problems.

- Coupled Rings

In order to derive more complex behaviours than can be obtained from a single Nv-net, instead of increasing the number of neurons serially in the loop of the network, a second loop can be placed in parallel with the first. By varying the level of connectivity between the rings, it is possible to vary their interaction and the resulting behaviour of the overall controller structure.

Coupled rings suffer from rapid growth of their phase-space, as the number of "layers" increases, additional to the complexity caused by the number of neurons ion each of the constituent loops (Tilden, 2006).

1.6.4 Applications

Since its development, nervous network technology has been employed in a diverse range of applications, particularly in those areas where conventional control techniques have had little success. One example of this is in the development of microsatellites to study the magnetosphere, a project undertaken at the Los Alamos National Laboratory.

Moore et. al. (1998) developed this system, designed to be deployed as a cloud of small satellites to perform measurements of the strength and direction of the earth's magnetic field, each satellite serving as a "pixel", allowing a three dimensional, real-time map of a portion of the earth's magnetic field. Nervous networks were used for the control of these satellites, as their analogue design allows them to operate within the "flow" of the surrounding environment, rather than being limited by the time-discrete nature of digital algorithmic control.

Particularly, nervous networks were used in this application due to their inherently adaptive nature, allowing for a system that was much more capable of responding "naturally" to changes in its capabilities and the environment around it than traditionally designed systems.

Another example of this is the development of biomorphic robots for the purpose of detecting and removing UXO – Unexploded Ordinance, such as mines or unexploded shells or bomblets from cluster munitions. UXO is a major logistical and humanitarian problem across many areas that have experienced wars and armed conflict in the past (Nicoud, 1997). Currently, the only UN approved method of mine removal is a slow and labour-intensive process undertaken by humans. As this process is both very tedious and

also dangerous to the individuals performing it, the development of a robotic tool to assist in this effort has been a major research goal in recent years.

Nervous network based designs have been proposed for this task due both to the adaptable nature of the controller, and because of the typical minimalistic design of such robots. The lightweight design of the robots allows them to withstand considerable buffeting from any ordinance explosion, and their adaptive characteristics allow the robot to compensate for any damage caused by such detonations. One trial of a hexapedal walker used in this application operated by "stomping" across a UXO-covered area with the intent of triggering detonations – although the robot lost several legs throughout the trial, it was able to successfully adapt to each loss, and reform a capable walking gait with each new leg topology (Tilden, 1995).

Another application area in which the minimalistic nature of nervous network-based designs is useful is in the development of commercial "toy" robots. Tilden, particularly, has been very successful in the adaptation of this technology to "toy" robots, which often have robust capabilities that are difficult to similarly implement using purely traditional robotic techniques. Examples of this include the Robosapiens – a series of capable humanoid walking robots – the FlyTech Dragonfly – a biomorphic flyer that *flaps its wings* – to name a few. The Robosapiens, the flagship model of robot produced by Tilden, is designed to be easily "hackable" by amateur roboticists, and as such serve as simple field testbeds for amateur experimentation and modification. The simplicity and elegance of the biomorphic design techniques used in the development of these robots means that the Robosapiens, a humanoid walking, is able to walk out of the box in a wide range of situations and surfaces, for around NZ$200, whereas a comparable "classical" humanoid, such as the Pirkus robot (www.robot-labs.jp), costs around NZ$3500, requires extensive programming and robotic knowledge in order to effect competent walking, and is limited in the adaptability of its walking algorithms by the capabilities of its onboard processor.

1.7 Summary

The study of robotic walking has a rich and varied history, and many design methodologies have been explored and/or created in the pursuit of this goal. A significant body of contemporary research, however, has focused on imitating nature more closely, both mechanically and with respect to controller, sensor and actuator design.

Nervous networks are a powerful and developing technology that employs non-linear dynamics to extract useful work. They are capable of producing competent biomimetic behaviours in many applications, and have demonstrated great potential for the development of adaptive, bio-mimetic, robotic "organisms", capable of dynamically adjusting to survive rapidly changing, unpredictable environments. Additionally, long-term study of Nv-neuron technology will likely result in the development of new mathematical techniques and theories in order to model and understand this type of neuronal controller.

This Masters thesis will explore & simulate the mechanical behaviour of a quadrupedal neuromorph which uses an Nv-net as its controller. The robot will only be considered at a geometrical level, with the controller and its expected behaviour providing contextual structure to the simulations. The behaviour of the Nv-net core will not be analyzed, nor will the influence of the interface electronics between the controller and its actuators/sensors. This scope will produce useful & significant results to be produced in this rapidly developing field, while providing opportunities for future work to build on these results.

2 Robotic Platform

2.1 Introduction

In order to explore Nv-net based robotics, it was necessary to construct a simple Nv-net robotic platform. It was decided to use a quadrupedal walking platform, as not only is this an area of great interest and development in contemporary mobile robotics, but it is one of the "flagship" areas of Nv-net development.

Many of the most successful and well-known applications of Nv-net technology involve walking technologies, including the Robosapiens series of robots, the demining robots developed at Los Alamos National Laboratory, and the vast range of walking robots developed by "amateur roboticists". Because of this, it was felt that exploring a walking platform would produce useful research conclusions, as well as allowing access to a significant knowledge base of user experience and circuit designs.

2.2 Controller

In the selection of a nervous network to serve as the 'brain' of the robotic platform, there were two main topologies that were considered - a dual coupled bicore, and a quadcore. Each of these networks is based on a contiguous loop of Nv-neurons, and each has been demonstrated to be capable of controlling a quadrupedal walking robot of nominal degrees of freedom.

A bicore is a topology with two Nv-neurons attached in a loop, and can be demonstrated to support only 2 stable patterns. It is the simplest possible Nv-net (because a network, by definition, must contain a minimum of two entities), but has been shown to be capable of producing useful control behaviours.

In a dual coupled bicore, two bicore circuits are constructed, and are coupled by connecting between points A & B (A = the output of neuron 1, B = the output of neuron

2) on each bicore (i.e. A1 to A2, B1 to B2). The behaviours produced by this circuit are somewhat more complicated and useful than that produced by a single bicore (Still et. al., 2006), but it remains a relatively simple circuit.

The quadcore topology is a four Nv-neuron loop that can support three stable patterns. This topology is considered one of the largest Nv-net topologies before the theory becomes prohibitively complicated (Tilden, 2006), and the smallest topology before the processing capability and capacity to produce useful behaviours starts to become overly minimal. In this respect, the quadcore has a large advantage over the bicore, in that it is an inherently more capable topology that can be designed for a wider range of applications.

For the purposes of this project, a quadcore controller was chosen, as this provided the most stable and predictable output, and had the greater capability for expansion in future work based on this project. This was implemented as a circuit made from discrete components, based on a design by (Ritger, 2002). The first revision circuit is shown in Figure 11 following.

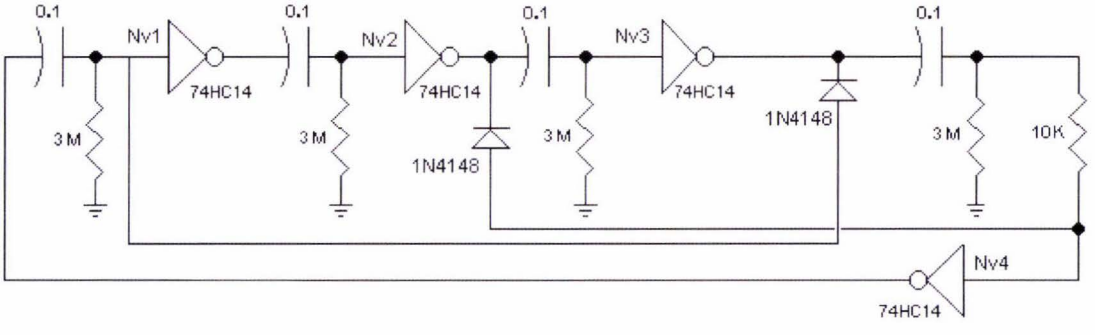

Figure 11: Quadcore Circuit Implementation (Ritger, 2002)

For the "draft" robot build, it was decided to build all the circuitry on prototyping board, or "verroboard". This is a circuit board with a series of pre-printed tracks on it, which allows a custom circuit to be installed thereon by cutting tracks and soldering components into the grid of holes. This was chosen due to the simplicity and low time-cost of this

approach, as well as the ability to easily adapt and modify the circuit without requiring a new circuit board for each change.

Following the development of this "prototyped" verroboard implementation, and in order to incorporate additional functionality for the controller, a new circuit was implemented incorporating variable resistors to allow the circuit's behaviour to be altered "on the fly". Variable resistors were placed in two major locations:

1. In series with the resistor-to-ground of each individual Nv-neuron, to allow the "reaction time" of the neuron to be dynamically adjusted;
2. In series with the outputs from the neurons to the motor drivers, to allow the level of influence between these two circuits to be controlled.

This was designed as a PCB to be machined on the ITE's own PCB-making equipment. The PCB design created for the second revision circuit is shown in Figure 12 below:

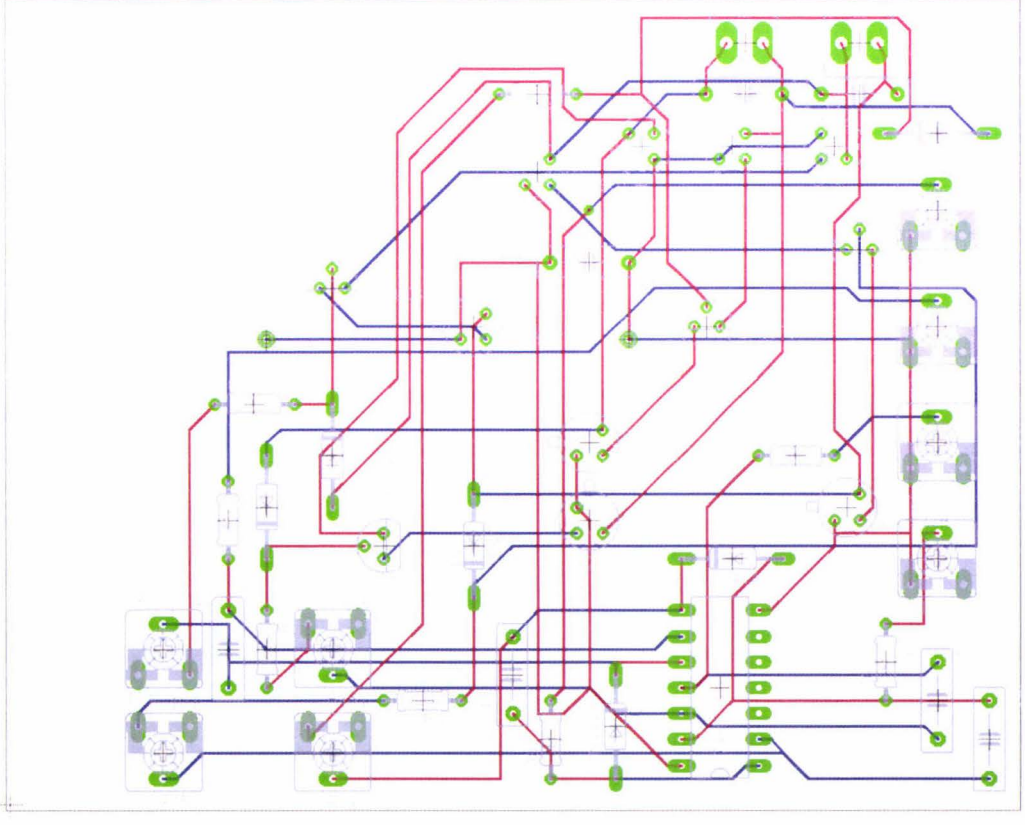

Figure 12: Second Revision Circuit PCB Schematic

Unfortunately, this circuit was never completed to an operational state. Due to unforeseen behaviour in the PCB design package used, the tracks were cut very thin during the (subtractive) machining process. It is thought that the current-limiting effect of these thin tracks was a significant factor in the malfunction of this circuit design – particularly, that the limited current supply to the motors reduced the available drive power below levels sufficient to successfully move the robot.

2.3 Actuators

For any robot, one of the most critical features of its design is the actuators - the means by which the robot interacts with (and thus navigates through) its environment. For walking robots, this is particularly important, as any actuators employed not only have to support and lift the robot, but also have to support and lift themselves.

When electrical motors are used, this can cause a problem - the weight/power ratio does not tend to scale favourably (Allen, et. al., 2003). This means that as the power and number of motors increases, the weight of those motors tends to increase out of proportion, causing a reduced net gain in drive power.

Offsetting this, however, is the fact that, particularly for a robot of the scale proposed here, the motor requires 'high' torque, but only a 'moderate' to 'low' speed. Because of this, it is practical to use a lower power, high speed motor, and then increase the torque to the necessary levels with a gear train. Although this results in significantly reduced motor speed, this is actually beneficial for the situation of a small walking robot.

During the author's initial "thought experiments" regarding this project, based on observations of some other implementations of Nv-net technology, it was thought that the walking motion of this type of robotic walker might be the result of exploiting the mechanical resonance of the robot. While this did not appear to the case in the design used here, there is the potential that as larger robots are constructed, care may need to be

taken to ensure the walking motions do not unstable resonances that could result in the robot tipping over.

Many researchers have developed robotic walkers utilising such 'natural dynamics' for vastly improved energy and control efficiency, for example Pratt's (2000) work with the passive/dynamic planar walking robot "Spring Flamingo". The potential exists for significant efficiency gains if this robot was designed to operate near its resonant frequency - due to the repetitive back-and-forth motion of the employed mode of walking, significant energy savings could be made. However this will not be analysed in this thesis, and the robot will be operated well below such resonant frequencies to eliminate this effect.

The actuators used on the first implementation of the Nv-bot were modified hobby servos, of the type often found in remote control cars. These were chosen due to the light weight, low power consumption, self-contained gear train, and ease of acquirement. As hobby servos are designed as PWM-controlled stepper motors, the only modification required to employ them as an ordinary DC motor/gearbox combination was the removal of the PWM decoding circuitry, and the soldering of new input leads directly to the motor terminals.

2.4 Sensors

Although the circuit design for this robot does not include any "explicit" sensors, the design of the motor drivers allows for a form of "implicit" sensing to take place. This allows the controller circuit, which would otherwise be completely 'blind' to adapt to the world around it, to interact with the environment, using the actuator signal paths as both inputs and outputs to the system. The mechanism allows the change in current drawn, caused by a change in motor loading due to environmental obstruction, for example, to cause a change in the voltage present on the "output" signal line from the Nv-net. If this perturbation reaches a significant level, it can cause the firing or inhibition of the neuron, outside of the normal pattern traveling on the Nv-net.

2.5 *Power Supply*

A critical component in any robotic design is the power supply - the ability to employ technology is directly proportional to the ability to supply energy at a suitable rate for a useful length of time. In order to supply power to the small, mobile robot used in this instance, only one energy source was logical - DC electricity, employed to power DC motors. In order to supply this power, two options were considered - on-board batteries, and an umbilical supply.

Supplying power from an umbilical has several advantages. Firstly, the burden of carrying energy storage is removed from the robot, thus vastly reducing its mass, and thus the load on its legs; Secondly, any bundle of wires running to/from the robot and a remote station could also carry a few more wires allowing for measurements to be taken, or sensor signals to be read from the robot. This would be a great help in terms of detailed testing, as it would allow much more detailed and 'concrete' measurements to be taken, rather than relying on more abstract measurements to categorise robot behaviour

Additionally this would allow the robot to use a power supply that a) would not degrade over time, and b) is able to supply larger amounts of current without degradation. This would remove the need to consider the drop-off of battery performance as its energy is depleted, while simultaneously allowing the control circuitry to better support the 'demands' of the motors in moving the robot.

One major concern in the consideration of an umbilical-based power supply system is also resultant of the robot's low mass. Because the robot has very little mass, any umbilical cabling would need to be designed such that it introduced no lateral forces or tipping moments to the robot. This would probably necessitate either the holding of the cable by the experimenter while the robot was in motion, or the development of a frame that would suspend the cable in such a way that it did not disturb the robot's operation.

The designs considered for such a frame followed two primary lines of thought. The first was to create a rigid frame somewhat higher than the robot, with 8-10 cm of clearance from the robots corners. The frame would then be supported by means of rollers or other low friction "feet" for the platform's legs, and would be moved with the robot, with the umbilical suspended from the frame's "ceiling". In this way, any tension applied to the umbilical would be likely to shift the frame, thus alleviating the effect on the robot, and additionally it would still allow simple intervention by the experimenter to resolve any undesirable situations.

The primary drawback of this approach is that it is only viable on flat smooth surfaces, where the frame can rest securely on the surface without tipping or sliding, and yet can be easily moved without significant movements that could be transmitted to the robot via the umbilical.

As an alternative to this arrangement, a frame was considered based on a philosophy of encompassing the robot's operative area, rather than continuously following the robot. Such a frame would thus be somewhat larger, most likely 1 - 1.5m at a side, and somewhat higher to improve the rigidity of the structure. The umbilical cable would, similar to the mobile design, be supported by the overarching superstructure of the frame; however, due to the non-mobile nature of this design, the cable supports would need to be designed such to allow the experimenter to keep the umbilical neutral relative to the robot.

It was thus decided, due to the additional complexity and variables that would be introduced by pursuing an umbilical-dependant robot, that it would be simpler and more measurable to use an on-board power source - a battery. In designing such a power source, the primary areas of concern were the voltage supplied, and mass cost. In order for the logic gates central to the nervous network to work correctly, a supply voltage of between 3.5 - 7.5 V was required. In terms of standard battery sizes, the simplest way to supply this was the usage of two or three AA cells, connected in series in a battery holder.

The other prime concern in the design of a battery-supplied robot is the mass of the power supply. As batteries rely on storing electrical energy in chemical form, a significant amount of mass (relative to the Nv-bot) is required to store the necessary energy. This is particularly significant for the actuators, as they are required to lift a portion of the robot's mass of the ground - any addition to the robot's mass will require the motors to draw more power for equivalent performance, this leading to faster draining of the power supply. However, the energy content and supply capabilities of standard AA cells should be amply sufficient to offset the additional power cost induced by the added mass.

2.6 Interface

The interface refers to the circuitry used to provide an interpretation service between the Nv-net and the "real world", and thus allow meaningful interaction to occur between them. By necessity, this infers a two-way link, allowing the environment to influence the Nv-net via the robot's sensors, and the robot to influence the environment via its actuators.

Because of the nature of this interface, this is one of the prime means by which an Nv-net controlled robot is programmed, by altering the "interpretation" of signals traveling between the environment and the robot.

For the purposes of this robot, all interaction between the robot and its environment occur via the motors. Thus the interface needs to interpret signals from the Nv-net controller to allow the robot to walk, and ideally provide a feedback path to allow the Nv-net to dynamically modify its gait in response to the environment.

For this implementation, the interface component of the design is represented by the H-bridge motor drivers used to transform the control signals from the Nv-net into drive signals of sufficient power to drive the motors. As noted in the "Sensors" section earlier, the h-bridge used here is designed such that some information about the motor's

interaction with the environment is able to be "transmitted" back to the controller. This h-bridge circuit was developed by Wilf Ritger (2002), and is shown in Figure 13 following.

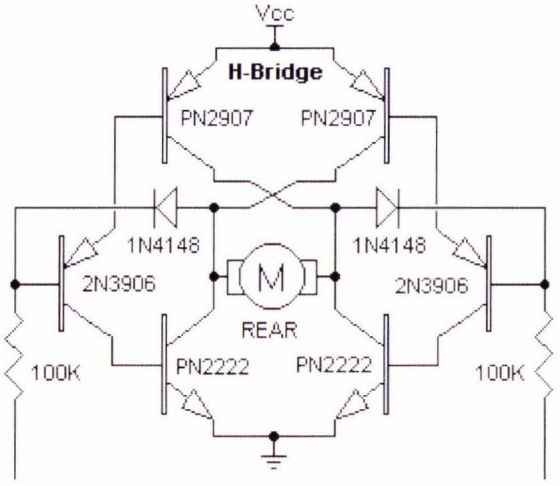

Figure 13: H-bridge Circuit Diagram

2.7 Legs

An overarching philosophy of BEAM design (a major school of thought concerning the development of Nv-net technology), is that the behaviour of a robot is resultant of a combination of all its attributes - controller, mechanics, structure, etc... This is also reflected in academia, such as Geng's (2007) study of the coupling between neural controllers and their mechanical context.

This being the case, the legs of the robot are of critical importance to the behaviour of the robot in two areas - the material characteristics of the legs (i.e. rigidity, elasticity), and the orientation and displacement of the leg endpoints relative to the robot's centre of mass (COM).

2.7.1 Material characteristics

Two materials were considered for usage as the Nv-bot's legs - 1.5 mm aluminium wire, and 1 mm copper wire. Copper wire was used in the initial prototype, as it seemed ideal due to its malleable nature - making it very easy to reconfigure into different leg shapes. Additionally, copper wire possessed a level of "spring", or elasticity, that it was thought may prove useful to the robot in its operation. However, copper wire's major strength was also its primary weakness - the soft wire was barely able to support the robot's weight at rest, and the introduction of the robot's attempted motion into the experiment rapidly resulted in the robot lowering itself onto its belly (as the legs simply folded out from underneath it).

Following this failure, the aluminium wire was sourced from Gordon Warren, and proved much more suitable to the task required of it. The aluminium wire, although significantly stiffer, still possessed a degree of elasticity, and was also able to be made into more accurate bends. The aluminium legs stood up well under robot motion conditions, and thus were selected for continuing work.

2.7.2 Shape

The shape of the legs is thought to have a significant effect on the robot's motive behaviour. There are two major reasons for this, their influence on the transmission of forces from the motors to the walking surface, and the interaction between the leg's feet and the walking surface.

As the legs essentially act as large levers driven by moments supplied by their respective motors, the shape of these legs has the ability to significantly alter the forces these moments exert on the surrounding environment.

However, for the rear, horizontal-stepping leg particularly, the primary deciding factor in the shape of the leg is the capability of the foot to deal with the intended walking surface.

For this robot, as the primary operating environment is likely to be soft arterials such as carpet, a leg shape was created that should help to increase friction on the foot pushing the robot forward, while decreasing friction on the opposing foot being dragged forward across the walking surface.

2.8 Chassis

The major concern of the "chassis" design for the robot is to provide a lightweight shell that will

- hold the circuit board
- hold (or support) the power supply)
- hold the motors
- maintain a fixed relationship between the robot's center of mass and the motor output shafts

Thus, the design of the chassis will need to be easily reconfigurable in order to accommodate alternate geometries, as well as being quite rigid in order to maintain a particular geometry while withstanding the weight of the power supply, and the forces induced by the robot's motion.

Several candidate materials were considered, including aluminium, balsa, cardboard, and pre-built plastic enclosures (i.e. "jiffy boxes"). It was decided that aluminium would be preferable for a final "proof" design of the robot, however due to the time necessary to design and machine such a chassis, and the permanence of the resulting structure, it was decided to pursue alternate materials for the "draft" robot prototypes used in the design and experimental processes.

Two primary materials were considered for the draft prototype of the robot. The first of these was balsa - a very light wood with considerable strength. This would be particularly advantageous for a small robot frame, placing minimal burden on the robots

motion, while providing a sufficiently stiff frame to maintain the geometrical relationship between the robots critical movement systems.

The alternate choice was heavy-duty cardboard. This was readily available, and had similar characteristics in that it was light, strong, and easily workable. The biggest single advantage of cardboard over balsa is the ability to construct a robot chassis as a "net" - a foldable 2D pattern that results in a 3D construct. Using this technique, it was far simpler to construct varying chassis designs with minimal time and effort using cardboard, and it was this reconfigurability, and the simplicity of having an enclosed 3D shell on which things could be anchored, that contributed significantly to the decision to use cardboard for the draft prototype.

However, as the major focus of this experiment is on the electrical tuning of the Nv-neurons and interface, and the mechanical tuning of the leg contact points relative to the robot's centre of mass, it was decided that chassis construction was a secondary priority. Due to this decision, it was decided that, providing a chassis structure met the criteria outlined in at the beginning of this section, no further efforts would be made to design or implement a more durable or substantial frame for the robot until viable tuning results had been produced. This allowed priority allocation of time and mental resources to the critical elements of the project, while still allowing provision for the construction of a "better" chassis.

2.9 Robot Stepping Stances

Under the control of the quadcore nervous network described in Controller, the robot performs a particular four-stance gait. The simulations however, add an additional stance that is not actually part of the gait – the "square" stance, a starting stance used for simplicity in designing the model & simulations, and ensuring consistency in their start conditions.

2.9.1 Stance 0

The "first" (strictly, zeroth) stance is the "square" stance – the only possible combination of joint angles which places all four "feet" in the same plane. The calibration of the joints will be such that a rotation of zero degrees on both joints will result in this stance. Figure 14 shows the simulated model in stance 0, highlighting the major features of this position.

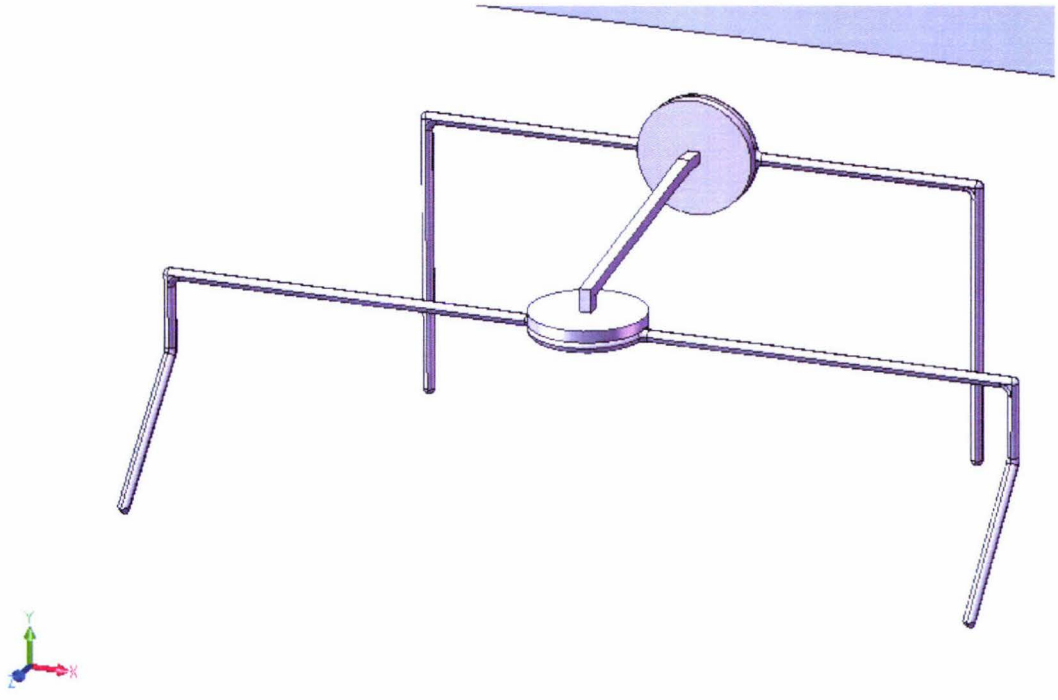

Figure 14: Stance Zero (V-leg Joint unrotated, H-leg joint unrotated)

Although this stance does actually appear in the gait proper as a transitionary state during the movement between other stances, it is not a true part of the gait. However, it was incorporated into the simulated model as the starting position of the virtual robot, for a variety of reasons.

Firstly, it was unknown how much effect the starting posture of the robot could have on various elements of the walking process. The square stance was chosen for this role, as it is the only stance that is unaffected by changes in the overall robot geometry – other

stances can be significantly affected by changes in the height or width of the legs, as the same joint angles will leave the robot in vastly different postures with varying geometry parameters.

Secondly, the square stance provided the simplest way of ensuring all legs started on "equal footing", as it were. Due to the necessity to start a simulation execution with the model a few millimeters above the ground plate, the square stance ensures that no unexpected motion is introduced to the robot when the model lands on the ground plate following simulation start.

2.9.2 Stance 1

Stances one and three are opposites and can thus be defined to either "side" arbitrarily; for the sake of this discussion, stance one will refer to the counterclockwise (CCW) positioning of both the horizontal- and vertical-turning legs. This configuration is the precursor to the robot's next right-foot-forward step, and descends from stance 4 by the CCW rotation of the v-leg joint. A diagram of this particular stance is shown in Figure 15 below.

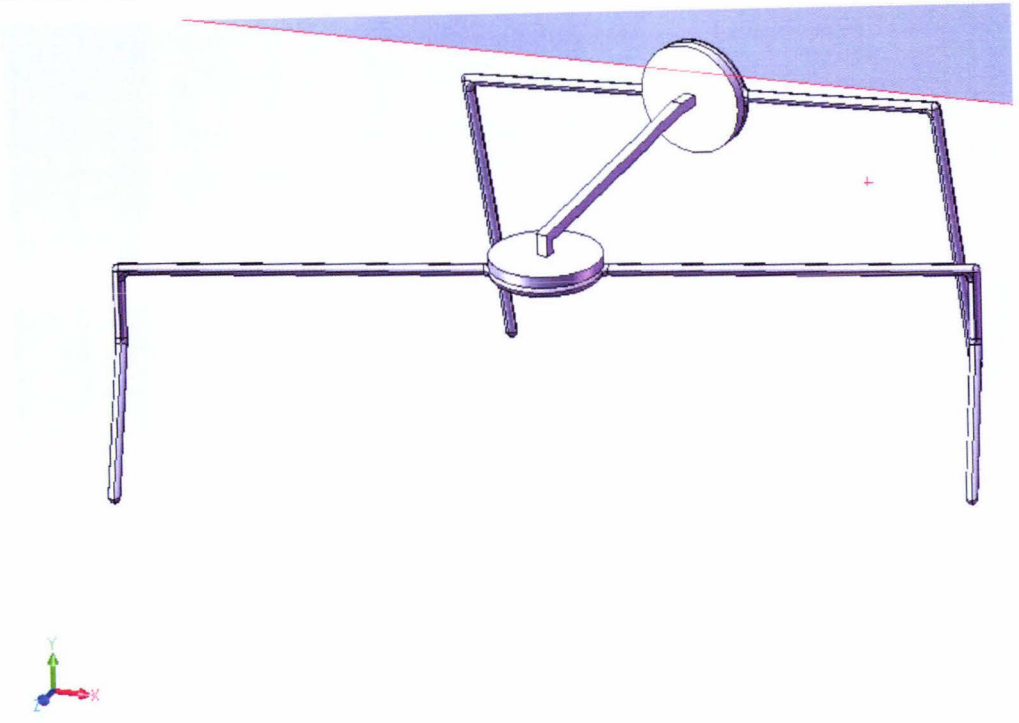

Figure 15: Stance One (V-leg Joint CCW, H-leg Joint CCW)

2.9.3 Stance 2

The next stage in the robot's walking pattern is the CW movement of the h-leg joint, thus stepping the robot's right foot forward. It is relatively common for the robot to "tip" during this transition, transferring from having the rear-left tripod of feet resting on the ground (front left foot, and both rear feet), to resting on the front-right tripod. The typical stance resulting from a tipping step is shown in Figure 16 below; for those geometries which do not undergo tipping, the robot remains balance on the rear-left tripod, but resolves naturally to stance three via the standard transition. The tipping motion, and its cause, is discussed in section 3.5.1: Vertical Leg.

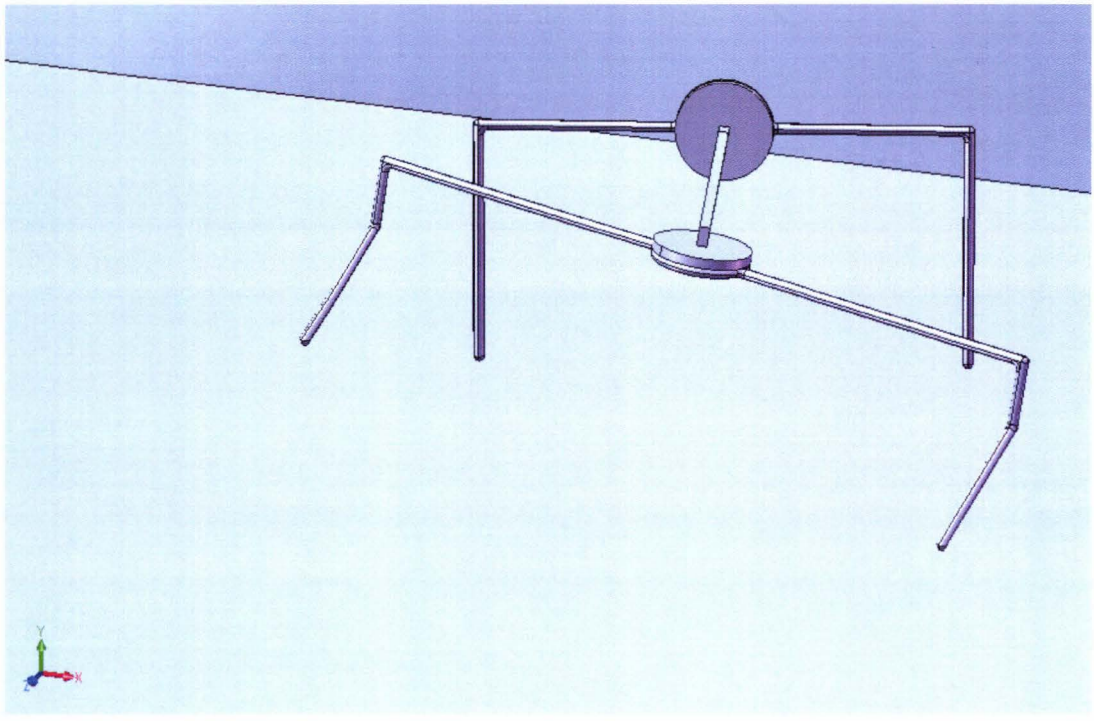

Figure 16: Stance Two (V-leg Joint CCW, H-leg Joint CW)

2.9.4 Stance 3

Stance three is the opposite-side equivalent of stance one, and as such has no particularly notable qualities other than those already outlined for stance one. Stance three's place in the walking sequence is shown in Figure 17 below, and is resultant from the CW movement of the v-leg joint.

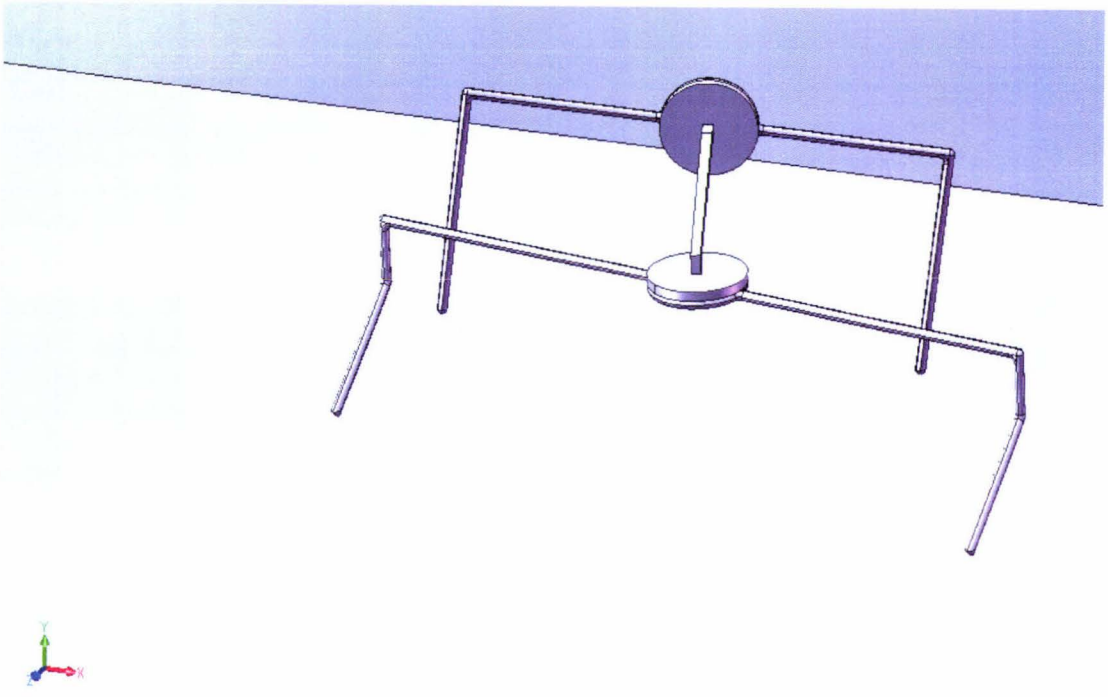

Figure 17: Stance Three (V-leg Joint CW, H-leg Joint CW)

2.9.5 Stance 4

The final stage in the robot's step cycle is the left-side step, derived from stance three by the clockwise movement of the h-leg, resulting in a stance directly opposite that of stance two. This is shown in Figure 18.

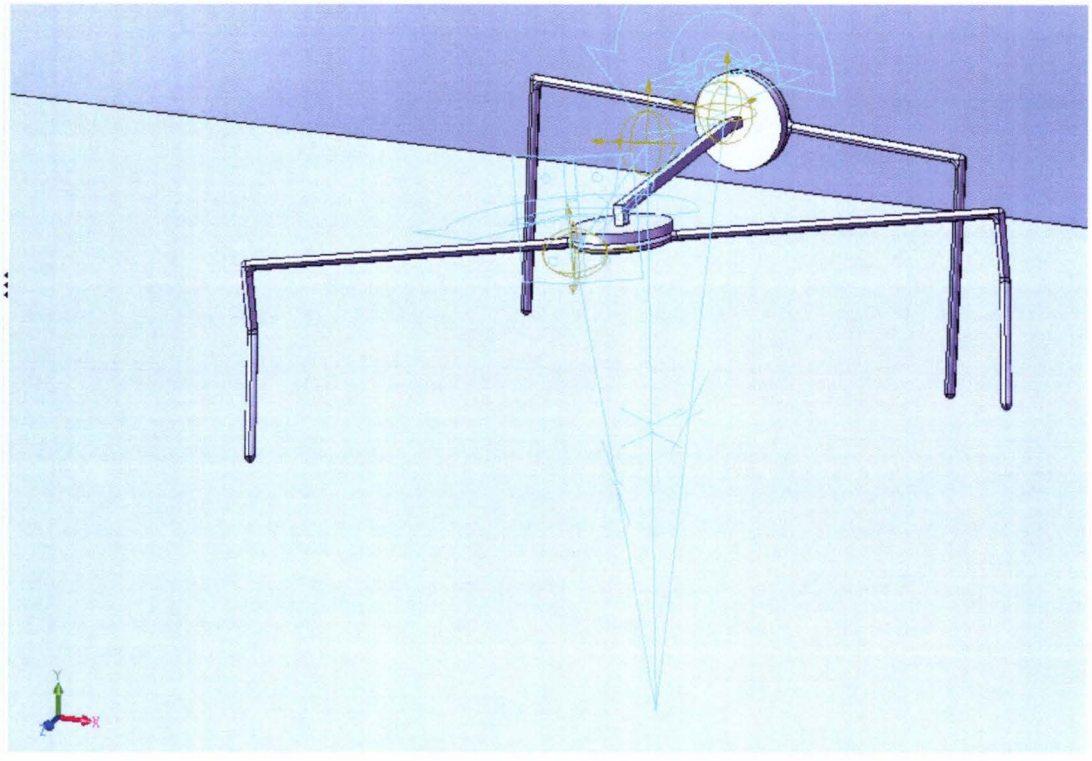

Figure 18: Stance Four (V-leg Joint CW, H-leg Joint CCW)

2.10 Untuned Behaviours

Before any tuning was applied to the robotic platform, a series of observations were carried out under nominal operating conditions. These essentially involved turning the robot on, and observing any resultant behaviour.

The initial observed behaviour upon activating the robot for the first time was nothing - the motors on the initial implementation of the robot were simply not powerful enough to support the mass of the robot. After the modified RC-servos were installed (section 2.3: Actuators), the initial reactivation of the robot resulted in a significant surprise - the robot began walking, somewhat jerkily, but progressed forward along a curving path for a total of about 6-7 steps, before the robot ran out of space to move and was manually deactivated.

2.11 Summary

The chosen mechanical context represents a simple quadrupedal walker, with two driven legs. This was chosen due to the relative simplicity of its design, its common nature in Nv-net experimentation, and its ready extensibility to add additional capability to the walker base. As outlined earlier in the literature review, robotic walking is currently a prime research goal, and significant work in Neuromorphics and Biomimetics is being furthered in this area.

A real world implementation of the targeted mechanical context was constructed, and designed to be simple, lightweight, reconfigurable & inexpensive. Upon completion, this model exhibited "untuned" walking behaviours, and provided insight & intuition for the development of simulations for further analysis of the mechanical context. This developed intuition was then valuable for the construction of a mechanical simulation that would produce behaviours similar to that of a robot controlled by an Nv-net.

3 Simulation

3.1 Introduction

In order to provide an objective, measurable environment in which to undertake trials to analyse the mechanical context, and in order to provide an environment in which aspects of the world and robot could be easily changed, it was decided to pursue a simulated solution. Of primary concern was the ability of any such simulation package to accurately model 3D motion, and in particular, accurate collision detection and friction modeling to allow for the model to "walk".

The selection of an appropriate simulation tool will be discussed, and the construction of the model will be detailed. Finally, the overall simulation environment settings used to provide a context for the implementation of the simulations will be discussed.

3.2 Simulation Tool

During the initial consideration of the possibilities of simulating the robot, it was hoped that a "complete" modeling of the robot would be possible – i.e. that both the mechanical and electronic components of the robot could be present in an integrated simulation.

To this end, the 3D CAD package Solidworks was chosen to construct the robot model, as the ITE has access to licenses of this software, and the author is familiar with its usage via his undergraduate studies. Solidworks allows for the creation of 3D parts as a combination of 2D sketches and three dimensional projections of those sketches; multiple parts can then be combined into an assembly, held together in a configuration defined by "mates" between various parts' features.

In order to implement the motion simulation component of this model, the Solidworks add-on COSMOS/Motion was used. COSMOS/Motion is seamlessly integrated into the

main Solidworks package, and will even automatically generate joints in the modeled mechanism based on the mates in the assembly. It is capable of running motion simulation and returning detailed results, including animations, and exporting any selection of those results to Excel, CSV, or several other file formats.

It was this ability to export detailed results of the physical simulations, as well as the simple yet powerful CAD tools Solidworks provided for the creation of the robot model, that cemented Solidworks as the simulation tool of choice for this project.

3.3 Pilot Run

In order to test the potential of the simulation to simulate a completely mobile robot, and to determine whether it was possible to design such a simulation to exhibit actual walking behaviours, it was decided to implement a rough "pilot run" simulation before proceeding with the design of simulation runs for analysis.

To this end, a very rough approximation of the robot's geometry was created, without attempting to build a perfect replica of the real world robot. The resulting design was thus implemented to allow for simple integration of the model into COSMOS/Motion, and thus allow for exploring the necessary settings & driving function possibilities without any complications that could be introduced by more accurately representing the robot. This simple representation of the robot is shown in Figure 19 below.

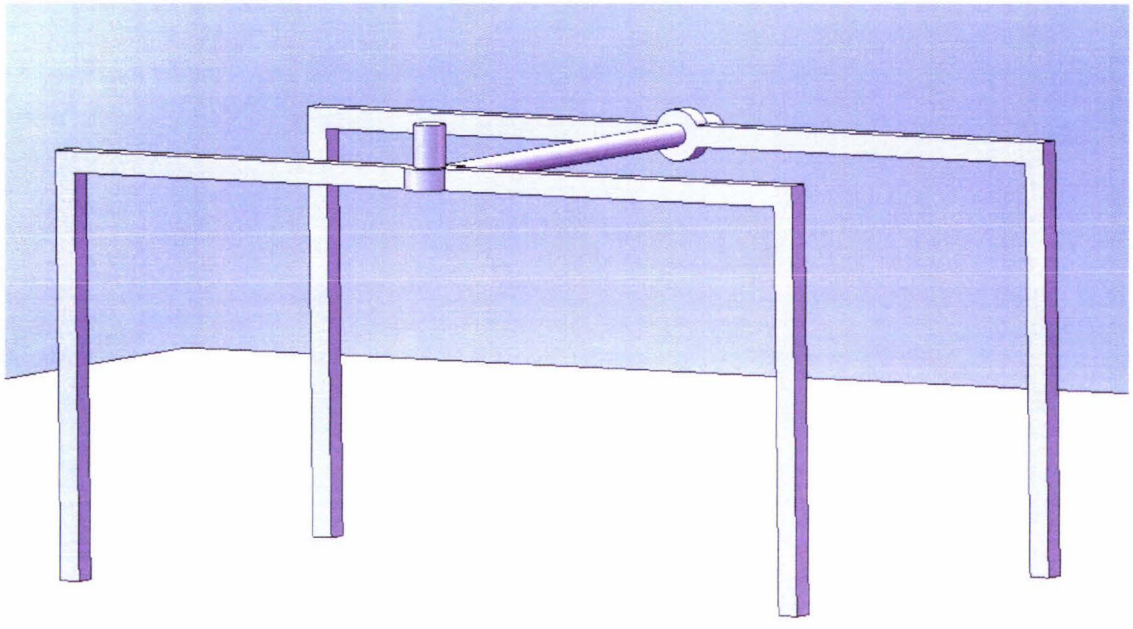

Figure 19: Rough SolidWorks model of the Mechanical Context

Of particular note regarding approximations made in this model are the "feet" (i.e. leg ends) and the axial studs on the legs. The feet used in this model are square in section rather the "point"/rounded feet of the actual robot to decrease the time investment required to build the 'draft' model.

At the rotary axis of each leg (i.e. the axis at which the leg is driven by the actuating motor), a circular protrusion was added to the leg model to aid in the creation of reference axes. These were then used to define the necessary joints and relationships to cause the model to exhibit properly constrained movement. All created "mates" (i.e. inter-part relationships) were defined so as not to cause any structural interference or conflicts, both "physically" and with the simulation's solver.

Once the quirks in the model simulation systems were sufficiently explored and understood, a similar exploration of the motion simulation functionality was undertaken. This was less for the purpose of checking for glitches, but rather with the intent of determining the practical effect of various settings integral to the simulation.

One such setting which had a demonstrably significant effect was the material characteristics of the "ground" surface. The initial "default" settings of the simulation for this surface are shown below in Figure 20.

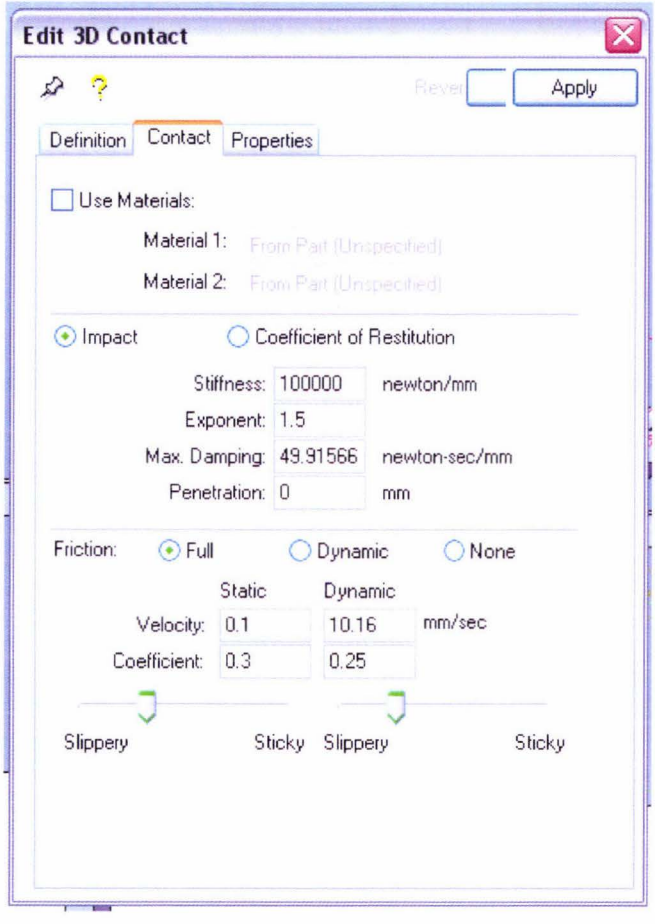

Figure 20: Initial COSMOS/Motion Ground Surface Contact Properties

These settings resulted in a very "bouncy" surface, introducing significant amounts of jitter into the simulation, decreasing accuracy while simultaneously increasing the computational effort required by the solver. This effect was additionally exacerbated due to a limitation of the model's implementation – because it is very difficult to place the moving model directly "on" the ground surface, the robot model's starting point was located several millimeters above the surface, in "square" stance, as shown below in Figure 21.

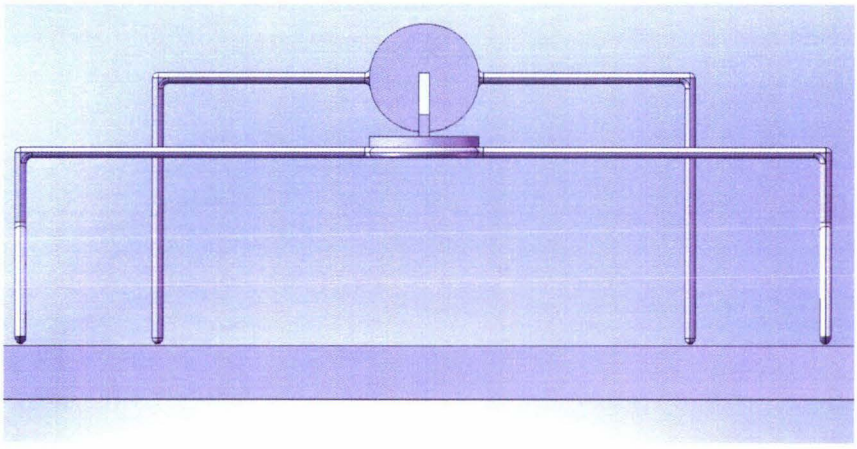

Figure 21: Square stance vertical offset

This allowed the model to run without fear of interference glitches (i.e. without the model becoming part of the ground surface, or projecting beneath its surface), although some further modifications were required to completely eradicate this problem, covered in the Environment section following.

3.4 Environment

A critical component of the modeling of a free-standing mobile robot is the environment in which it will move during the simulation. In this model, for simplicity of analysis, this was implemented as a single square "ground plate". It was sized such that the robot would require several full strides to reach the edges, and locked in place relative to both the Assembly origin, and the Assembly's "top" plane. Additionally, in COSMOS/Motion, this object was defined as a "grounded" object, that is, an object which is immovable in the simulation, and represents the reference "ground" for any moving parts in the simulation.

The major influencing factor in its design, however, is the friction characteristics and flexibility assigned to the material. COSMOS/Motion allows the modeling of both static and dynamic friction during simulations, and as such the friction coefficients on the ground plane were exaggerated somewhat – the static coefficient of friction was made

somewhat higher, and the dynamic coefficient of friction somewhat lower, than would be normal. This has the effect of exaggerating the behaviour of bodies making surface contact with the ground plate. However, in order to ensure this will not unduly affect the outcome of the investigation, a simulation run will be undertaken to consider the effect of friction on the model's behaviour.

A major refinement in the tuning of the robot was the modification of the ground plane material. Initially, the ground plate was set to its default – theoretically supposed to simulate dry steel. However, a major problem with this stetting, exacerbated by the relatively light weight of the robot, and the need to start it slightly off the surface, was that the robot bounced up and down continuously throughout the simulation, resulting in spurious movement when there was none, and intermittent contact for the feet, making it difficult to determine the geometric origins of any resultant motion..

Two solutions were implemented to fix this problem. Firstly, the weight of the robot was significantly increased, to a "unit" weight intended to be representative of a likely real world robot. The primary effect of this added weight, in addition to increasing the force due to gravity, was to make any COM-dependant features of the movements more obvious. As a side-effect, this also acted as a scaling influence on the power graphs measured from the displacement drivers; this is not an issue, however, as the power only serves to give an idea of the drivers' performance and aid in the determination of critical features in the robot's gait, without absolute data being required.

The other implemented solution was to modify the material applied by the simulation to the ground plate. The "springiness" was removed by reducing the penetration allowable, thus making it impossible for any bounce-back to occur. Additionally, friction forces were exaggerated somewhat, to allow for clearly defined features to be seen in any recorded graphs. When applying the observed results of these explorations to a physical robot, this is a major consideration – what surface(s) will the robot be required to navigate, what approximations of their static & dynamic friction coefficients can be

made, and what compromise must be made between the various surfaces to allow the robot to successfully navigate a wide range of environments.

3.5 Mechanical Model

For this simulation, the mechanical model was constructed of three components, two mechanically accurate components to represent the robot's legs, and a geometrically correct component to maintain the desired spatial relationship between the joints of the two legs. The distinction is made as, although the legs closely represent the legs of the real world robot, the "link" component does not – it merely geometrically represents the relationship imposed on the structure by the presence of the chassis.

3.5.1 Vertical Leg

The robot's "front" leg is the leg with a rotational axis along the horizontal axis – thus the leg "steps" in the vertical plane. Due to this fact, the leg is commonly referred to throughout this thesis as the vertical leg, vertical-stepping leg, or v-leg. This leg is responsible for shifting the robots centre of mass in such a way that facilitates walking. Because this leg operates purely in the vertical plane, no possible contribution to forward movement can be directly made by this leg, unless the change in geometry caused by its movement causes the robot to "tip".

This "tipping" behaviour is a special geometric case where the robot's centre of gravity passes across the robot's "centre of balance" - the line between the hypotenuse of a triangular stance – the stance that results if any rotation is present on the front leg. The following figure shows this sequence of events to illustrate the geometrical process that allows tipping to occur.

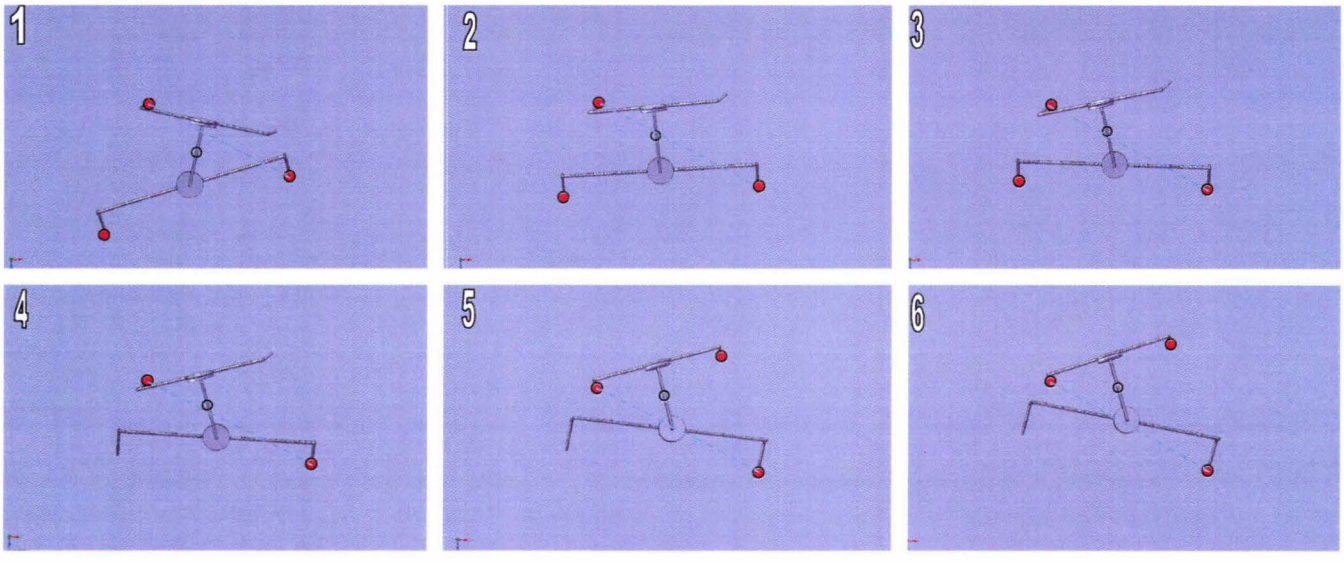

Figure 22: Sequence of Events for Tipping within a Step

In terms of design, the vertical leg is very simple – it was only necessary to measure the legs that were to be used for the real-world implementation of the robot, and create a single planar profile in Solidworks that could be extruded to form a 3-dimensional structure. In order to approximate the leg's shape, the resulting "square-edged" structure was filleted all over, with a fillet diameter equal to the thickness of the extrusion. This created a wire like structure, with round "feet", which is a reasonably close equivalent to the real world robot.

The effect of the foot shape on the robot's behaviour is likely to vary greatly across different surface types, and there was some concern that the usage of round feet might significantly increase the simulation time required to model the legs' movement. However it was decided to continue using the rounded legs, as these were likely to

maintain the most consistent contact profile as the legs move and rotate. Due to the time limitations of a Masters thesis, this design aspect of the Nv-walker mechanical context will not be considered in depth.

3.5.2 Horizontal Leg

The rear leg of the robot rotates around a vertically oriented axis, and is slightly more complicated in structure than the v-leg. Although there are several h-leg designs common within the community, this particular design was chosen due to its likely improved friction profile when operating on surfaces such as carpet and the outdoors. The following figure shows the chosen design geometry for the h-leg.

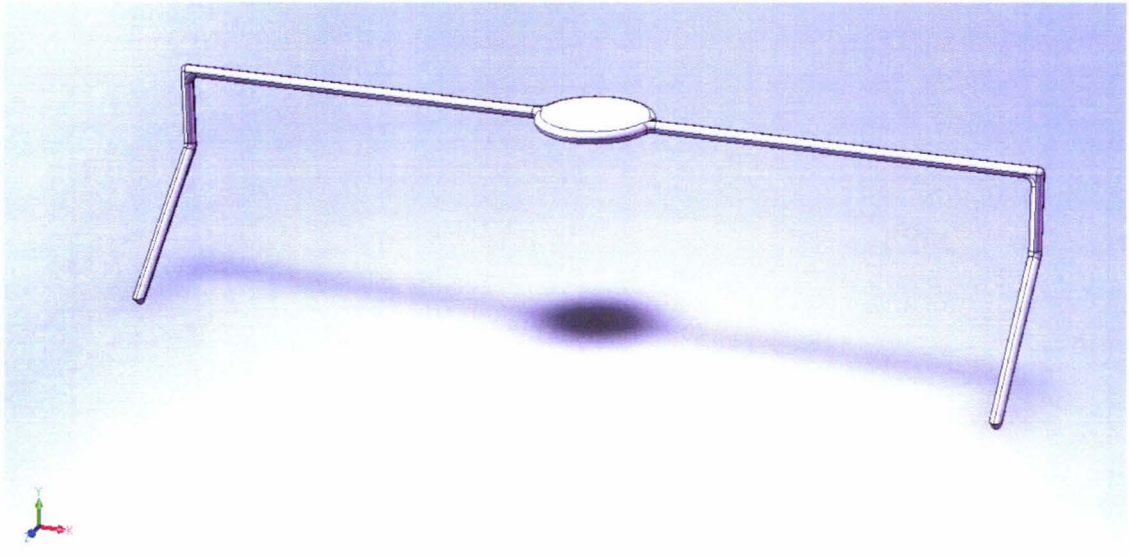

Figure 23: Simulated H-leg Geometry

3.5.3 Chassis Link

The chassis link is the final component in the model, and holds the h- and v-legs in a fixed geometric relationship between each other. The major complication in the design of this component relates to this purpose, and thus the model's geometry must be

designed such that, as the dimensions of the legs change (during the course of the simulations, for example), the chassis link changes also to maintain their relationship.

In order to implement this, large discs were incorporated into the model of each leg, centered about each leg's respective centers of rotation. Similar discs were than created at the end points of the chassis link, to provide for a reference point, and also to ensure that very simple mates (i.e., co-radial mates) between the various parts could produce the desired effect.

The relationship between the legs is defined such that, if each leg's rotation is 0, all four feet lie within the same plane. Connecting these discs was a diagonal member, with its dimensions designed such that, as the parameters of the various legs change, the chassis link will maintain the necessary relationship to ensure that the robot's "square" stance remains intact.

The chassis link is currently designed under the assumption that the h-leg will always be shorter than the v-leg; the ability to deal with other cases could be provided for, but these cases are beyond both the need and timescale of this project.

3.6 Motion Model

In implementing the motion model in COSMOS/Motion, three motion relationships were also used: One each to correctly create the joints between the legs on the link, and one container to manage the contact between any part of the model and the ground plate.

3.6.1 Joints

Between mechanically linked objects, joints are defined which express and manage the constraints of the mechanism. Only two joints are present in this robot, implemented as 1-DOF actuated rotary joints, representing the motors on the real-life robot.

These joints are defined as being related to two bodies, and constrained about some particular axis of rotation. The joints implemented in the simulation are shown in the Figure 24 and Figure 25 following.

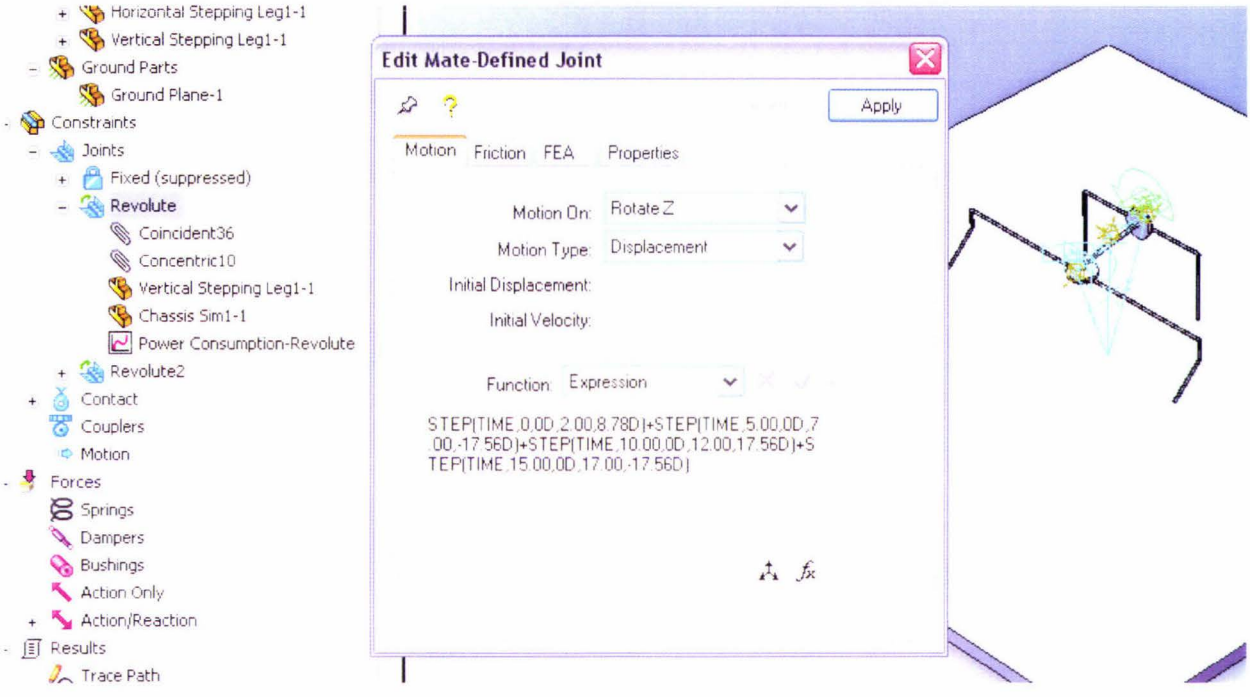

Figure 24: Horizontal-stepping Leg Joint on Simulated Mechanical Context

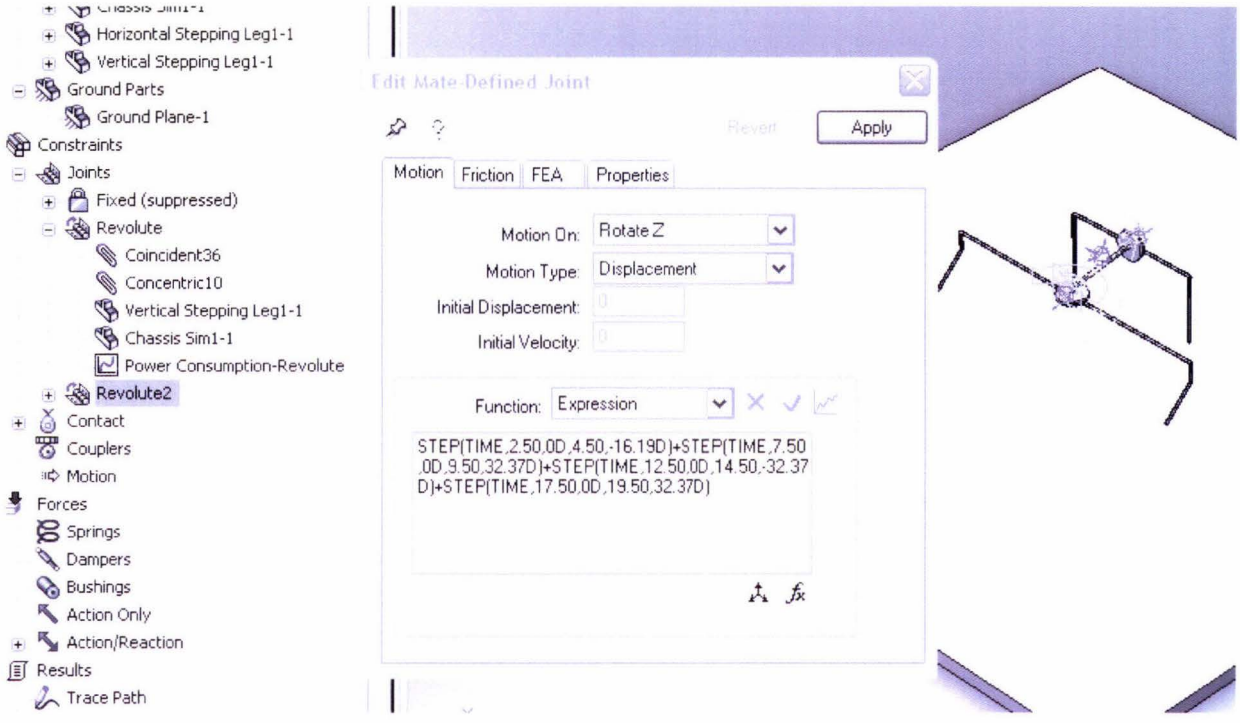

Figure 25: Vertical-stepping Leg Joint on Simulated Mechanical Context

3.6.2 Contact

In order to manage contact between entities, COSMOS/Motion requires the creation of an object that manages and associates "collidable" objects with one another. In the case of this model, the "3D Contact" container object was used, and the ground plate associated as the grounded object in the model, while the others were established as being dynamic. The settings for this contact object are shown in the screenshot Figure 26 below.

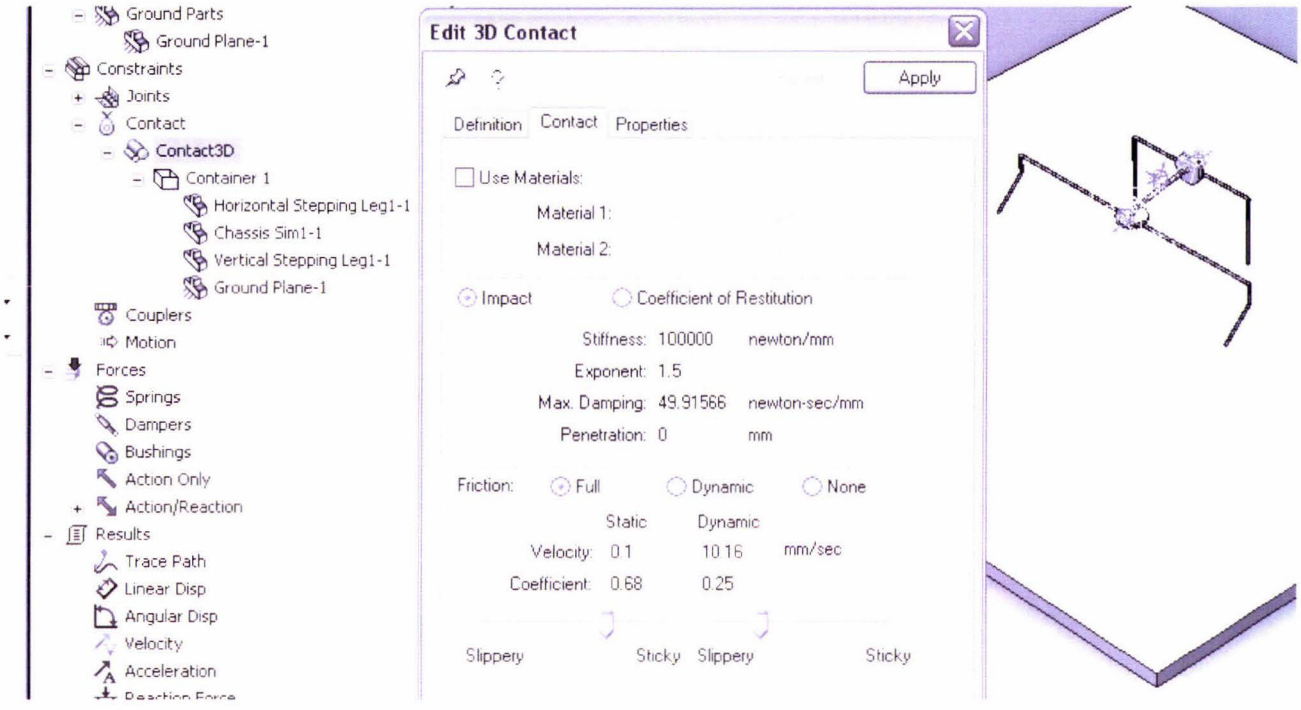

Figure 26: 3D Contact Definition for Simulated Mechanical Context

3.7 Simulation Settings

The settings used to configure the simulator were the same for each of the simulations executed, and are shown in Figure 27 below. These settings were chosen to provide a high accuracy simulation, particularly the production of ~2000 "frames" over 20 seconds (100 frames of simulation per second), the usage of the "Precise Geometry for 3D Contacts" option (which facilitated more accurate interactions between the feet and the ground plate), and the usage of a fine accuracy setting in the solver configuration. The WSTIFF solver was chosen after trialing various solvers using the "pilot run" simulation - the WSTIFF algorithm produced the cleanest results, with the minimal glitching.

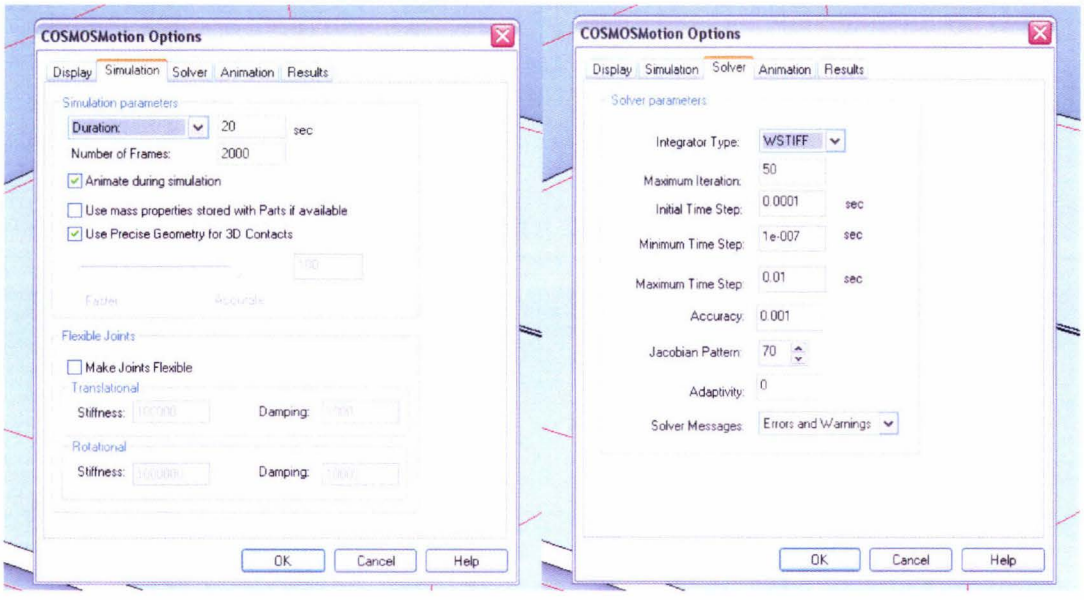

Figure 27: Simulation Settings

3.8 Summary

In order to investigate the behaviour of the mechanical context studied in this thesis, and due to complications in developing a real-world implementation of it in a robot, a simulation was developed in the SolidWorks CAD software package, using the COSMOS/Motion add-on.

A rough model of the mechanical context's geometry was then developed within this simulation environment, and a successful pilot run undertaken as a "proof of concept" test. This also served to provide information regarding the simulation package's influence and necessary design considerations, providing a framework and knowledge base from which the primary simulations of this study could be developed.

A more representative model of the mechanical context was then designed, and implemented in such a way as to be useful for a wide variety of possible trials. Solutions were found to problems identified in the pilot run, and the various parts of the context

were linked using appropriate simulational relationships, in order to manage inter-part contact, as well as enabling the joints to be controlled by a time-domain transfer function.

The internal settings of the COSMOS/Motion simulation engine were then fine-tuned to provide results that were as-accurate-as-practical; this combined configuration of simulated model and simulation settings was then used to create a template "document", such that each simulation or trial could be derived from the same starting point.

4 Methodology

4.1 Introduction

The development of experimental methodologies for the simulated mechanical context was approached using a hybrid approach of top-down and bottom-up techniques.

From the bottom up, the problem was considered in terms of simulation inputs - what factors would have a significant impact on the robot's walking behaviour. This includes influences such as leg shape, relative leg sizes, environmental friction, motor turn angles, etc...

Considering the problem from the top down, the required simulation outputs were listed - particularly that a stable, identifiable operating point needed to be found for driving the simulated robot's actuators to effect walking, and the testing of this operating point to determine its optimality.

The following sections present the methodologies and simulations employed by this project, approached from each of the above described angles.

4.2 Experimental Design

In exploring the relationship between the leg geometry of an Nv-net walker and its gait, there are a multitude of factors that require consideration.

Due to the mechanical focus of this thesis, the control of the robot's legs is only considered from the perspective of the system output – it will be left to a future study to analyse the tuning and influence of the internal controller and interface design on such robots' performance.

This appears superficially contrary to the BEAM design methodology often employed in the design and implementation of such robots, however in order to solidify the theoretical underpinning of such machines, it is not feasible to considerable all factors at once. The research conducted in the course of this thesis revealed significant insights into the influence of the robot's leg geometry of the walking gait; this can be used as a stepping-stone for future research into the interface and controller, and eventually to developing a theoretical understanding the entire system as a truly mechatronic device – a artificially design symbiotic system involving highly interdependent subsystems of different engineering fields.

To analyse the mechanical properties of the mechanism, it was first necessary to isolate the potential factors that could influence the robot's walking gait, and to determine suitable metrics which could be used to quantify the behaviour of the robot's gait, and the outcome of explorations into individual components of the robot's geometry.

There are three major components to the robot's mechanical structure when viewed from a geometric perspective: the front, vertical leg (or v-leg), the rear, horizontal leg (or h-leg), and the linkage between them that maintains the relative positions of the functional axes of these legs. Connecting this linkage and the relevant legs are revolute joints with 1 DOF. The following figure shows an exploded view of the robot, to illustrate the relationship between these parts.

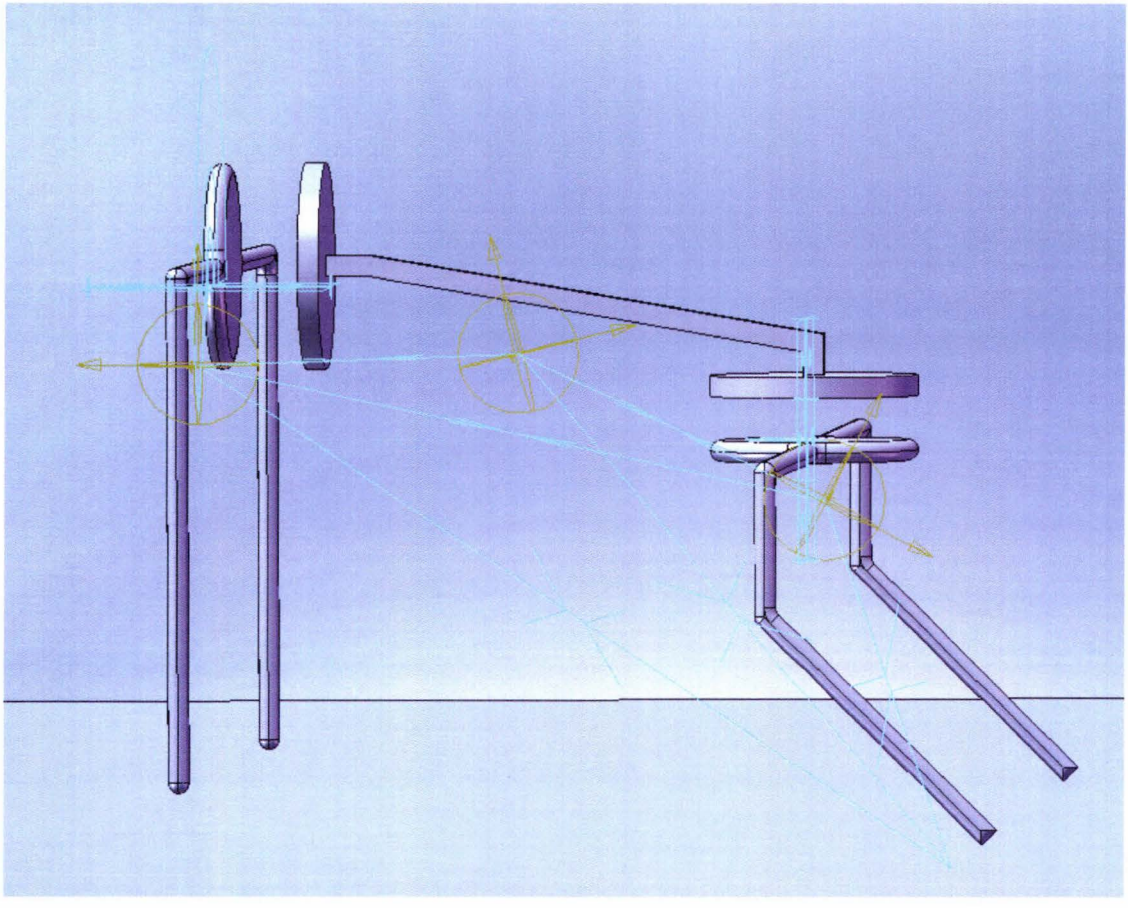

Figure 28: Exploded view of Simulated Robot showing Joint/Link Geometry

Under the robotic design outlined in section 2 (Robotic Platform), the vertical leg is the simpler of the two legs, and its behaviour can be defined by three parameters which determine the influence of the leg on the rest of the mechanical system.

Firstly, and certainly the factor most influencing the design of the control and interface portions of the robot, the motion of the h-leg joint is critical to defining the behaviour of the robot. It is assumed in this project that the motion will be perfectly mirrored around the "square' stance, however in the real-world, it is often necessary to apply external forces to the rotation of this joint, in order to ensure its centricity (such as the construction of hardware stops, the incorporation of limit-switches, or the usage of springs to bias the legs). Without the centricity of the leg's rotation as a factor, this leaves the maximum angular displacement of the leg as the sole factor here; the angular

velocity at which the leg moves may be a minor factor, particularly at higher velocities, however this factor will not be considered.

The other two factors involve the shape of the leg itself, its height and width. The v-leg is planar, and thus posses no meaningful dimension in the z-axis. In concert, these two parameters determine the force the motor's torque applies to the ground, and will likely have other influence over the behaviour of the robot's gait, when considered in light of other components of the robot's geometry.

The h-leg is slightly more complicated that than the v-leg, in the sense that it possesses three dimensional properties, height, width and depth. However, to express the shape of the leg accurately, it is necessary to construct it from *four* dimensions – the leg's height, width and depth (y, x, z), and also the angle (alpha) at which the lower portion of the leg departs from the vertical. Figure 29 below demonstrates the influence of each of these quantities on the shape of the leg.

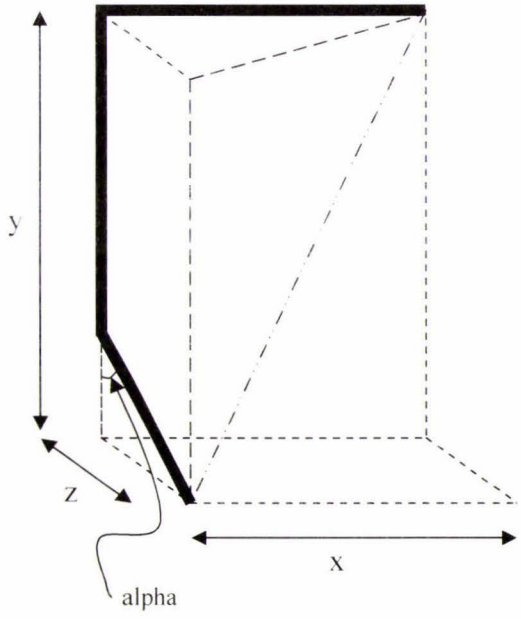

Figure 29: H-leg Geometry

For the purposes of this investigation, and in part due to the limitations of the simulation environment, the influence of the angled portion of the leg geometry will not be considered. This shape is designed primarily to aid the movement of the robot across real world surfaces, such as carpet or sand, by increasing the force the turning h-leg is able to apply before slipping of the feet occurs.

Much like the v-leg, the width and depth of the leg determine the transmission of force from the motor to the environment, and as such could likely have a significant effect on the motion of the robot. Additionally, all three leg dimensions are likely to influence the motion of this part in response to other movement in the geometry.

Also similar to the v-leg, the other defining influence of the h-leg component is the joint between the h-leg and the linkage between the leg; more precisely, the maximum angles to which it turns. Again assuming perfect centricity of the leg's motion, the maximum rotation angles and their accompanying velocity characteristics are the parameters of the joint that can influence the behaviour of the robotic system.

Lastly, the linkage between the two legs has several critical influences. One of considerable import is that the robot's centre of mass is located in (or very near to) this part, as it represents the main chassis of the robot, were all circuitry, batteries, and non-moving structural components are located. As the legs move, the COM will shift slightly, but will always be "attached" to this component.

Additionally, this component inherently determines the relationship between the motion of the two leg's, as it fixes the relative positions of the joints controller the legs' motion. Thus when one leg moves, reaction forces "oppose" its movement from two sources – firstly the ground via the leg's feet, and secondly the ground via the *other* leg's feet, transmitted through the leg and inter-leg linkage. For the purposes of this exploration, several assumptions about the structure of this linkage will be made including:

- Symmetry: the linkage is shaped such that it lies in the plane defined by the axis unit vectors of the two legs. This ensures that all forces in the leg

system act in concert, and that no power is wasted unnecessarily. Despite this, the effect of a non-symmetric linkage between the legs could provide interesting insights, but would be best investigated in another study

- Height: the local vertical component of the linkage's dimensions will be such that, when added to the shorter of the leg's local vertical dimensions, it is equal to the vertical dimension of the longer leg. Or, phrased differently, the local vertical height of the legs and linkage will be determined such that the local vertical vectors are parallel to the *global* vertical vector when the robot is in square stance.

As there are quite a large number of variables that would be possible to analyse or measure, in order to form exploratory experiments feasible within the scope of this thesis, the following variables were chosen as control, driving, or output variables.

Controlled variables are those that will be altered in the course of an experimental run, to observe the difference in the behaviour of output variables from consecutive executions of the experiment. Variables chosen as influencing factors to consider in this manner include:

- h-leg width
- h-leg height (and thus the height of the linkage)
- v-leg angles
- h-leg angles

The width and height of the v-leg will not be considered in this manner, but rather set to a "unit" dimension, and the experiments conducted considering primarily the ratio between respective v-leg and h-leg dimensions. Some trials will be performed to determine the validity of this approach, and it is hoped the ratio may exhibit linear correlation to a measured output characteristic.

Driving variables are effectively the input to the transfer function defined by the control variables, and produce an output that will (presumably) be different depending on the

exact value of the control variables. Variables chosen to "drive" experiments in this fashion include:

- h-leg angle
- v-leg angle

Output variables are those that will be measured in order to quantify the robot's behaviour; they describe how the system moves in response to the changes in control and incidental variables. These variables include:

- COM position (for simplicity's sake, measured as the centre of mass of the linkage, without taking the position of the legs into account)
- H-leg angle
- V-leg joint power
- H-leg joint power

4.3 Research Questions

In the context of the variables and other considerations outlined above, a series of questions were developed, summarised in Table 3. These will then be used to develop suitable simulations, needed to be answered, in order to provide an overall picture of the mechanical factors influencing the Nv-bot's gait.

ID	Research Question Summary
1	Are there any identifiable features in the robot's simulated behaviour that could be used to extract parameters to define the walking motion?
2	If such features are found, and parameters extracted, do these parameters lead to successful walking?
3	Are these parameters optimal?
4	Are walking parameters primarily affected by changes in horizontal or vertical leg dimension?
5	Are walking parameters primarily influenced by relative or absolute geometry factors?

6	What influence do surface properties (particularly, static & dynamic coefficients of friction) have on the optimality of the extracted parameters?

Table 3: Summary of Research Questions

4.4 Simulation Runs

In order to satisfy the research questions outlined in the previous section, several simulation runs were developed to test the influence of various factors on the mechanical context's walking behaviour. Each of these experiments takes a series of inputs, applies a methodology to those inputs across multiple simulations, and produces a series of outputs, which may be used in further simulations or simply as results for consideration and analysis.

4.4.1 Alpha

Answering Question 1: Are there any identifiable features in the robot's simulated behaviour that could be used to extract parameters to define the walking motion?

Inputs	Alpha Simulation Run	Outputs
Fixed Variables: • V-leg Width • V-leg Height • H-leg Height	Controlled Variables: • H-leg Width Driven Variables: • V-leg Rotation Measured Variables • H-leg Rotation	Produced Results: • Graph: V-leg Angle versus H-leg Angle

Table 4: Alpha Run Variables and Outputs

The intent of the Alpha run simulations is to produce data that can be used in the development of viable walking gaits for the robot. Thus it does not seek to test anything

per sé, but rather to provide a body of data from which useful information and observations regarding the robot's mechanical behaviour can be derived. In particular, it is hoped that clearly defined features will be apparent in the graphs produced from this simulation run, that can then be used as reference points to understand the walking behaviour of the robot, and calculate appropriate gaits.

To this end, the Alpha run simulations will drive the v-leg joint in one direction, starting from square stance; the h-leg joint will be left free, and the behaviour of the h-leg in response to the v-leg's movement will be observed.

Experiences gained from the pilot run (section 3.3 (Pilot Run)) suggested that the relationship between the horizontal- and vertical-stepping legs held promise for answering this question. Although each leg certainly exerts a force on the other. Each movement of each leg influences the other, however the geometrical arrangement of the robot is such that no force introduced to the v-leg by the action of the h-leg is capable of altering the position of the v-leg, even if the v-leg joint is free to move.

Because the v-leg's axis of rotation is parallel to the ground plate, and the v-leg is symmetrical around this axis, the stable position of the leg is to rest in "square" stance – i.e. with both feet on the ground. Any forces introduced by the h-leg will be along the plane of the ground plate, and thus while it could potentially displace the v-leg (and indeed, this is necessary for walking to occur), it is incapable of applying a moment to the v-leg's joint. In order to do this, uneven forces would need to be exerted on each foot, with non-zero components to the force that were orthogonal to the axis of rotation. This is an impossible situation, and thus it is not necessary to run a trial with a driven h-leg, and free v-leg.

The v-leg's movement, however, has the capability to significantly affect the status of an unrestrained h-leg. It stands to reason that, as the v-leg turns, one of two things must happen to accommodate the movement of the robot. The first possibility is that as one of

the v-leg feet rises off the ground, the other slips – this will result in the vertical motion of the v-leg joint, and little other direct impact on the robot's stance.

The second possibility is that, as the v-leg joint turns, the h-leg joint rotates sympathetically, skewing the robot such that the v-leg foot still on the ground does not move. Interestingly, both of these cases are equivalent; they each result in the same final robot stance.

Thus, while there is little possibility for the h-leg's movement to influence the position of the v-leg, there exists significant potential for the v-leg's movement to cause motion of the h-leg – determining the extent and nature of this influence is expected to be critical in determining the mechanical conditions of a proficient walking gait for this robot.

The controlled variables in this simulation run are the widths of the h- and v-legs. The width of the v-leg will be held constant, and the h-leg's width will be varied relative to this. In the Delta simulation run, both the v-leg width and h-leg width will be varied with similar driving and free variable conditions, which will demonstrate whether these Alpha run results may be analysed from a relative perspective, or if they must be considered in absolute terms.

The principle outputs of this simulation run will be any identified behavioural features identified in the simulation executions, and the conjectured walking gait parameters derived from any such features. The features will be discussed in the results section, and the derived parameters will be used in "downstream" simulation runs. Additionally, other simulation runs, particular Gamma through Epsilon, will use this simulation as a reference point to compare the effect of controlling other variables involved in the simulation.

4.4.2 Beta

Answering Question 2: If such features are found, and parameters extracted, do these parameters lead to successful walking?

and

Question 3: Are these parameters optimal?

Inputs	Beta Simulation Run	Outputs
Fixed Variables: • V-leg Width • V-leg Height • H-leg Height Upstream Variables • V-leg Rotation • H-leg Rotation	Controlled Variables: • H-leg Width Driven Variables: • V-leg Rotation • H-leg Rotation	Measured Variables • COM Displacement • Power consumed Produced Results: • Graphs: ○ Centre of Mass Path (X, Y, Z) • Metrics: ○ Peak Power Dissipated per Step

Table 5: Beta Run Variables and Outputs

The purpose of the beta simulation run is twofold - first, to establish whether the results obtained from the alpha run simulations lead to a successful walking gait, and second, to establish whether this gait is the "most proficient" possible from that configuration.

In the pursuit of both of these goals, the results from the alpha run simulations will be analysed to determine any prominent features or relationships between the motions of the two legs. Experience gathered during the pilot run simulations suggest that, for many values of h-leg width w_h, that rotating the v-leg joint will actually result in v-stepping of the h-leg. It is hoped that plots of h-leg angle against v-leg angle will display significant features, or at least a very strong relationship, to allow useful data points to be extracted for gait isolation.

The primary parameters of concern are the angles suggested by the results to describe the optimum gait for that leg topology. This can be expressed as a four dimensional parameter vector,

$$p_{w_h} = \begin{Bmatrix} w_v \\ w_h \\ |\theta_v| \\ |\theta_h| \end{Bmatrix} \begin{Bmatrix} mm \\ mm \\ \circ \\ \circ \end{Bmatrix}$$

For example, at $w_h = 150mm$:

$$p_{w_h=150} = \begin{Bmatrix} 100 \\ 150 \\ 8.617 \\ 28.340 \end{Bmatrix}$$

Equation 3: Beta Run Parameter Vector Construction

Once this parameter set has been developed for a particular leg geometry, a driving function must be created to define a gait using the specified joint angles.

To implement this driving function, it was necessary to use an "expression" type function within COSMOS/Motion's joint editor. This allowed for the definition of a mathematical composite of the "simple" function types (such as STEP, HARMONIC, CONSTANT, etc...), and thus for the creation of a piecewise-nonlinear equation. The parameters from the vector (Equation 3 above) are distributed across the driver function expression as shown below in Equation 4:

$$\theta_v = \text{"STEP}(\text{TIME},0,0\text{D},2,|\theta_{v_{max}}|) + \text{STEP}(\text{TIME},5,0\text{D},7,-2\cdot|\theta_{v_{max}}|) + \text{STEP}(\text{TIME},10,0\text{D},12,2\cdot|\theta_{v_{max}}|) + \text{STEP}(\text{TIME},15,0\text{D},17,-2\cdot|\theta_{v_{max}}|)\text{"}$$

$$\theta_h = \text{"STEP}(\text{TIME},2.5,0\text{D},4.5,-|\theta_{h_{max}}|) + \text{STEP}(\text{TIME},7.5,0\text{D},9.5,2\cdot|\theta_{h_{max}}|) + \text{STEP}(\text{TIME},12.5,0\text{D},14.5,-2\cdot|\theta_{h_{max}}|) + \text{STEP}(\text{TIME},17.5,0\text{D},19.5,2\cdot|\theta_{h_{max}}|)\text{"}$$

Equation 4: Vertical & Horizontal Leg Driver Functions for Beta Simulation Run

These driving expressions cover 2.5 steps (i.e. first "half" step from square stance, followed by two full steps), and are constructed via an automated excel spreadsheet developed by the author, in order to minimise errors and allow for rapid turnover between simulations. The output expressions are copied into the relevant joints in the simulated model, and the simulation executed.

Upon completion of a simulation, the first purpose of this run can be fulfilled by mere observation: did the robot successfully move due to the implemented gait parameters? Were there any particularly odd movement artifacts to suggest non-viable parameters? Despite the "subjective" nature of this test, the human eye is capable of judging the proficiency of walking gaits with surprising facility, likely in part due to the observation of many different such gaits of people and creatures around us everyday (Tilden, 2006).

For the second purpose of this simulation run, however, more objective evidence needs to be collected. In particular, to determine the proficiency of the gait, the behaviour of two major variables will be considered - the movement of the robot's COM, and the power expended by the leg joints over the course of the gait. The 3D path traced by the robot's COM provides several useful insights - how "bumpy" the gait is (described by Y-axis movement, and is expected to be similar for most gaits), the "stride benefit", or forward distance gained per stride, and the "stride loss", or the sideways deviation from a perfectly straight movement. For the purposes of this thesis, the stride benefit and loss will not be considered separately, but rather be combined (via the Pythagorean theorem) to provide the stride length, or total displacement per stride of the COM's projection on the ground plane.

Ideally, the power expended throughout the gait would have been represented by calculating the work done by each joint over the gaits' operation; unfortunately, however, trial runs showed that the results returned (Instantaneous Power against Time) by the simulation engine were, for the most part, far too noisy and/or distorted for meaningful Work calculations to be done.

Instead, peak power consumed by the joints in each of their respective steps will be reported - this gives an idea of the "effort" the robot must expend to complete each step, as the shape of the power trace across each step is regular. For further study, a more accurate simulation engine is recommended so that work done can be studied directly; if this is not available, a regression analysis on a non-distorted section of data could potentially yield a generalisable equation for the shape of the power trace across a step. This could then be applied across noisier sections of data, as the critical dimensions (x-axis width and y-axis height) of the trace can be measured or inferred in almost all cases. By integrating this "shape function", with the relevant parameters extracted from the step in question, an estimate of the work done per step could be produced; the time required for a suitable regression analysis unfortunately puts this prospect beyond the acceptable time-scope for this project.

In order to provide a context within which these "proficiency indicators" can be rated, the beta simulation for a single parameter vector will consist of five runs - one exactly as described in the vector, and simulations at 5% and 10% both above and below the h-leg angle defined in the vector (as well as the corresponding v-leg angle corresponding to this from the alpha run results). The h-leg's behaviour is expected to be a major determinant of a gait's performance, and by considering a range of values around the conjectured "optimum point" identified in the alpha simulation, this will provide an indication of how predictive the originating features of the parameter vector is of stride optimality.

4.4.3 Gamma

Answering Question 4: Are walking parameters primarily affected by changes in horizontal or vertical leg dimension?

Inputs	Gamma Simulation Run	Outputs
Fixed Variables: • V-leg Width • V-leg Height	Controlled Variables: • H-leg Width • H-leg Height Driven Variables: • V-leg Rotation	Measured Variables • H-leg Rotation Produced Results: • Graphs: ○ V-leg Angle versus H-leg Angle

Figure 30: Gamma Run Variables & Outputs

The gamma simulation run is intended to indicate the likely effect of h-leg height (relative to v-leg height) on the walking behaviour of the robot's geometry.

In order to observe this relationship, a simulation format similar to that of run alpha will be used. In this case, however, only a small number of executions will be made. Several h-widths will be chosen, across the breadth of the range tested in the alpha run, and at each of these four h-widths, three different h-heights will be test → 60%, 90%, and 110% of the v-leg height (the original Alpha trial providing a fourth dataset). These trials will be undertaken in exactly the same way as for run alpha, with the v-leg driven through a suitable "STEP" function, and the h-leg remain free-jointed.

A sub simulation run, similar to the beta run, will then be undertaken; should a significant relationship between h-leg height (and thus extracted walking parameters) and the walking behaviour of the mechanical context exist, this will become evident in the results produced by this trial. A comparison of the relative influences of relative h-leg height and width will then be possible, thus aiding future robotic designs by highlighting the criticality of these features.

4.4.4 Delta

Answering Question 5: Are walking parameters primarily influenced by relative or absolute geometry factors?

Inputs	Delta Simulation Run	Outputs
Fixed Variables: • H-leg Height • V-leg Height	Controlled Variables: • H-leg Width • V-leg Width Driven Variables: • V-leg Rotation	Measured Variables • H-leg Rotation Produced Results: • Graphs: ○ V-leg Angle versus H-leg Angle

Figure 31: Delta Run Variables & Outputs

In order to answer this question, a simulation similar to that used in the alpha run will be employed, in which there are two controlled variables – the widths of the horizontal- and vertical-stepping legs. Four v-leg widths will be chosen, and for each of these, simulations will be run at any relative h-leg widths which are shown to be significant after the completion of the alpha run simulations. The alpha run simulations themselves will also contribute a dataset, thus meaning the delta run will effectively examine five different v-leg widths.

It is thought that, if each of these trials (i.e. per v-leg width) exhibit similar behaviours, then the mechanical context's walking behaviour is primarily influenced by the relative widths of the legs. The reverse to this – that walking behaviour is determined by the absolute widths of the legs – will be considered true if the various v-leg widths demonstrated significantly different behaviours at the same relative h-leg widths.

Should the latter be the case, and the behaviour of the mechanical context when applied to walking be determined by the absolute widths of the h-leg, this would suggest that the horizontal-stepping leg is the primary contributor to walking behaviour.

4.4.5 Epsilon

Answering Question 6: What influence do surface properties (particularly, static & dynamic coefficients of friction) have on the optimality of extracted parameters?

Inputs	Epsilon Simulation Run	Outputs
Fixed Variables: • H-leg Height • V-leg Height • V-leg Width	Controlled Variables: • H-leg Width • Static friction coeff. • Dynamic friction coeff. Driven Variables: • V-leg Rotation	Measured Variables • H-leg Rotation Produced Results: • Graphs: ○ V-leg Angle versus H-leg Angle

Figure 32: Epsilon Run Variables & Outputs

A major component of the simulations that will not be analysed in any of the aforementioned simulation runs is the friction properties of the ground plane. As mentioned in section 3.4 (Environment), the friction characteristics of the ground surface were deliberately exaggerated in the development of the simulation, in order to accentuate the results and allow for easy identification of features. However, as a major component of the robot's movement is dependant on the interplay between static & dynamic friction forces (due to the necessity of friction between the "feet" and the ground for the feet to move the robot), it is necessary to at least briefly explore the implications of the friction coefficients on a walking gait's proficiency.

In order to do this, the Epsilon run simulation will be modeled on the Alpha run, and thus use a fixed v-leg width, with a varied h-leg width to provide context. The primary controlled variables will be the static and dynamic coefficients of friction, however instead of being expressed as two absolute values, these two coefficients will be described as an absolute value (for the dynamic coefficient of friction) and a delta value (for the static coefficient of friction, relative to the dynamic coefficient).

This simulation run will be run at each of five h-leg widths (95%, 100%, 150%, 200% and 205% of the v-leg width), and nine executions will be run at each h-leg width. The parameter values for each of these executions is shown below, relative to a base value "A":

Execution	Dynamic Coefficient	Static Coefficient
1	A	A
2	A	A*105%
3	A	A*110%
4	2A	2A
5	2A	(2A)*105%
6	2A	(2A)*110%
7	3A	3A
8	3A	(3A)*105%
9	3A	(3A)*110%

Table 6: Friction Coefficients for Executions of Epsilon run Simulations

If enough time is available, an additional round of executions will be performed with the static coefficient *less* than the dynamic coefficient of friction – although this situation is rarely encountered in nature, the results of these simulations may prove insightful into robot behaviour in environments exhibiting "unusual" or "unexpected" physics.

The output data from this simulation run will be compared with the Alpha run results, thus providing an indication of the influence of friction on the robot's walking behaviour.

This data will not be used in any further simulation, unless features of significant import are observed in the robot's behaviour; rather, these results will be purely used for discussion purposes.

4.5 Selection of Driver Functions

Two major options were considered when attempting to determine the best type of driver to power the simulation: displacement or force drivers.

Force drivers were the means used to power the initial "pilot run" simulation run. They were chosen as they allowed for the interaction of forces external to the leg's driving joint to influence the motion of the leg. This meant that, as one leg turned, it could influence the other leg, and vice versa .

In the author's initial thinking and hypotheses for how the Nv-bots walking gait might work, it was thought this interaction between the legs during the walking cycle would be a necessary factor, and that by being able to precisely control the force supplied, it would possible to exactly model the motor used in the real-world robot.

The latter of those points was one of the strongest arguments for the usage of force drivers, as the more accurate the simulated model was to the author's real-world robot, the easier it would be to implement the results, and provide more time for the debugging of the real-world robot. However, this could also be considered a significant disadvantage, as, in order to accurately represent the motor in COSMOS/Motion, it would be necessary to somehow determine the motor's characteristics – an activity which was beyond the capability of the equipment the author had access to.

Additionally, being able to precisely determine the force applied to the motors over time could be considered a mixed blessing – this approach would vastly increase the amount of mathematical modeling required to manually determine the correct force/time functions to turn the motors to the correct angles, in both simulation and reality

Displacement drivers are simpler to implement, as they are encoded as a motion directly in the joint definition. No additional maths is required to model exactly how far the joint will turn, as this is defined in the function parameters. Additionally, a displacement driver will deliver whatever power is necessary to successfully turn the joint to the required angle, reducing the need for external mathematical modeling. These results can then be used to assist in motor selection for the construction of real-world Nv-net based walker robots.

The major disadvantage of displacement drivers, however, is exactly that – they will supply however much power is needed to rotate the joint to the specified angle. Because of this, displacement-driven joints are "locked" at the angle determined by their driving function, and are not free to respond to forces introduced by, for example, the motion of the other leg.

This is not entirely unrealistic, however, as in the highly geared motors often used in the construction of Nv-walkers (such as modified hobby servos, as outlined in section 2.3 (Actuators)), large amounts of power are required to drive the gear train from the output back to the motor, anyway. The resultant effect of this is that, if a real-world robot's joints are twisted into a certain pose, it is likely to stay that way – even with no power being supplied to the motors supporting the joints in question.

The two deciding factors in the choice of the driver type were thus the mathematical overhead, and the "realism" of the simulation's treatment of the drivers relative to motors in the real world. Displacement drivers significantly decreased the mathematical overhead of the simulations, as it was not necessary for the author to manually calculate the necessary parameters to produce a desired step from the leg, so this was a major point in their favour.

However, it was decided that the "realism" of displacement drivers as analogous to highly geared motors took precedence over the "realism" of force-drivers as representing

the "commands" sent to the motors from the driver circuits. The primary reason for this was that the scope of these simulations was to consider the geometry and mechanical aspects of the design, and not consider the controller and interface circuitry.

This would allow for the mechanical factors to be considered in isolation, providing a platform for future work to explore the necessary factors to be considered in interface and Nv-controller design to service the requirements of the mechanical geometry.

Thus, displacement drivers were chosen, with one caveat – because the movement from one joint would no longer affect another driven joint, it would be necessary to do exploratory simulations of joint behaviour separately on each joint, with the other joint free to rotate. Without this, it would be difficult to isolate the critical factors in any such experiments.

4.6 Summary

In order to develop results for this thesis in the absence of an operational physical model, a 3D CAD model of the mechanical context was developed in order to simulate the operation of the robot. This was developed in COSMOS/Motion, an add-on for the Solidworks CAD package, which allows the development of a complete physical model, including friction, gravity, and custom-defined driving functions. Thus it was possible to not only statically model the mechanics of the robot, but also to dynamically study a simulation of the mechanical context's walking behaviour, and define an "idealised" world in which the simulated robot can operate.

A longstanding theme in many fields of research has been the conflict between simulation & experiments implemented in the "real world".

Simulations have significant advantages in terms of implementation time and cost, however perhaps their biggest benefit is in the area of simplicity. By undertaking a simulation as opposed to performing experiments in the real world, one is a) able to

precisely control many factors that could influence the outcome of the study, and b) operate in an "idealised" version of the real world.

The downside to this, however, is that errors are often introduced into the simulation due to these "simplifications of reality". With respect to this project, the largest simplification present is the assumption of a perfectly smooth walking surface. A walking robot in the real world will likely encounter widely varied terrains and surfaces, and thus results derived purely from simulation on a flat surface will not necessarily be entirely applicable to such situations.

In this context however, useful results can be obtained and conclusions drawn, provided they are qualified as requiring real world validation. To this end, this project planned to undertake several rounds of simulation, each designed to determine some aspect of the robot's behaviour. The simulation runs are summarised in Table 7 following:

Sim Run	Question ID	Summary
Alpha	1	To determine the effect of h-leg width (relative to v-leg width) on robot behaviour, and to investigate the presence of any identifiable features for developing operating parameters for robotic walking.
Beta	2,3	To determine the "optimality" of parameters developed from Alpha trials, by applying them to a walking robot simulation.
Gamma	4	To determine the influence of the h-leg's height over the robot's walking behaviour, and compare its influence to that of h-leg width as tested in the Alpha run.
Delta	5	To explore whether robot walking behaviour is primarily influenced by absolute or relative leg dimensions
Epsilon	6	To determine the influence of both static and dynamic friction coefficients on the simulated robot's walking behaviour.
		Note that each of Gamma, Delta and Epsilon runs will require the conductance of a Beta-type simulation run subsequent to their completion, in order to evaluate the walking behaviours produced by their developed

		parameters.

Table 7: Summary of Simulation Runs

While this project is heavily dependant on simulation-derived results, the author is aware of the implications and limitations of these simulations for the derivation of research results useful in the "real world". For this reason, it is strongly advocated in section 6.2 (Recommendations) of this thesis that a real world study be undertaken that mirrors the simulations undertaken herein. This will provide a point of reference for the simulated data, and help point out particular areas in which the simulated results are inadequate.

5 Results

5.1 Introduction

After each simulation execution was carried out, pertinent data was exported from COSMOS/Motion into Microsoft Excel, a spreadsheeting program, in the form of both graphs and the accompanying raw data. This section will detail the most relevant data for each of the simulation runs, highlighting any unusual aspects of particular executions, before proceeding to explore and discuss the implications and conclusions that are attendant to these results.

5.2 Alpha Run

5.2.1 Variations

The only major surprise in the execution of the Alpha run simulation was that the critical point phenomena, observed as the "elbow" on the angle plots, was not present in simulation runs with W_h>200 mm, or W_h<100 mm. Insights from completion of the Delta run simulations will suggest whether this is directly related to the width of the vertical leg (W_v, which here is equal to 100 mm), however additional simulations undertaken outside this 100 mm to 200 mm range either experienced collisions between leg components, or rapidly progressed into a "runaway" collapse of the geometry. In the results below, this is exhibited as Geometry Alpha-1.

5.2.2 Results

The graph trace labeled "Revolute" refers to the rotation of the vertical-stepping leg joint; "Revolute2" refers to the horizontal-stepping leg joint.

5.2.2.1 Geometry Alpha-1: Wh = 90 mm

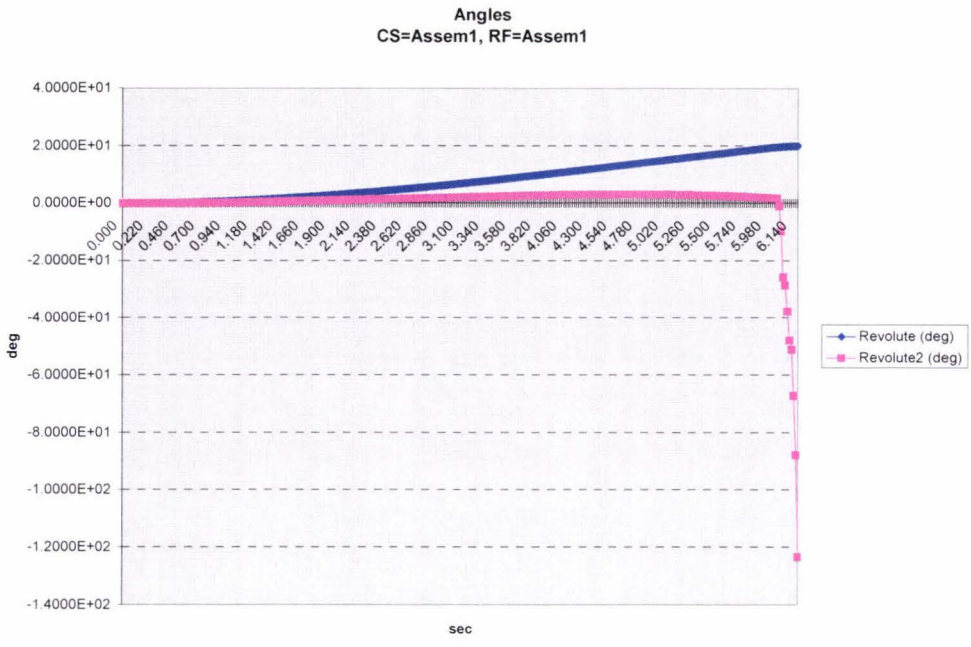

Figure 33: Angle Relationship, Alpha Run, Geometry Alpha-1, Wh=90 mm

5.2.2.2 Geometry Alpha-2: Wh = 100 mm

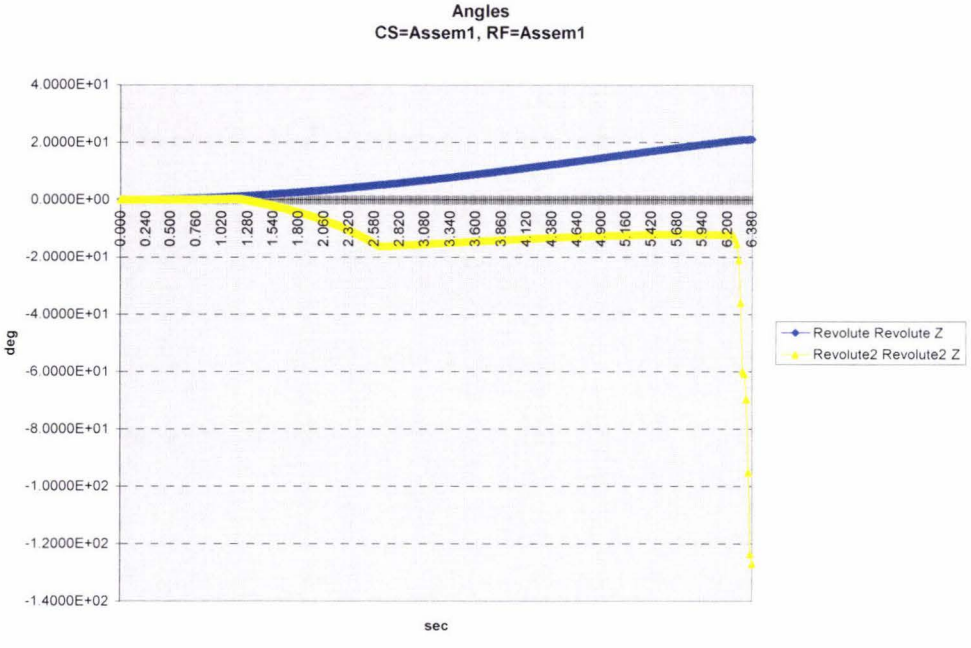

Figure 34: Angle Relationship, Alpha Run, Geometry Alpha-2, Wh = 100 mm

5.2.2.3 Geometry Alpha-3: Wh = 110 mm

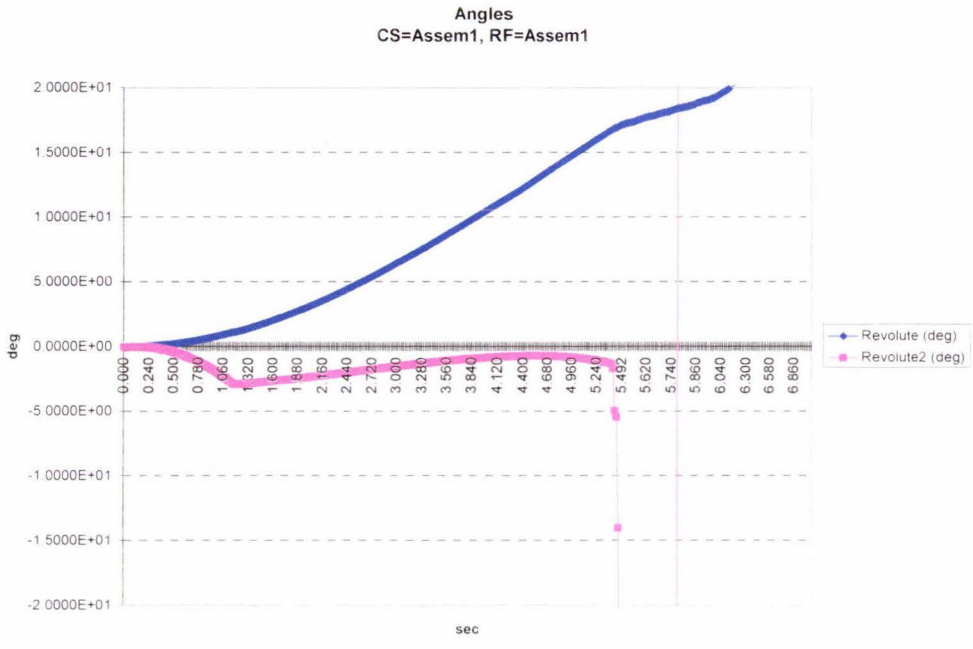

Figure 35: Angle Relationship, Alpha Run, Geometry Alpha-3, Wh=110 mm

5.2.2.4 Geometry Alpha-4: Wh = 120 mm

Figure 36: Angle Relationship, Alpha Run, Geometry Alpha-4, Wh=120 mm

5.2.2.5 Geometry Alpha-5: Wh = 130 mm

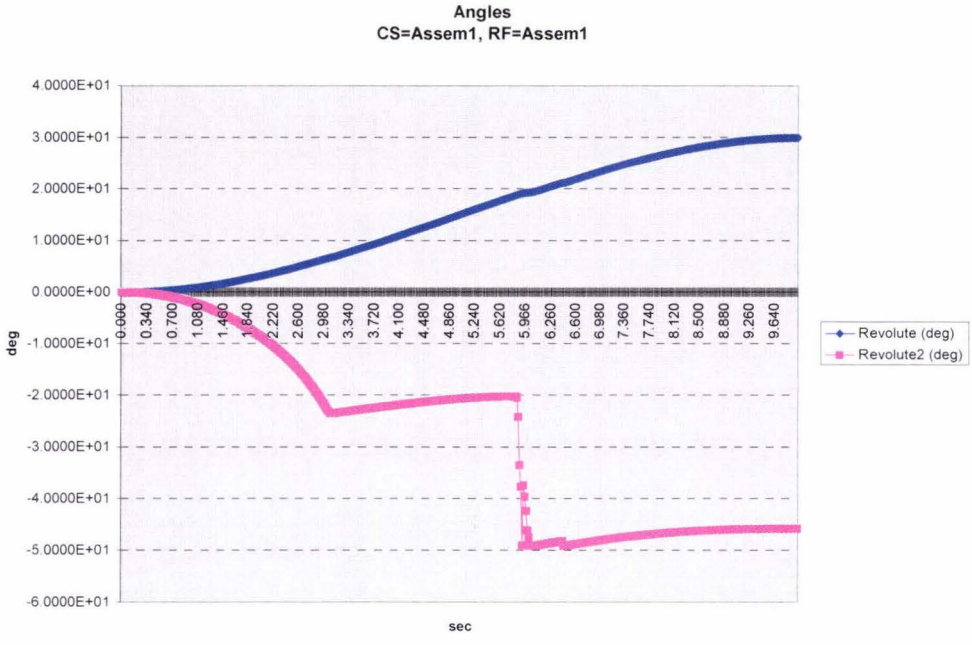

Figure 37: Angle Relationship, Alpha Run, Geometry Alpha-5, Wh=130 mm

5.2.2.6 Geometry Alpha-6: Wh = 140 mm

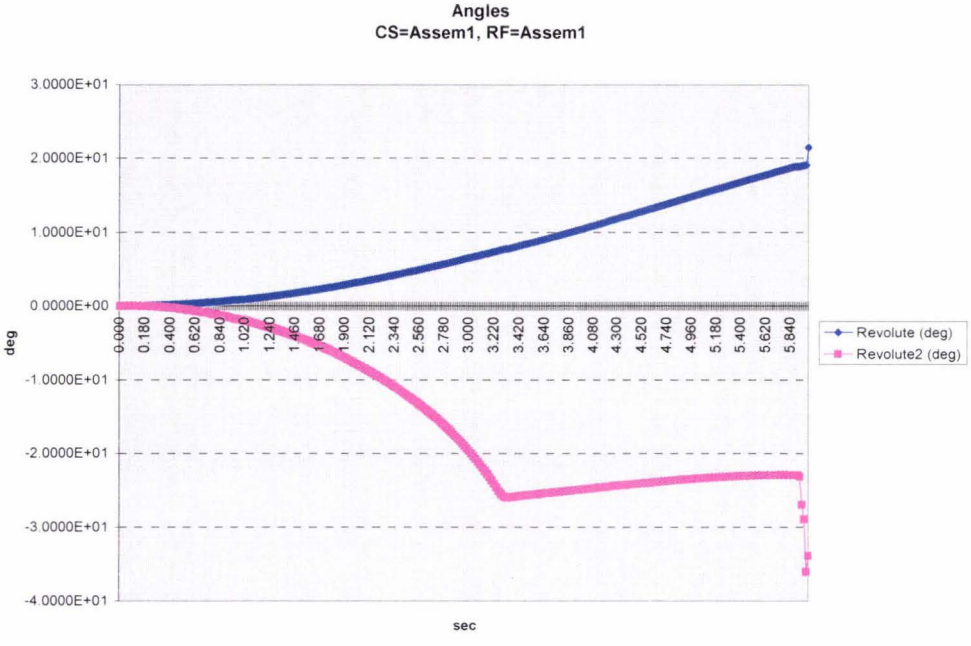

Figure 38: Angle Relationship, Alpha Run, Geometry Alpha-6, Wh=140 mm

5.2.2.7 Geometry Alpha-7: Wh = 150 mm

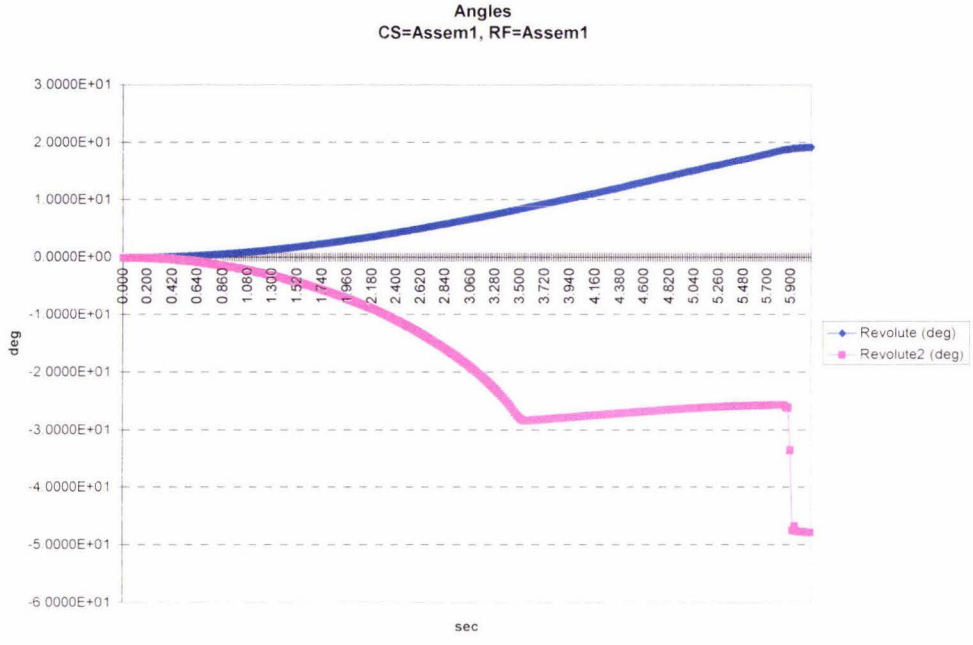

Figure 39: Angle Relationship, Alpha Run, Geometry Alpha-7, Wh=150 mm

5.2.2.8 Geometry Alpha-8: Wh = 160 mm

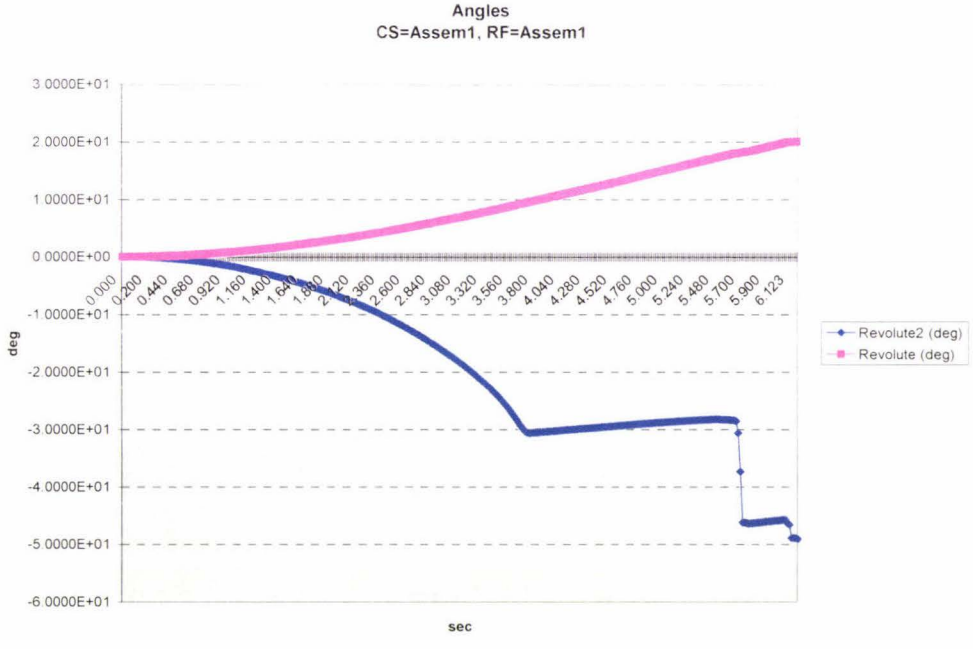

Figure 40: Angle Relationship, Alpha Run, Geometry Alpha-8, Wh=160 mm

5.2.2.9 Geometry Alpha-9: Wh = 170 mm

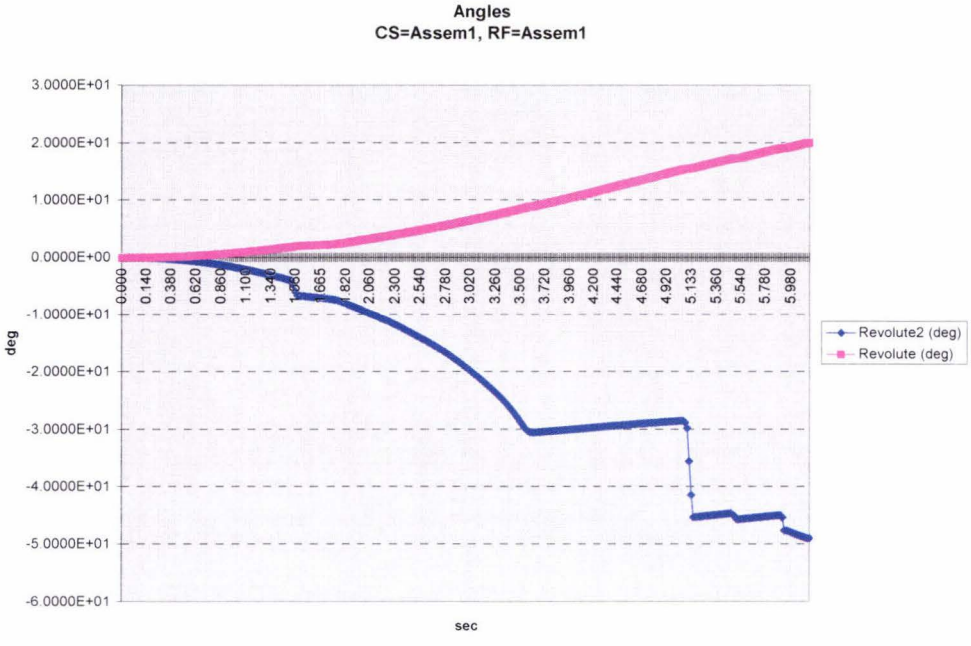

Figure 41: Angle Relationship, Alpha Run, Geometry Alpha-9, Wh=170 mm

5.2.2.10 Geometry Alpha-10: Wh = 180 mm

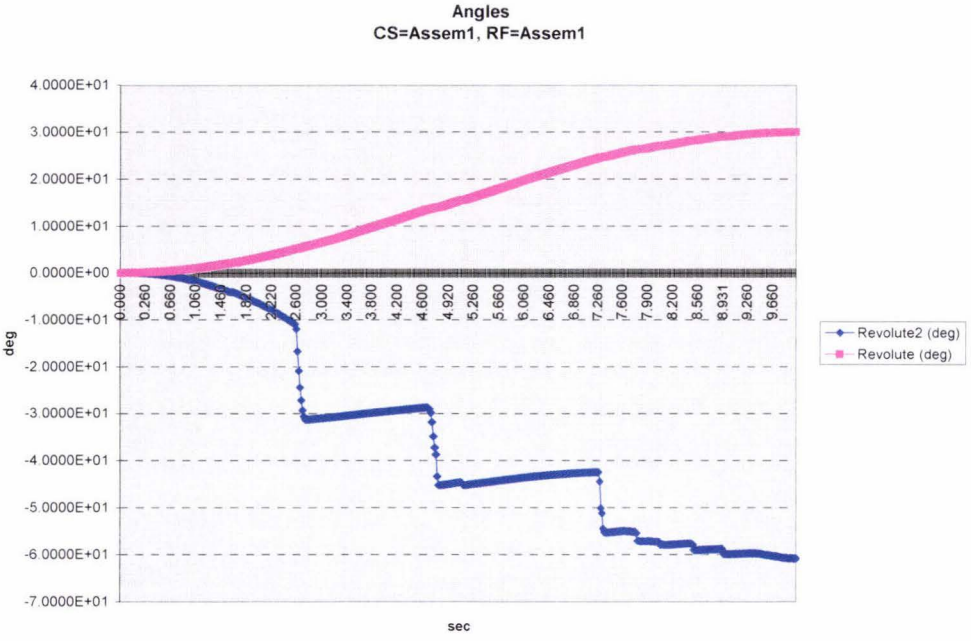

Figure 42: Angle Relationship, Alpha Run, Geometry Alpha-10, Wh=180 mm

5.2.2.11 Geometry Alpha-11: Wh = 190 mm

Figure 43: Angle Relationship, Alpha Run, Geometry Alpha-11, Wh=190 mm

5.2.2.12 Geometry Alpha-12: Wh = 200 mm

Figure 44: Angle Relationship, Alpha Run, Geometry Alpha-12, Wh=200 mm

5.3 Beta Run

5.3.1 Variations

5.3.1.1 Simulation Density

Originally, it was intended that a beta simulation would be run for each of the geometries tested in the alpha run simulations. However, after testing the first few geometries, this was deemed impractical, for several reasons.

Significantly, the time required to simulate the required 20 second time period varied anywhere between 10 minutes to several hours, although longer simulation times tended to be associated with runs that had encountered a glitch or developed a simulational singularity. However, it was deemed inadvisable to prematurely terminate such "overtime" simulations, as some such simulations did eventually resolve; also, the COSMOS/Motion engine seemed to exhibit distinct differences across repeated executions of exactly the same simulation file. In some cases, this appeared to run in a period-4 cycle, wherein every fourth execution (relative to any execution in the sequence) appeared to be identical (including any glitches).

This time requirement was compounded by the inability for the process of running a simulation to be automated in any manner. Several avenues were explored, included the scripting of a Windows-based batch file to operate COSMOS from the command line, writing a program in a dedicated scripting language to interface with the Solidworks and COSMOS/Motion APIs, and using Solidworks' own built-in automation and task-scheduling functionality. However, it did not appear that COMOS/Motion was accessible via a command line batch file, and documentation on programmed scripting for the various APIs was hard to come by and sketchy at best. While Solidworks' own macro and task scheduling functionality would have been sufficient for the job, Solidworks'

macro recording engine seemed unable to record interactions with the COSMOS/Motion add-in, rendering this option untenable also.

These factors contributed to the decision to prioritise the running of the beta simulation, ensuring that the results were no less valuable in the event of significant time-overruns of the simulations. To effect this, simulations were chosen at each end of the range of alpha-run tested geometries, and also spread throughout the range between these end-points.

In order to gain an even more accurate picture of the influence the studied design factors have on the robot's walking behaviour, it will be left to a future study to repeat the various simulations, using a far higher "resolution" – that is, each simulation should be tested using a far greater range and density of values for each of the relevant controlled variables.

5.3.1.2 Simulation Input Generation

A spreadsheet was developed to calculate the necessary driving functions and simulation points for each geometry, using the data output by the appropriate alpha run simulation.

To operate the spreadsheet, the alpha run data for the "revolute2" joint is examined by a series of conditional formatting rules that examine the deltas between subsequent data points. These highlight negative deltas in red, and positive deltas in green, allowing the "elbow point" to be clearly identified from the raw simulation data. For consistency, the A sample is taken as the point on the left hand side of the elbow, with the additional benefits that this ensures the gait is very close to the edge of the sub-critical zone, while guaranteeing the sample is not on the super-critical side of the boundary.

This "A sample" is then input into the appropriate portion of the spreadsheet, which then provides target values at 90%, 95%, 105% and 110% of the initial values. The raw data

is then referred to again, to ensure accuracy, by selecting the actual data points closest to the generated target values. Once again, the limit from t- is taken.

These values are then copied across into the appropriate areas, whereupon they are used to generate the expressions representing the necessary COSMOS/Motion driving functions for each simulation execution, which can then be copied as necessary into the simulator package.

5.3.2 Results

5.3.2.1 Geometry Beta-1: Wh = 100 mm

5.3.2.1.1 Driving Functions

		4.976836	4.714897	5.500713	5.762652
Revolute	A	B	C	D	E
v-leg angle	5.238774	4.960349	4.6875	5.522591	5.73888
time to turn	2	2	2	2	2
time gap	0.5	0.5	0.5	0.5	0.5

Equations STEP(TIME,0.0D,2.00,5.24D)+STEP(TIME,5.00.0D,7.00,-10.48D)+STEP(TIME,10.00.0D,12.00,10.48D)+STEP(TIME,15.00.0D,17.00,-10.48D)

STEP(TIME,0.0D,2.00,4.96D)+STEP(TIME,5.00.0D,7.00,-9.92D)+STEP(TIME,10.00.0D,12.00,9.92D)+STEP(TIME,15.00.0D,17.00,-9.92D)

STEP(TIME,0.0D,2.00,4.69D)+STEP(TIME,5.00.0D,7.00,-9.38D)+STEP(TIME,10.00.0D,12.00,9.38D)+STEP(TIME,15.00.0D,17.00,-9.38D)

STEP(TIME,0.0D,2.00,5.52D)+STEP(TIME,5.00.0D,7.00,-11.05D)+STEP(TIME,10.00.0D,12.00,11.05D)+STEP(TIME,15.00.0D,17.00,-11.05D)

STEP(TIME,0.0D,2.00,5.74D)+STEP(TIME,5.00.0D,7.00,-11.48D)+STEP(TIME,10.00.0D,12.00,11.48D)+STEP(TIME,15.00.0D,17.00,-11.48D)

	A	B	C	D	E
Revolute2					
h-leg angle	-16.1694	-14.886	-13.3344	-16.0237	-15.9136
time to turn	2	2	2	2	2
time gap	0.5	0.5	0.5	0.5	0.5

Equations STEP(TIME,2.50,0D,4.50,-16.17D)+STEP(TIME,7.50,0D,9.50,32.34D)+STEP(TIME,12.50,0D,14.50,-32.34D)+STEP(TIME,17.50,0D,19.50,32.34D)

STEP(TIME,2.50,0D,4.50,-14.89D)+STEP(TIME,7.50,0D,9.50,29.77D)+STEP(TIME,12.50,0D,14.50,-29.77D)+STEP(TIME,17.50,0D,19.50,29.77D)

STEP(TIME,2.50,0D,4.50,-13.33D)+STEP(TIME,7.50,0D,9.50,26.67D)+STEP(TIME,12.50,0D,14.50,-26.67D)+STEP(TIME,17.50,0D,19.50,26.67D)

STEP(TIME,2.50,0D,4.50,-16.02D)+STEP(TIME,7.50,0D,9.50,32.05D)+STEP(TIME,12.50,0D,14.50,-32.05D)+STEP(TIME,17.50,0D,19.50,32.05D)

STEP(TIME,2.50,0D,4.50,-15.91D)+STEP(TIME,7.50,0D,9.50,31.83D)+STEP(TIME,12.50,0D,14.50,-31.83D)+STEP(TIME,17.50,0D,19.50,31.

Table 8: Driving Functions, Beta Run, Geometry Beta-1, Wh = 100 mm

5.3.2.1.2 COM Paths

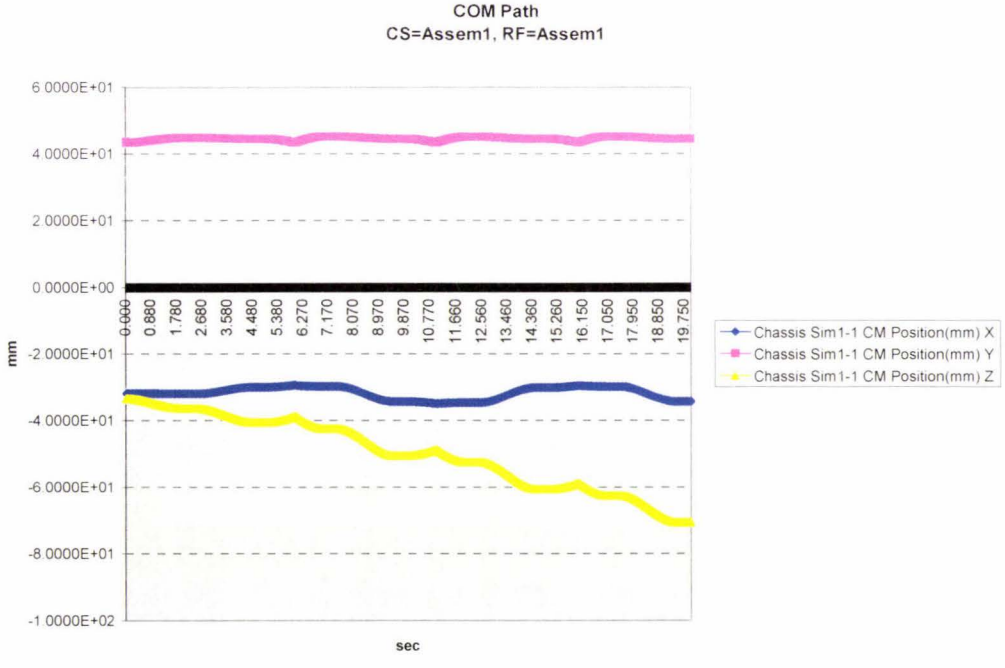

Figure 45: COM Path, Beta Run, Geometry Beta-1, Sample C

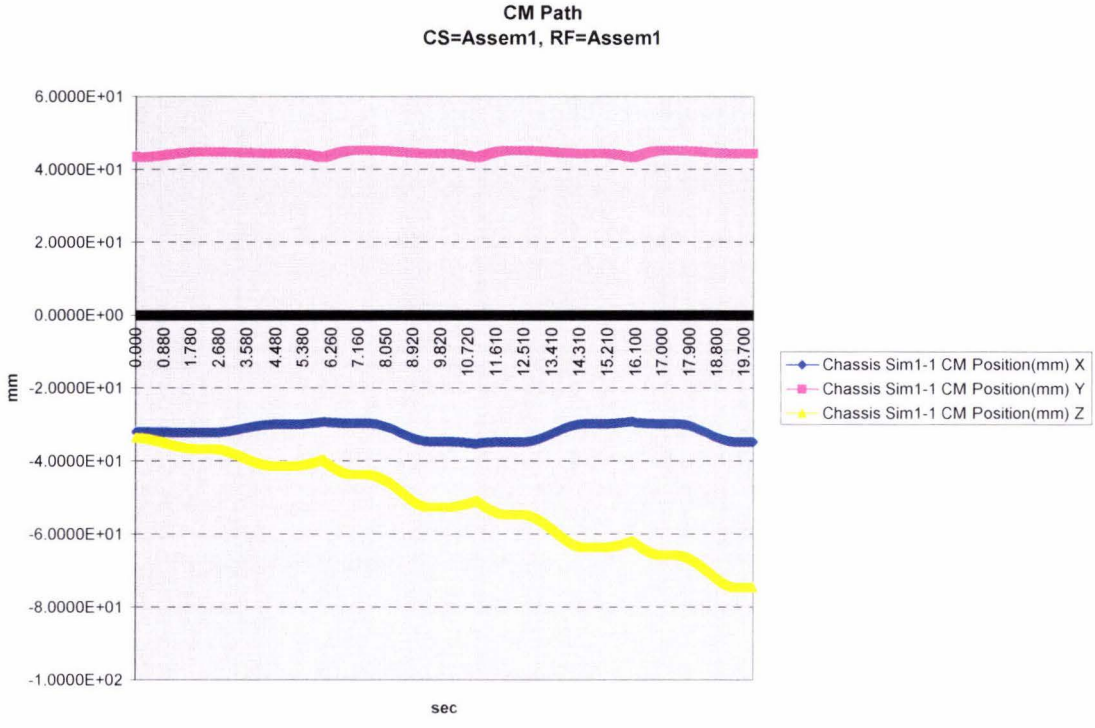

Figure 46: COM Path, Beta Run, Geometry Beta-1, Sample B

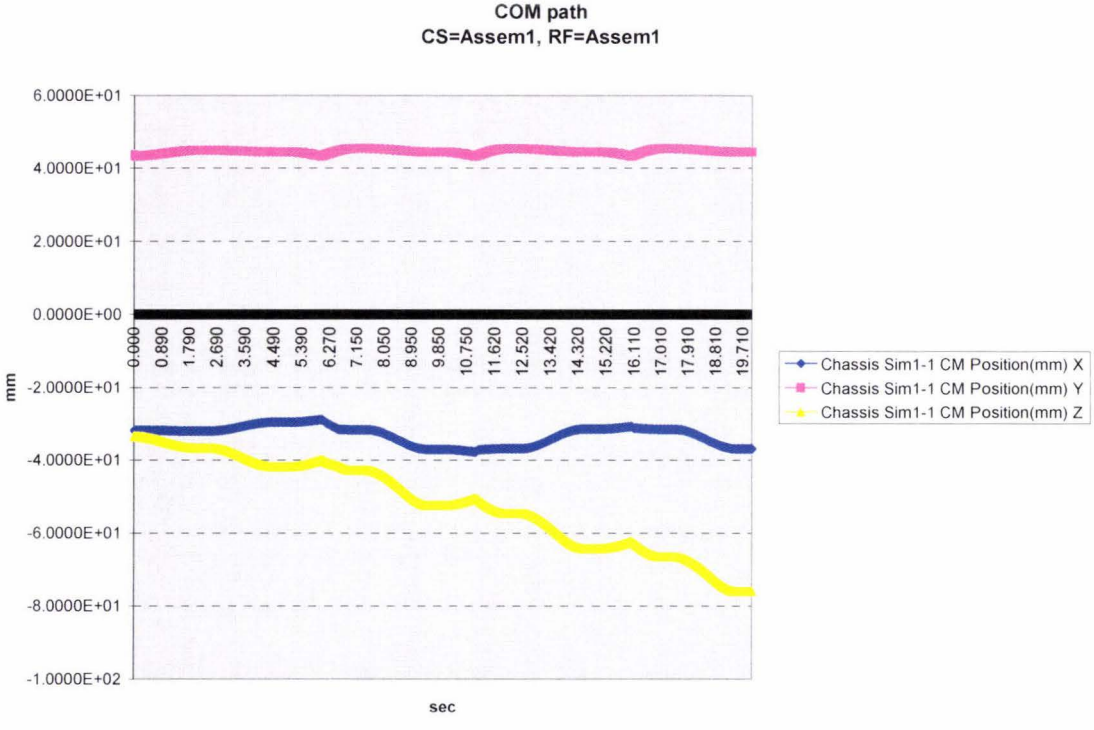

Figure 47: COM Path, Beta Run, Geometry Beta-1, Sample A

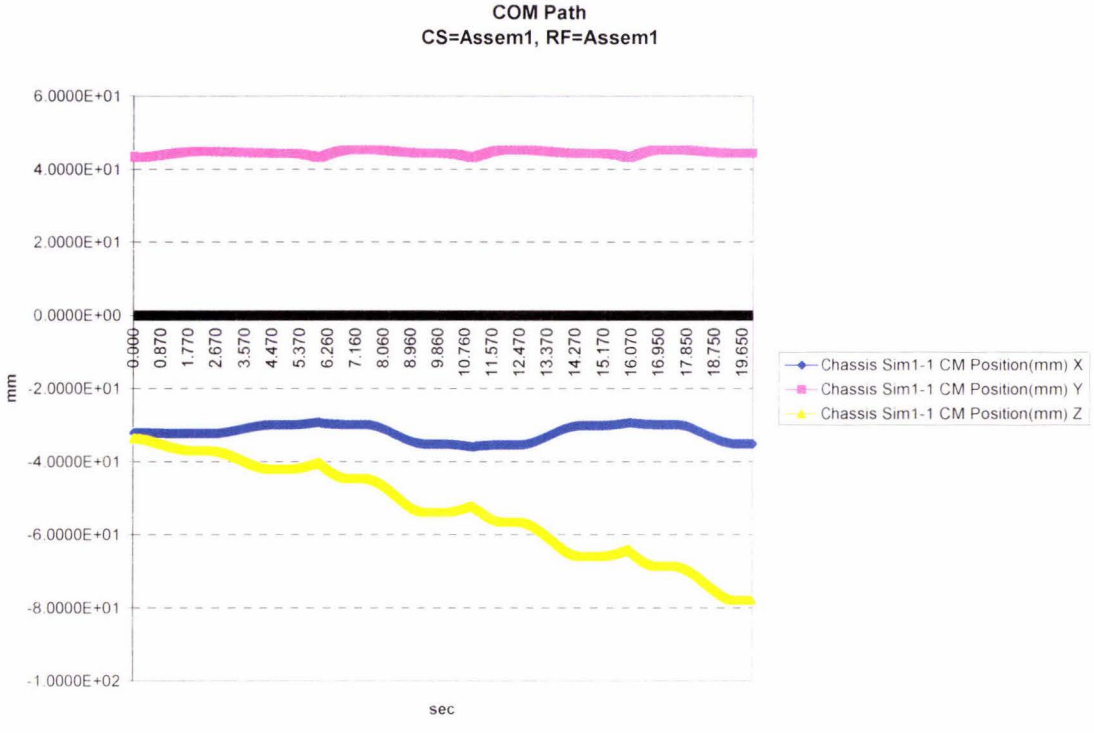

Figure 48: COM Path, Beta Run, Geometry Beta-1, Sample D

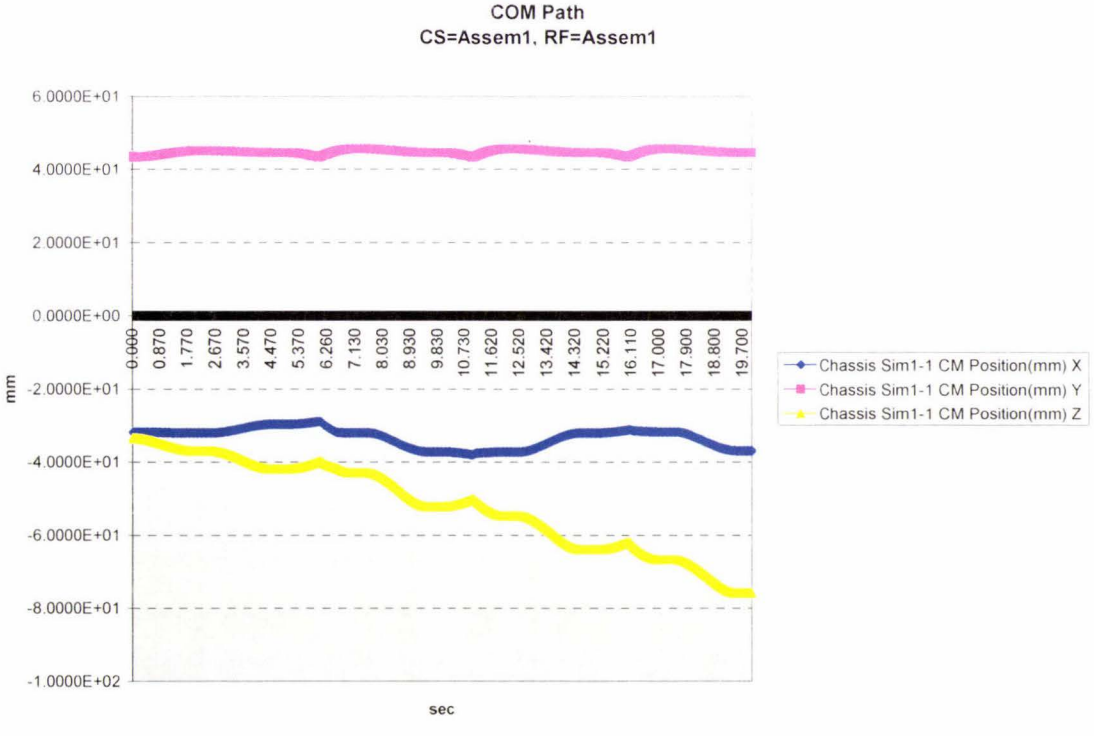

Figure 49: COM Path, Beta Run, Geometry Beta-1, Sample E

Geometry Beta-2: Wh = 120 mm

5.3.2.1.3 Driving Functions

Revolute		5.529	5.238	6.111	6.402
	A	B	C	D	E
	v-leg angle	5.82	5.522591	5.238774	6.10566	6.404544
	time to turn	2	2	2	2	2
	time gap	0.5	0.5	0.5	0.5	0.5
	Equations	STEP(TIME,0,0D,2.00,5.82D)+STEP(TIME,5.00,0D,7.00,-11.64D)+STEP(TIME,10.00,0D,12.00,11.64D)+STEP(TIME,15.00,0D,17.00,-11.64D)				
		STEP(TIME,0,0D,2.00,5.52D)+STEP(TIME,5.00,0D,7.00,-11.05D)+STEP(TIME,10.00,0D,12.00,11.05D)+STEP(TIME,15.00,0D,17.00,-11.05D)				
		STEP(TIME,0,0D,2.00,5.24D)+STEP(TIME,5.00,0D,7.00,-10.48D)+STEP(TIME,10.00,0D,12.00,10.48D)+STEP(TIME,15.00,0D,17.00,-10.48D)				
		STEP(TIME,0,0D,2.00,6.11D)+STEP(TIME,5.00,0D,7.00,-12.21D)+STEP(TIME,10.00,0D,12.00,12.21D)+STEP(TIME,15.00,0D,17.00,-12.21D)				
		STEP(TIME,0,0D,2.00,6.40D)+STEP(TIME,5.00,0D,7.00,-12.81D)+STEP(TIME,10.00,0D,12.00,12.81D)+STEP(TIME,15.00,0D,17.00,-12.81D)				

Revolute2	A	B	C	D	E
h-leg angle	-20.56	-19.22	-17.657	-20.4012	-20.2679
time to turn	2	2	2	2	2
time gap	0.5	0.5	0.5	0.5	0.5
Equations	STEP(TIME,2.50,0D,4.50,-20.56D)+STEP(TIME,7.50,0D,9.50,41.12D)+STEP(TIME,12.50,0D,14.50,-41.12D)+STEP(TIME,17.50,0D,19.50,41.12D)				
	STEP(TIME,2.50,0D,4.50,-19.22D)+STEP(TIME,7.50,0D,9.50,38.44D)+STEP(TIME,12.50,0D,14.50,-38.44D)+STEP(TIME,17.50,0D,19.50,38.44D)				
	STEP(TIME,2.50,0D,4.50,-17.66D)+STEP(TIME,7.50,0D,9.50,35.31D)+STEP(TIME,12.50,0D,14.50,-35.31D)+STEP(TIME,17.50,0D,19.50,35.31D)				
	STEP(TIME,2.50,0D,4.50,-20.40D)+STEP(TIME,7.50,0D,9.50,40.80D)+STEP(TIME,12.50,0D,14.50,-40.80D)+STEP(TIME,17.50,0D,19.50,40.80D)				
	STEP(TIME,2.50,0D,4.50,-20.27D)+STEP(TIME,7.50,0D,9.50,40.54D)+STEP(TIME,12.50,0D,14.50,-40.54D)+STEP(TIME,17.50,0D,19.50,40.54D)				

Table 9: Driving Functions, Beta Run, Geometry Beta-2, Wh = 120 mm

5.3.2.1.4 COM Paths

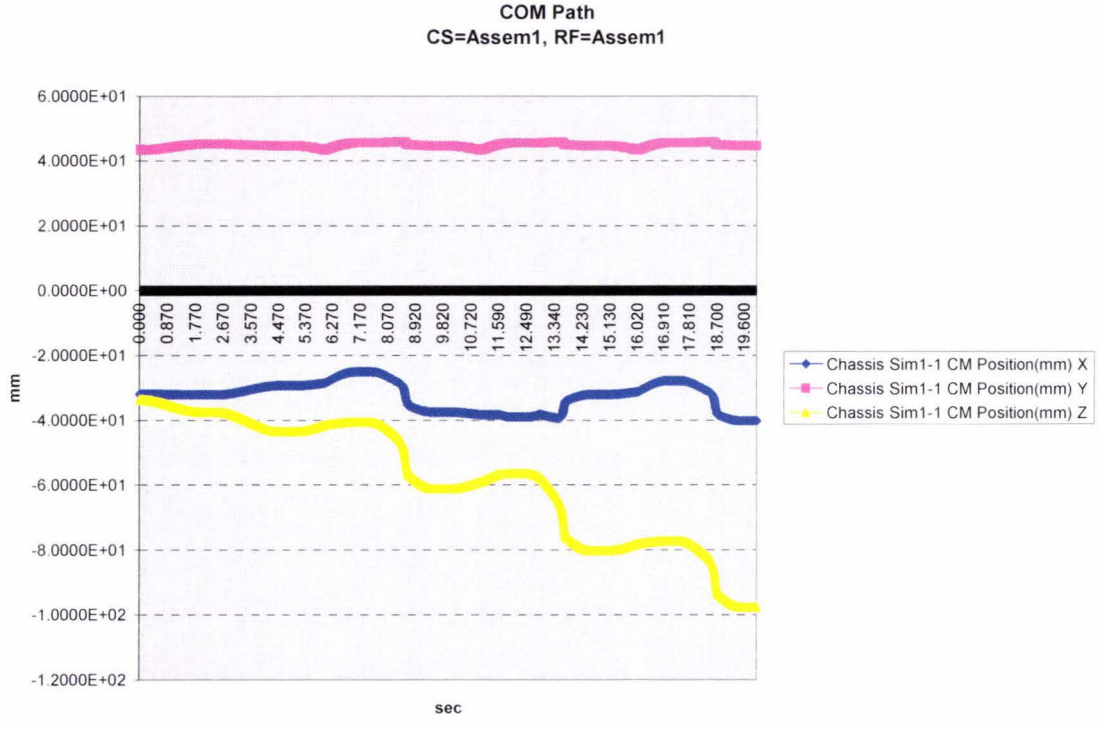

Figure 50: COM Path, Beta Run, Geometry Beta-2, Sample C

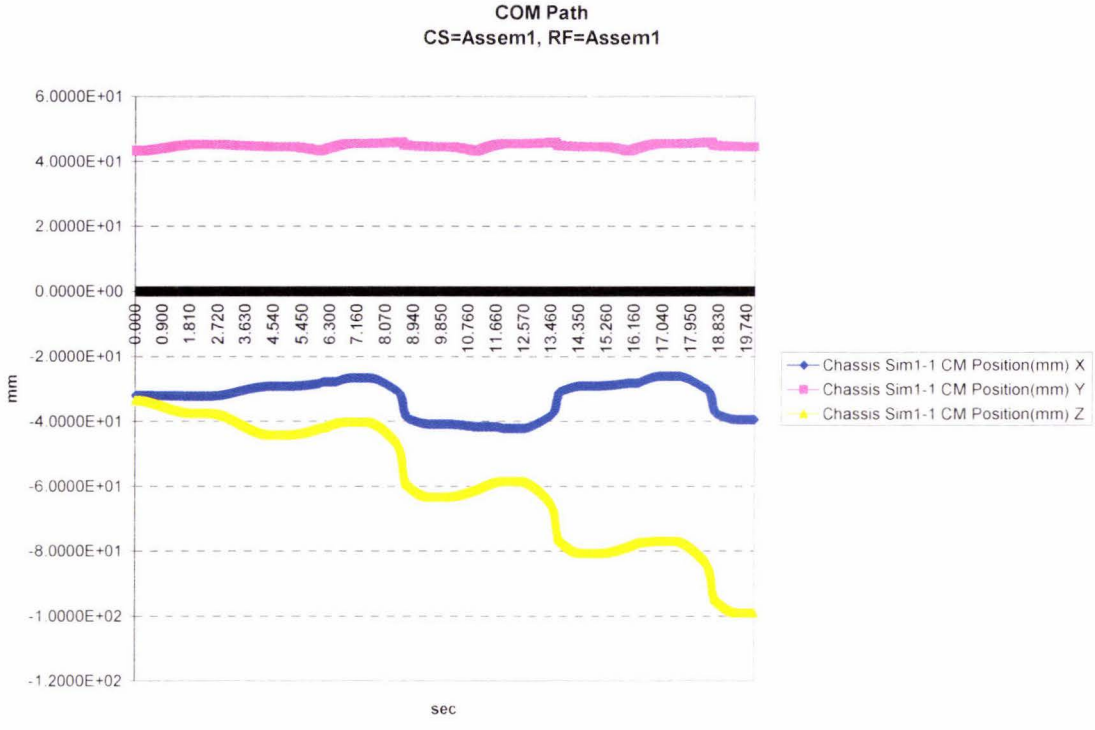

Figure 51: COM Path, Beta Run, Geometry Beta-2, Sample B

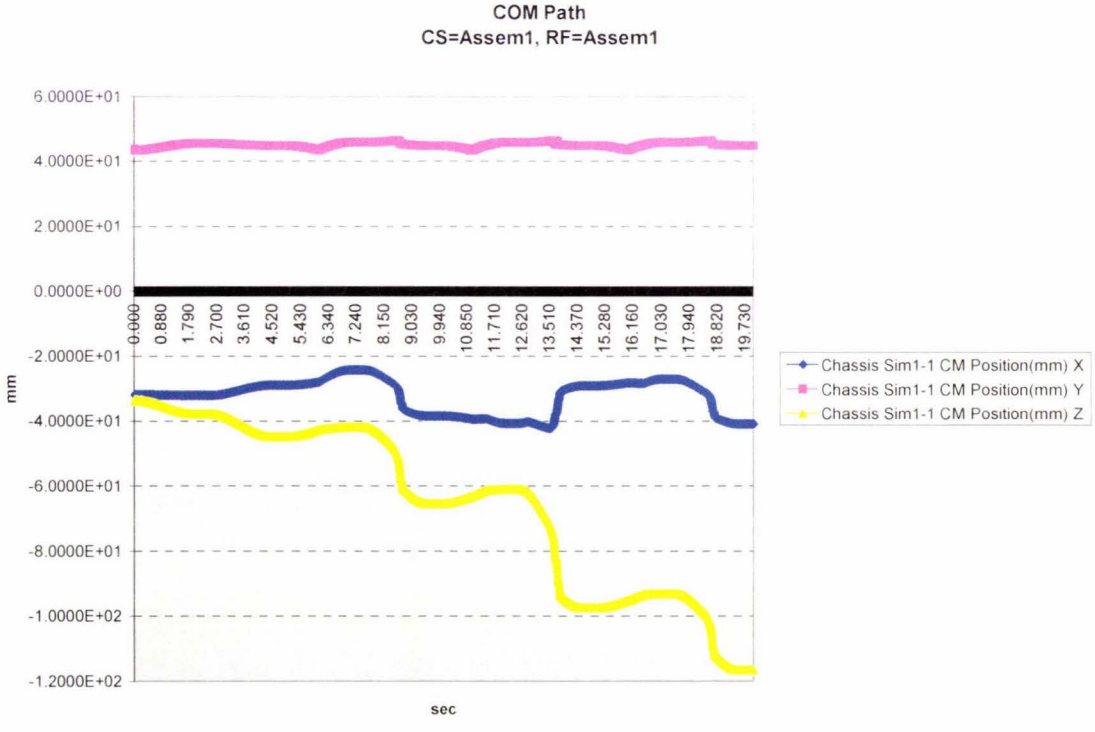

Figure 52: COM Path, Beta Run, Geometry Beta-2, Sample A

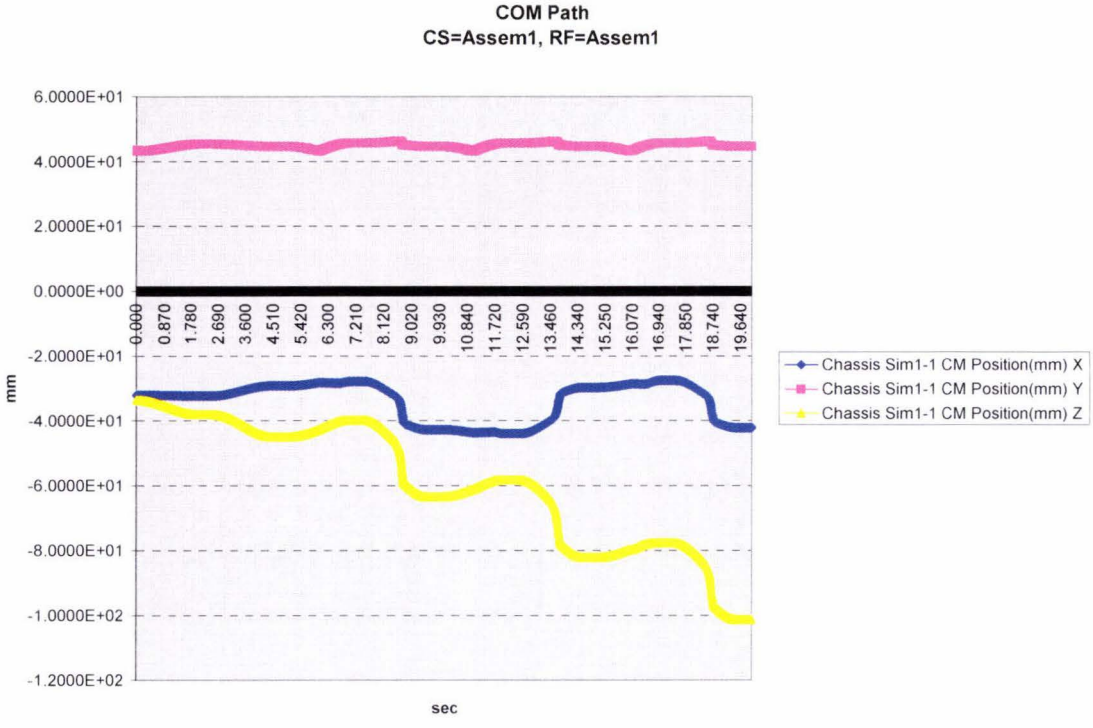

Figure 53: COM Path, Beta Run, Geometry Beta-2, Sample D

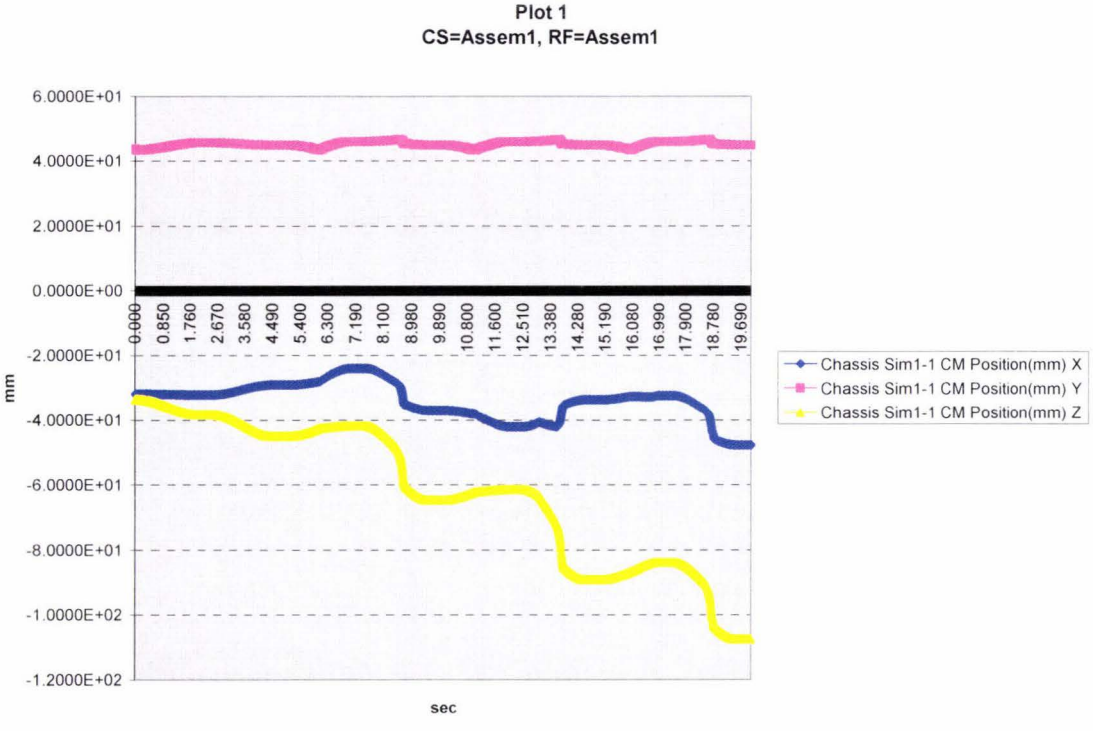

Figure 54: COM Path, Beta Run, Geometry Beta-2, Sample E

5.3.2.2 Geometry Beta-3: Wh = 150 mm

5.3.2.2.1 Driving Functions

		8.185892	7.755055	9.047565	9.478401
Revolute	A	B	C	D	E
v-leg angle	8.616728	8.207785	7.724498	9.030927	9.450023
time to turn	2	2	2	2	2
time gap	0.5	0.5	0.5	0.5	0.5
Equations	STEP(TIME,0.0D,2.00,8.62D)+STEP(TIME,5.00,0D,7.00,-17.23D)+STEP(TIME,10.00,0D,12.00,17.23D)+STEP(TIME,15.00,0D,17.00,-17.23D)				
	STEP(TIME,0.0D,2.00,8.21D)+STEP(TIME,5.00,0D,7.00,-16.42D)+STEP(TIME,10.00,0D,12.00,16.42D)+STEP(TIME,15.00,0D,17.00,-16.42D)				
	STEP(TIME,0.0D,2.00,7.72D)+STEP(TIME,5.00,0D,7.00,-15.45D)+STEP(TIME,10.00,0D,12.00,15.45D)+STEP(TIME,15.00,0D,17.00,-15.45D)				
	STEP(TIME,0.0D,2.00,9.03D)+STEP(TIME,5.00,0D,7.00,-18.06D)+STEP(TIME,10.00,0D,12.00,18.06D)+STEP(TIME,15.00,0D,17.00,-18.06D)				
	STEP(TIME,0.0D,2.00,9.45D)+STEP(TIME,5.00,0D,7.00,-18.90D)+STEP(TIME,10.00,0D,12.00,18.90D)+STEP(TIME,15.00,0D,17.00,-18.90D)				
Revolute2	A	B	C	D	E
h-leg angle	-28.3396	-26.3219	-23.5484	-28.2071	-28.042
time to turn	2	2	2	2	2
time gap	0.5	0.5	0.5	0.5	0.5
Equations	STEP(TIME,2.50,0D,4.50,-28.34D)+STEP(TIME,7.50,0D,9.50,56.68D)+STEP(TIME,12.50,0D,14.50,-56.68D)+STEP(TIME,17.50,0D,19.50,56.68D)				
	STEP(TIME,2.50,0D,4.50,-26.32D)+STEP(TIME,7.50,0D,9.50,52.64D)+STEP(TIME,12.50,0D,14.50,-52.64D)+STEP(TIME,17.50,0D,19.50,52.64D)				
	STEP(TIME,2.50,0D,4.50,-23.55D)+STEP(TIME,7.50,0D,9.50,47.10D)+STEP(TIME,12.50,0D,14.50,-47.10D)+STEP(TIME,17.50,0D,19.50,47.10D)				
	STEP(TIME,2.50,0D,4.50,-28.21D)+STEP(TIME,7.50,0D,9.50,56.41D)+STEP(TIME,12.50,0D,14.50,-56.41D)+STEP(TIME,17.50,0D,19.50,56.41D)				
	STEP(TIME,2.50,0D,4.50,-28.04D)+STEP(TIME,7.50,0D,9.50,56.08D)+STEP(TIME,12.50,0D,14.50,-56.08D)+STEP(TIME,17.50,0D,19.50,56.08D)				

Table 10: Driving Functions, Beta Run, Geometry Beta-3, Wh = 150 mm

5.3.2.2.2 COM Paths

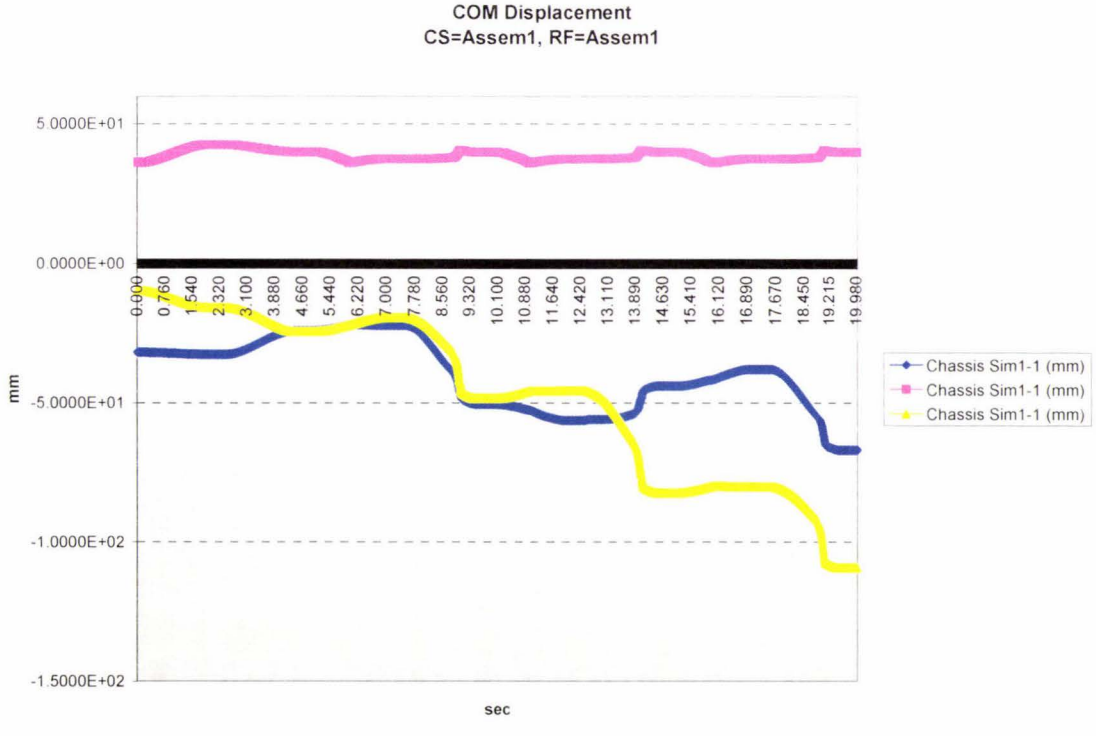

Figure 55: COM Path, Beta Run, Geometry Beta-3, Sample C

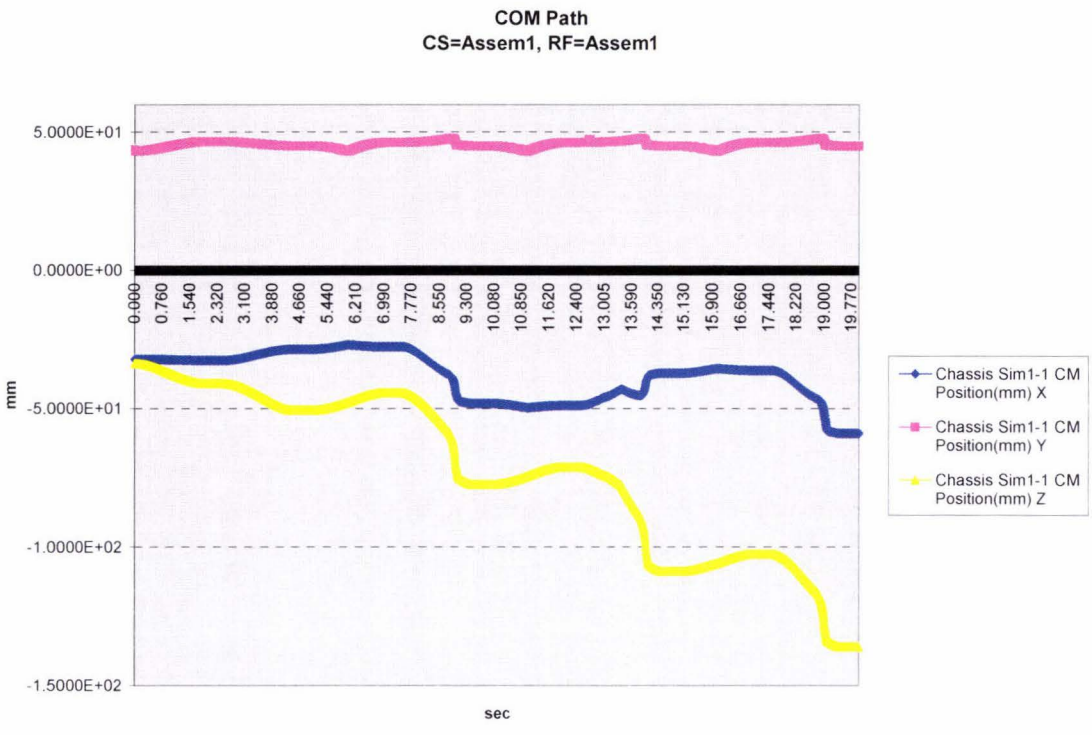

Figure 56: COM Path, Beta Run, Geometry Beta-3, Sample B

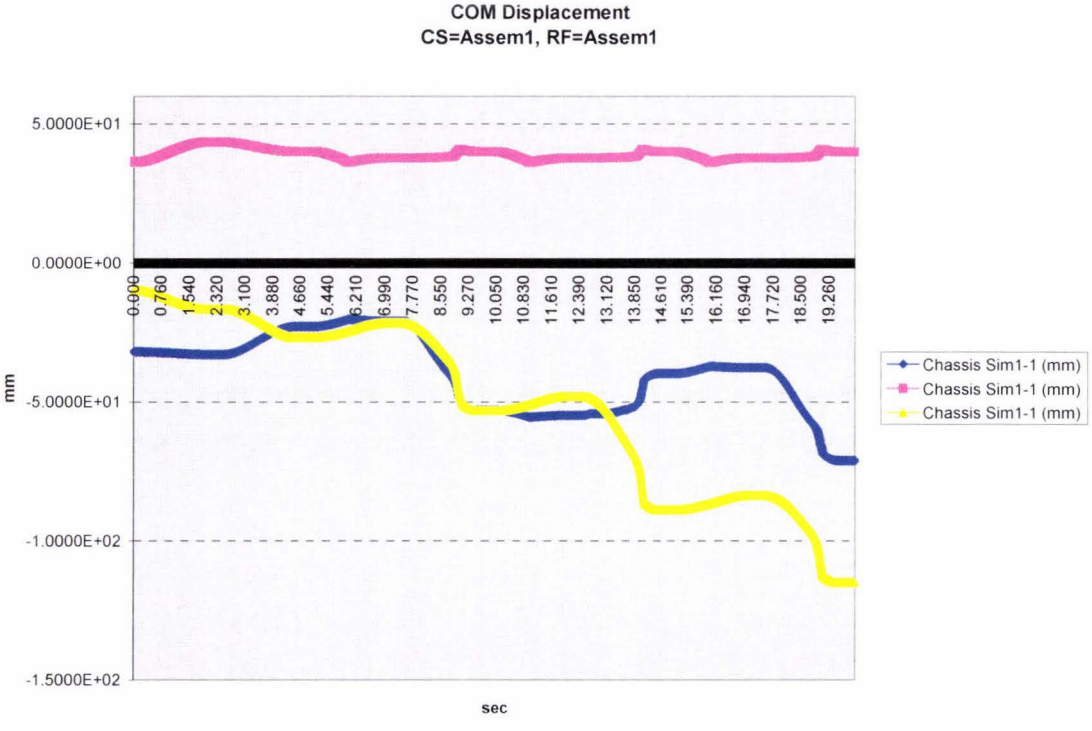

Figure 57: COM Path, Beta Run, Geometry Beta-3, Sample A

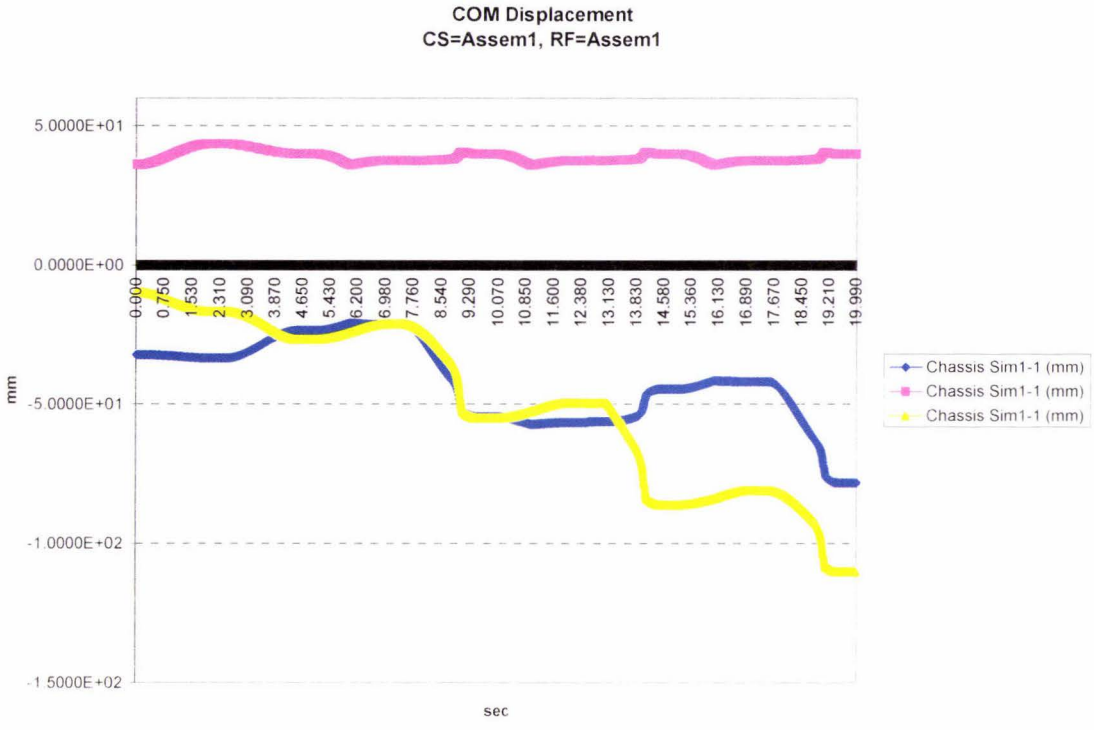

Figure 58: COM Path, Beta Run, Geometry Beta-3, Sample D

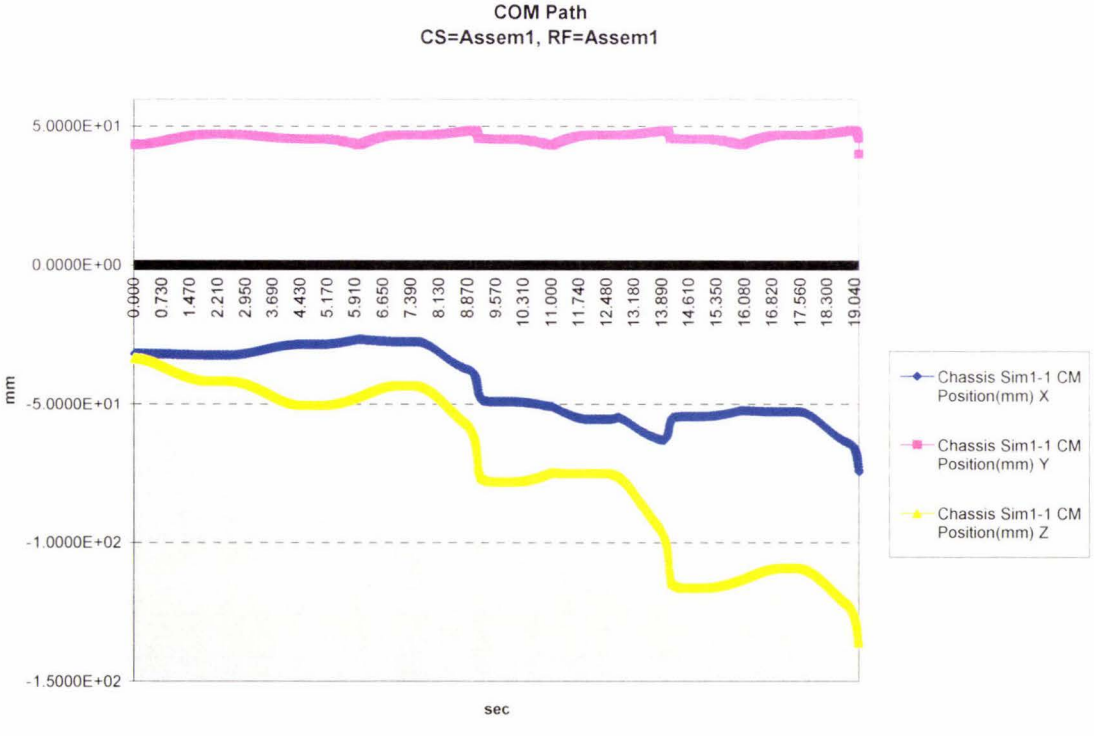

Figure 59: COM Path, Beta Run, Geometry Beta-3, Sample E

5.3.2.3 Geometry Beta-4: Wh = 170 mm

5.3.2.3.1 Driving Functions

		8.421408	7.978176	9.307872	9.751104
Revolute		A	B	C	D	E
	v-leg angle	8.86464	8.4525	7.965092	9.28182	9.788582
	time to turn	2	2	2	2	2
	time gap	0.5	0.5	0.5	0.5	0.5
	Equations	STEP(TIME,0.0D,2.00,8.86D)+STEP(TIME,5.00,0D,7.00,-17.73D)+STEP(TIME,10.00,0D,12.00,17.73D)+STEP(TIME,15.00,0D,17.00,-17.73D)				
		STEP(TIME,0,0D,2.00,8.45D)+STEP(TIME,5.00,0D,7.00,-16.91D)+STEP(TIME,10.00,0D,12.00,16.91D)+STEP(TIME,15.00,0D,17.00,-16.91D)				
		STEP(TIME,0,0D,2.00,7.97D)+STEP(TIME,5.00,0D,7.00,-15.93D)+STEP(TIME,10.00,0D,12.00,15.93D)+STEP(TIME,15.00,0D,17.00,-15.93D)				
		STEP(TIME,0,0D,2.00,9.28D)+STEP(TIME,5.00,0D,7.00,-18.56D)+STEP(TIME,10.00,0D,12.00,18.56D)+STEP(TIME,15.00,0D,17.00,-18.56D)				
		STEP(TIME,0,0D,2.00,9.79D)+STEP(TIME,5.00,0D,7.00,-19.58D)+STEP(TIME,10.00,0D,12.00,19.58D)+STEP(TIME,15.00,0D,17.00,-19.58D)				
Revolute2		A	B	C	D	E
	h-leg angle	-30.4607	-28.5309	-25.777	-30.3567	-30.173
	time to turn	2	2	2	2	2
	time gap	0.5	0.5	0.5	0.5	0.5
	Equations	STEP(TIME,2.50,0D,4.50,-30.46D)+STEP(TIME,7.50,0D,9.50,60.92D)+STEP(TIME,12.50,0D,14.50,-60.92D)+STEP(TIME,17.50,0D,19.50,60.92D)				
		STEP(TIME,2.50,0D,4.50,-28.53D)+STEP(TIME,7.50,0D,9.50,57.06D)+STEP(TIME,12.50,0D,14.50,-57.06D)+STEP(TIME,17.50,0D,19.50,57.06D)				
		STEP(TIME,2.50,0D,4.50,-25.78D)+STEP(TIME,7.50,0D,9.50,51.55D)+STEP(TIME,12.50,0D,14.50,-51.55D)+STEP(TIME,17.50,0D,19.50,51.55D)				
		STEP(TIME,2.50,0D,4.50,-30.36D)+STEP(TIME,7.50,0D,9.50,60.71D)+STEP(TIME,12.50,0D,14.50,-60.71D)+STEP(TIME,17.50,0D,19.50,60.71D)				
		STEP(TIME,2.50,0D,4.50,-30.17D)+STEP(TIME,7.50,0D,9.50,60.35D)+STEP(TIME,12.50,0D,14.50,-60.35D)+STEP(TIME,17.50,0D,19.50,60.35D				

Table 11: Driving Functions, Beta Run, Geometry Beta-4, Wh = 170 mm

5.3.2.3.2 COM Paths

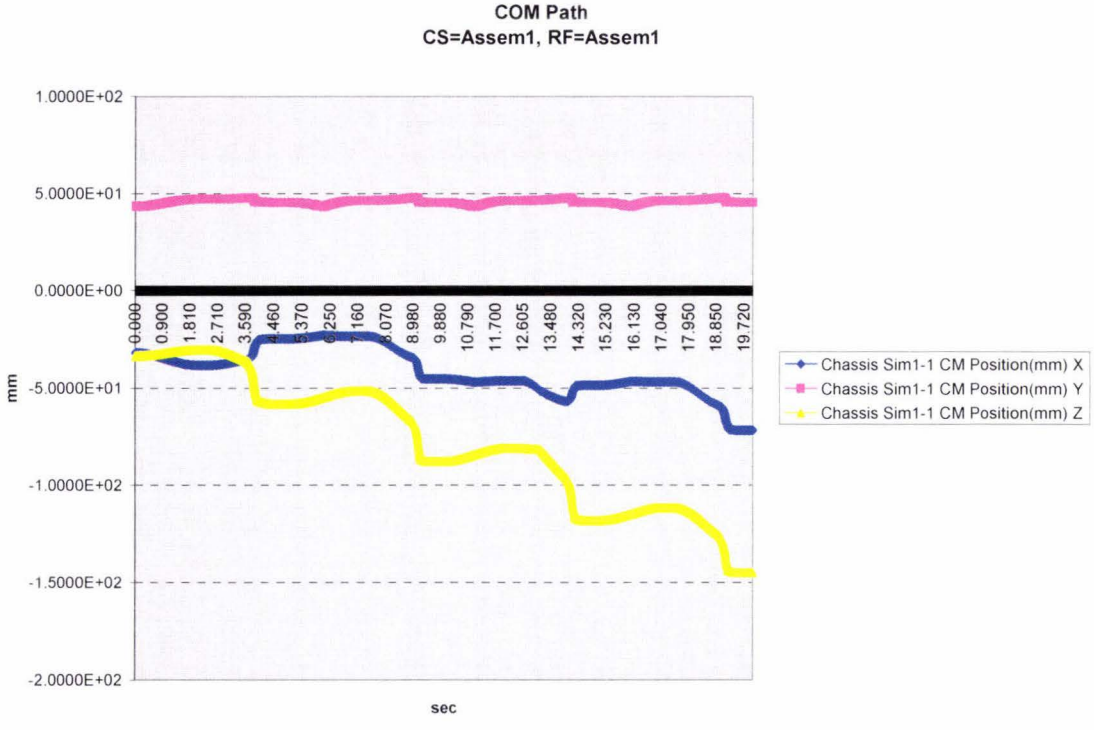

Figure 60: COM Path, Beta Run, Geometry Beta-4, Sample C

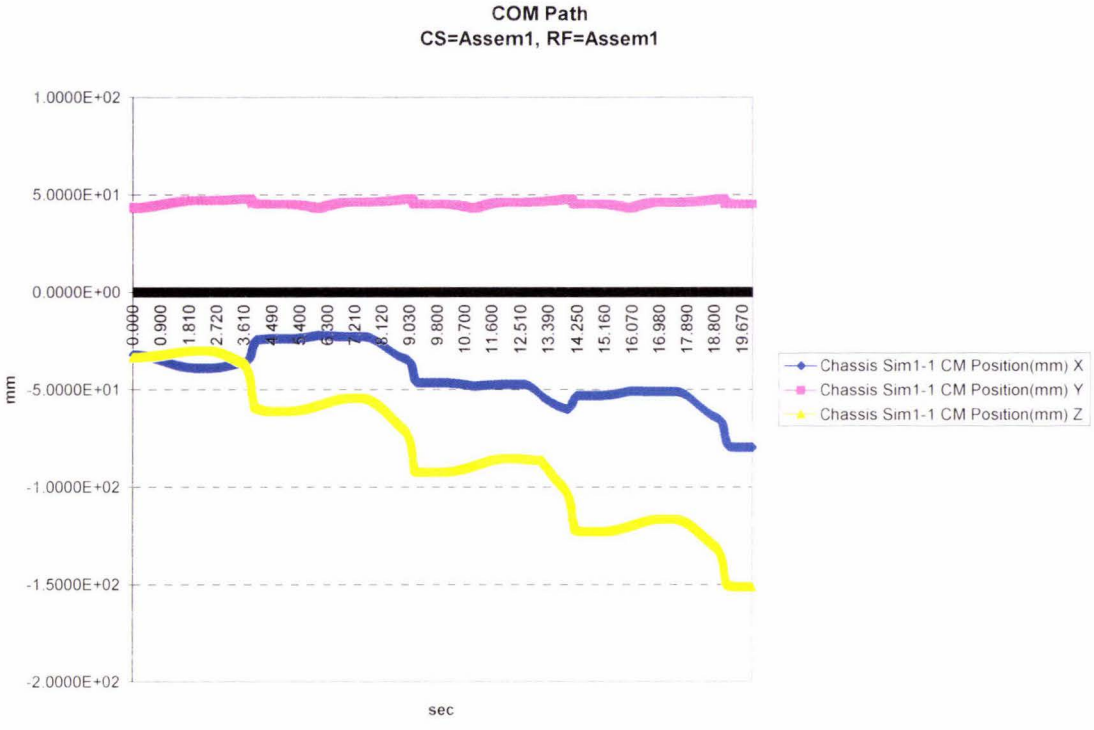

Figure 61: COM Path, Beta Run, Geometry Beta-4, Sample B

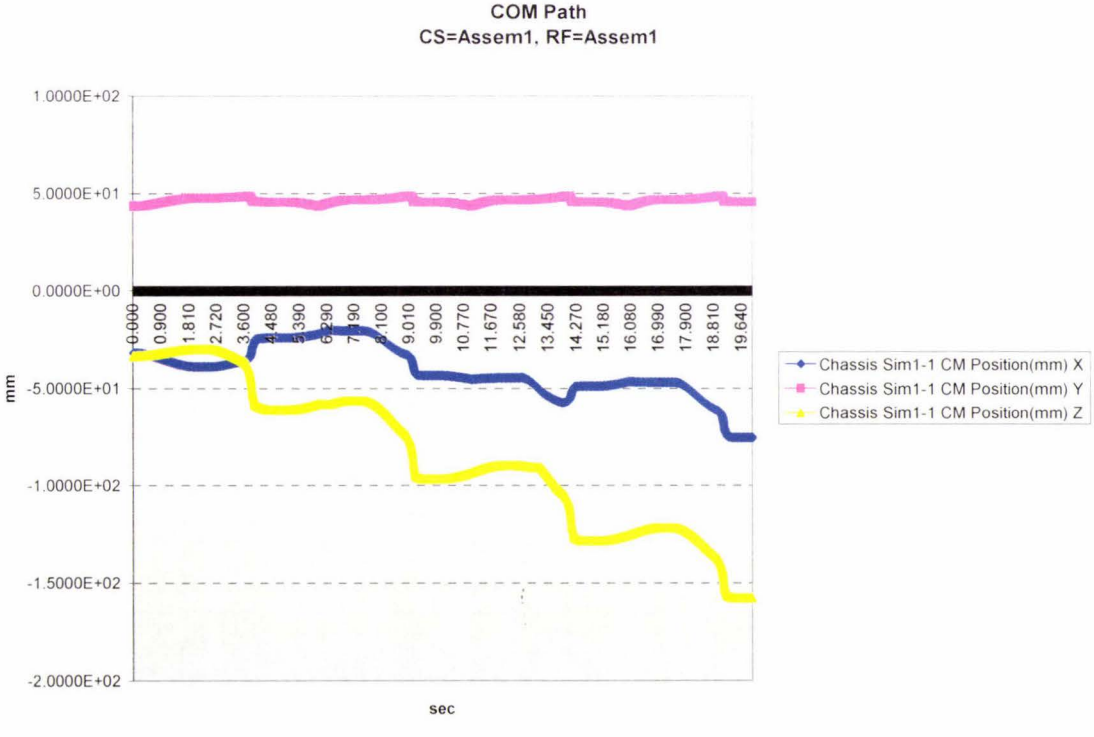

Figure 62: COM Path, Beta Run, Geometry Beta-4, Sample A

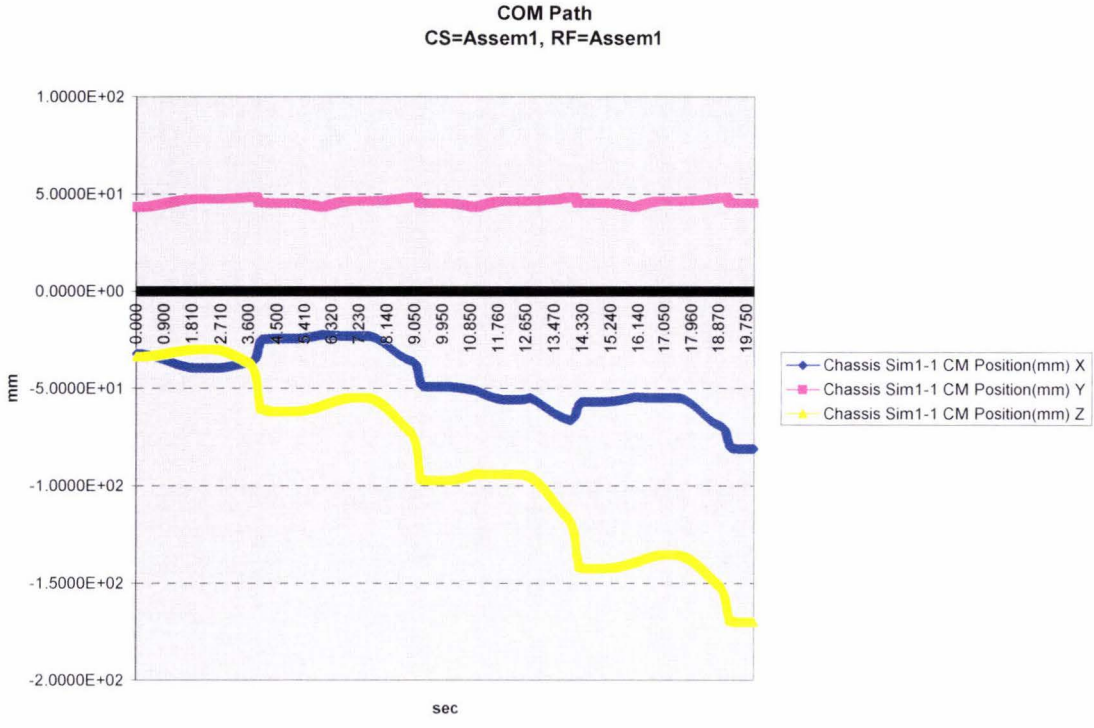

Figure 63: COM Path, Beta Run, Geometry Beta-4, Sample D

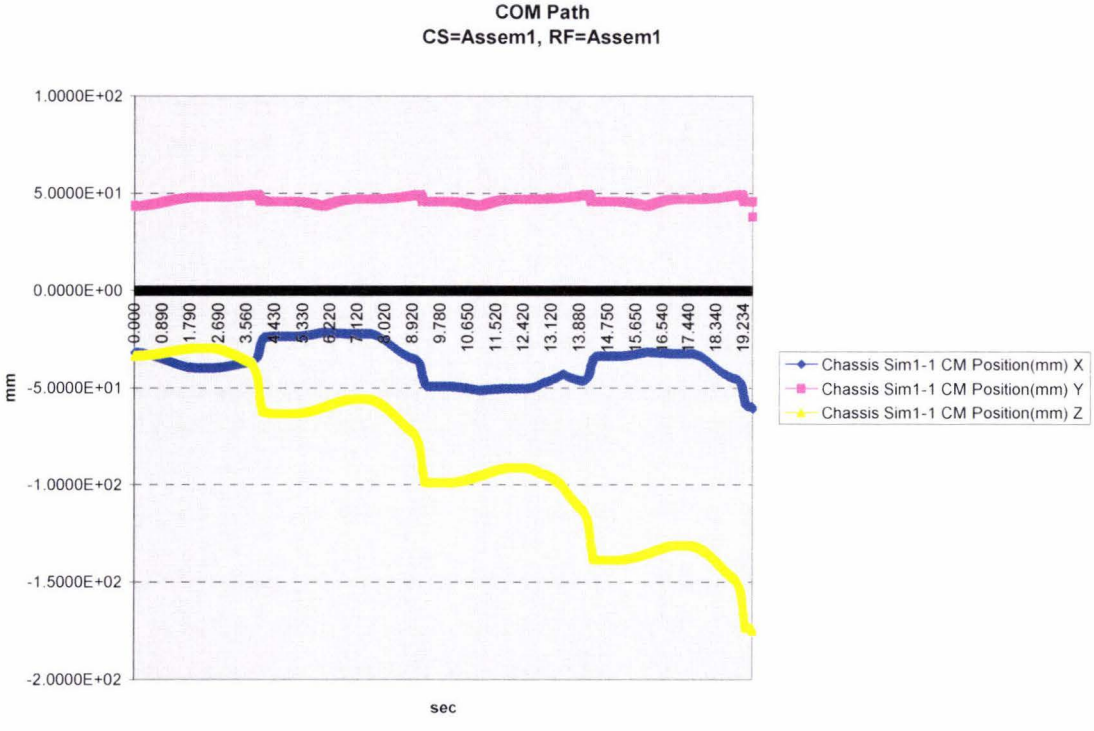

Figure 64: COM Path, Beta Run, Geometry Beta-4, Sample E

5.3.2.4 Geometry Beta-5: Wh = 200 mm

5.3.2.4.1 Driving Functions

		7.566837	7.168583	8.363346	8.761601
Revolute		A	B	C	D	E
	v-leg angle	7.965092	7.565307	7.171714	8.370708	8.781797
	time to turn	2	2	2	2	2
	time gap	0.5	0.5	0.5	0.5	0.5
	Equations	STEP(TIME,0.0D,2.00,7.97D)+STEP(TIME,5.00,0D,7.00,-15.93D)+STEP(TIME,10.00,0D,12.00,15.93D)+STEP(TIME,15.00,0D,17.00,-15.93D)				
		STEP(TIME,0.0D,2.00,7.57D)+STEP(TIME,5.00,0D,7.00,-15.13D)+STEP(TIME,10.00,0D,12.00,15.13D)+STEP(TIME,15.00,0D,17.00,-15.13D)				
		STEP(TIME,0.0D,2.00,7.17D)+STEP(TIME,5.00,0D,7.00,-14.34D)+STEP(TIME,10.00,0D,12.00,14.34D)+STEP(TIME,15.00,0D,17.00,-14.34D)				
		STEP(TIME,0.0D,2.00,8.37D)+STEP(TIME,5.00,0D,7.00,-16.74D)+STEP(TIME,10.00,0D,12.00,16.74D)+STEP(TIME,15.00,0D,17.00,-16.74D)				
		STEP(TIME,0.0D,2.00,8.78D)+STEP(TIME,5.00,0D,7.00,-17.56D)+STEP(TIME,10.00,0D,12.00,17.56D)+STEP(TIME,15.00,0D,17.00,-17.56D)				
Revolute2		A	B	C	D	E
	h-leg angle	-16.5721	-16.344	-15.3317	-16.4264	-16.1864
	time to turn	2	2	2	2	2
	time gap	0.5	0.5	0.5	0.5	0.5
	Equations	STEP(TIME,2.50,0D,4.50,-16.57D)+STEP(TIME,7.50,0D,9.50,33.14D)+STEP(TIME,12.50,0D,14.50,-33.14D)+STEP(TIME,17.50,0D,19.50,33.14D)				
		STEP(TIME,2.50,0D,4.50,-16.34D)+STEP(TIME,7.50,0D,9.50,32.69D)+STEP(TIME,12.50,0D,14.50,-32.69D)+STEP(TIME,17.50,0D,19.50,32.69D)				
		STEP(TIME,2.50,0D,4.50,-15.33D)+STEP(TIME,7.50,0D,9.50,30.66D)+STEP(TIME,12.50,0D,14.50,-30.66D)+STEP(TIME,17.50,0D,19.50,30.66D)				
		STEP(TIME,2.50,0D,4.50,-16.43D)+STEP(TIME,7.50,0D,9.50,32.85D)+STEP(TIME,12.50,0D,14.50,-32.85D)+STEP(TIME,17.50,0D,19.50,32.85D)				
		STEP(TIME,2.50,0D,4.50,-16.19D)+STEP(TIME,7.50,0D,9.50,32.37D)+STEP(TIME,12.50,0D,14.50,-32.37D)+STEP(TIME,17.50,0D,19.50,32.37D)				

Table 12: Driving Functions, Beta Run, Geometry Beta-5, Wh=200 mm

5.3.2.4.2 COM Paths

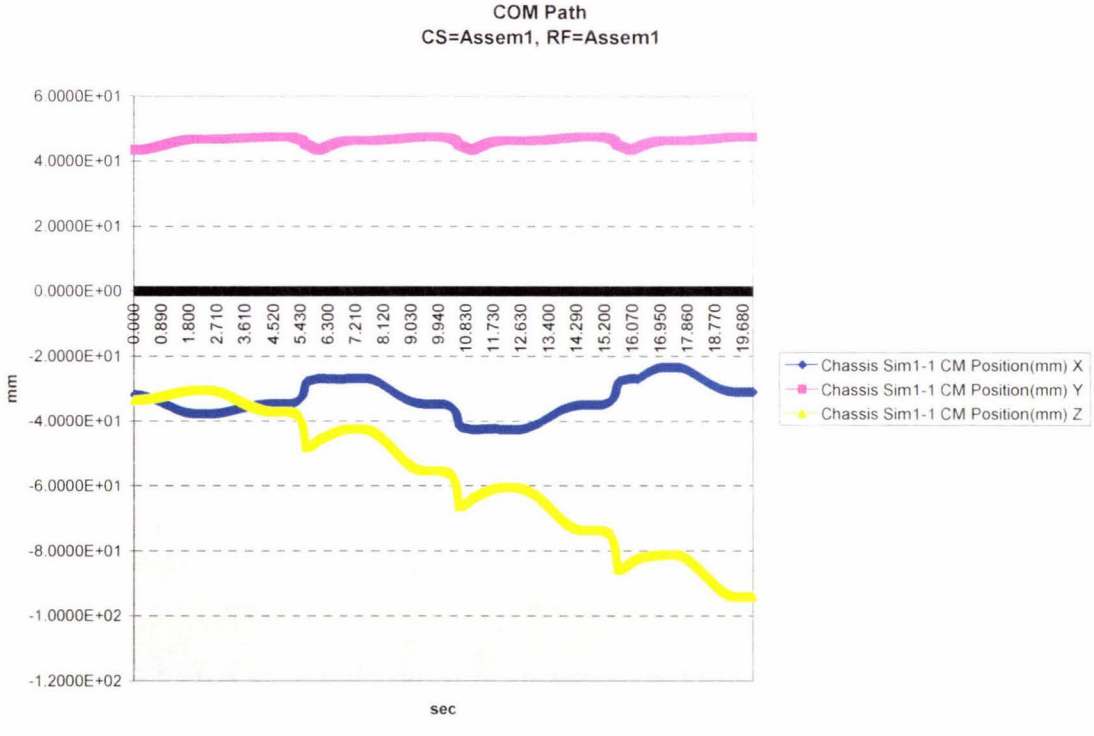

Figure 65: COM Path, Beta Run, Geometry Beta 5, Sample C

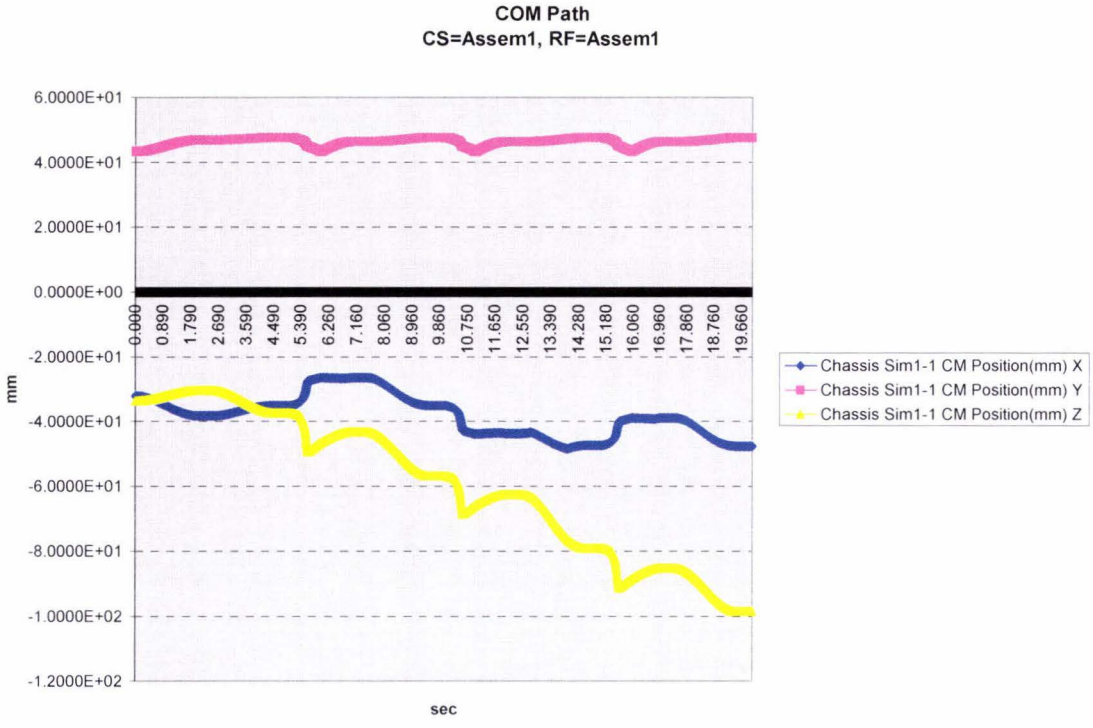

Figure 66: COM Path, Beta Run, Geometry Beta 5, Sample B

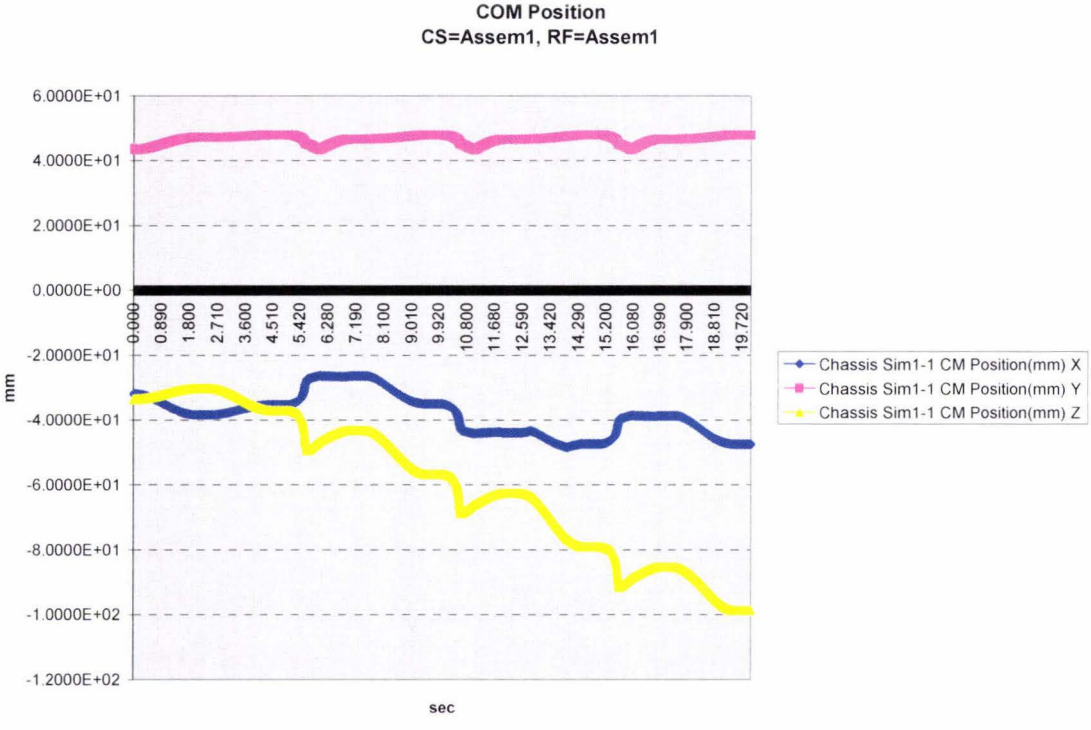

Figure 67: COM Path, Beta Run, Geometry Beta 5, Sample A

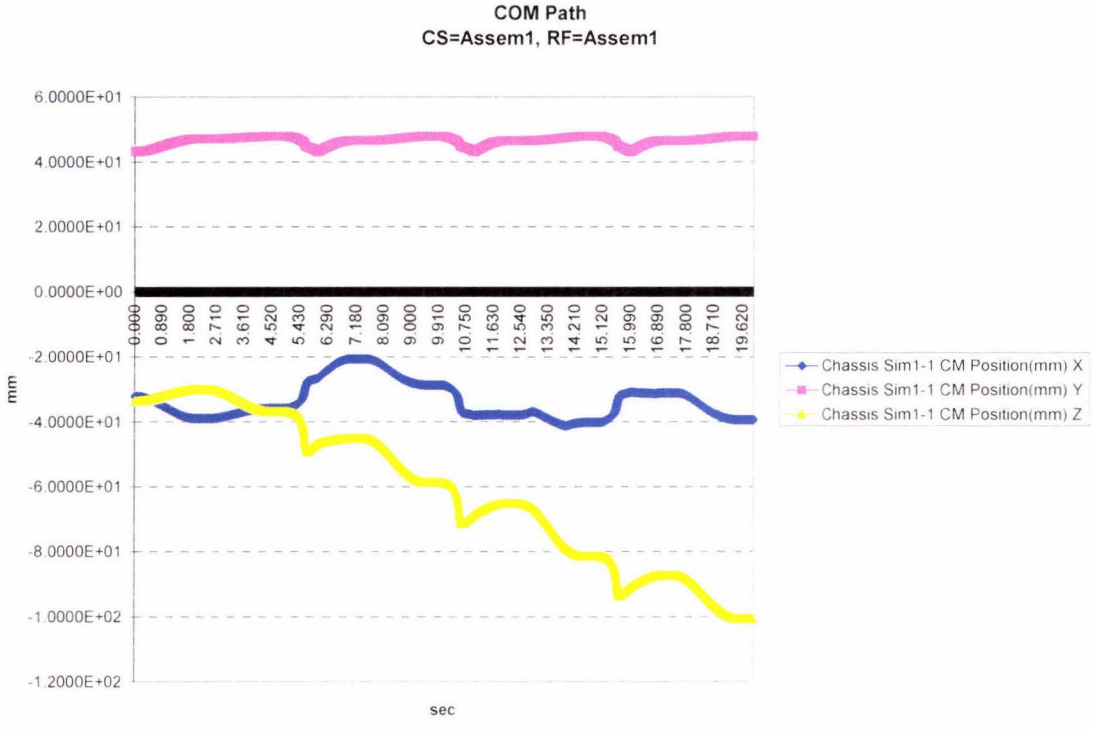

Figure 68: COM Path, Beta Run, Geometry Beta 5, Sample D

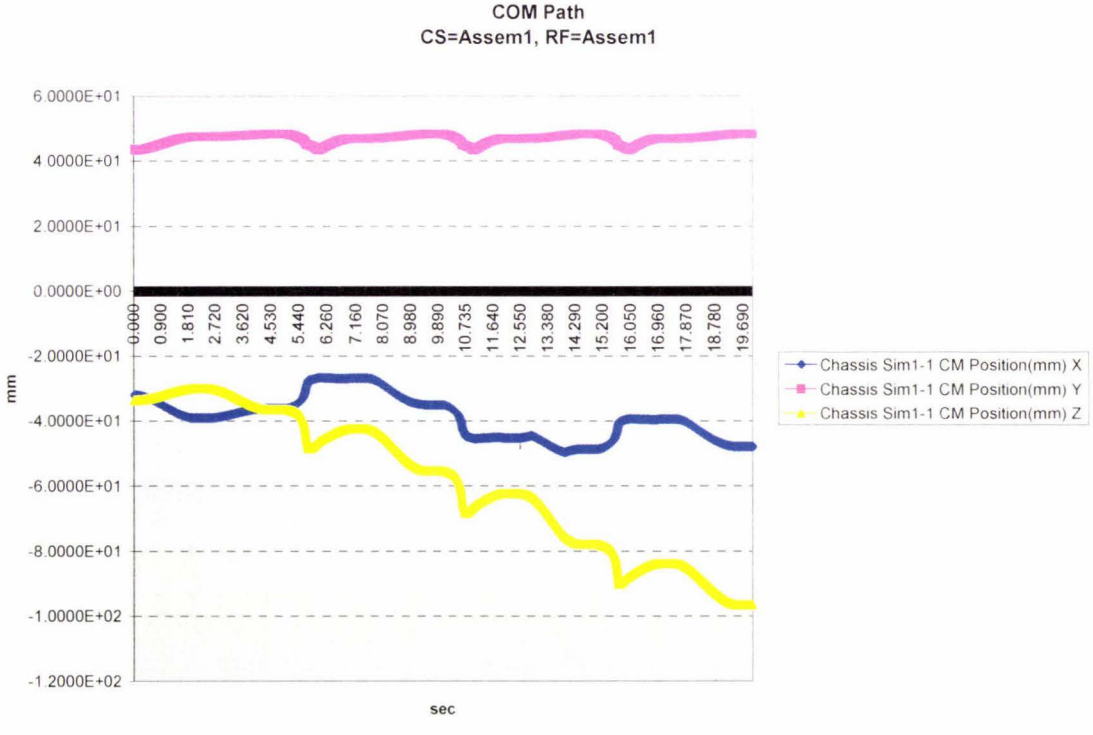

Figure 69: COM Path, Beta Run, Geometry Beta 5, Sample E

6 Discussion & Conclusion

6.1 Discussion

6.1.1 Simulation Run Alpha

This simulation run consisted of rotating the joint between the v-leg and the main chassis, with the h-leg joint allowed to rotate freely. The controlled variable was the width of the h-leg, and the outputs that were measured were the angles of the joints, the power expended by the moving joint, the position of the robot's COM (Centre of Mass), and the reaction forces between the robot and the ground.

In analyzing these results, one of the most obvious features is three distinct zones, observed in the joint-angle/time graphs. As the v-leg angle increases, the h-leg angle also increases. It appears to behave in three distinct behaviours, categorized in three zones: sub-critical, critical, and super-critical.

When in the sub critical mode, the leg's feet do not slip on the ground, and generally as the v-leg turns, one of the h-leg's feet is lifted off the ground. This is the sub-critical zone, and demonstrates a level of geometric stability – within this zone the robot reposes in a stable tripod stance.

The one major deviation from this condition is when one of the v-leg's feet initially lift off the ground. At a certain point, the robot's balance is upset as the COM is moved across the balance line defined by the two "inside" legs of the tripod stance. When this occurs, the robot tips over so that both front feet are resting on the ground; the rest of the step continues as normal. This "tipping" behaviour is covered in more detail in Vertical Leg.

The sub-critical zone is the range in which most of the forward movement of the COM occurs.

At a certain point, henceforth referred to as the "critical point" or "critical angle", the geometry has shifted such that the single rear foot on the leg now begins to slip. The angle of the h-leg remains relatively constant as the v-leg angle continues to increase; this is the "critical zone". Some gradual motion of the COM still occurs in this region.

At the "super-critical angle", the system then progresses into the super-critical zone – robot collapse. At this point, the h-leg loses traction on the ground and collapses, dragging the rest of the robot with it. COSMOS/Motion often didn't deal with these situations very well, and thus either the simulation was terminated with an ADAMS error, or a glitch occurred, potentially throwing the robot energetically off the ground object.

As an example case to consider, we will examine the case where the width of the v-leg w_v=100 mm, and the width of the h-leg w_h=140 mm.

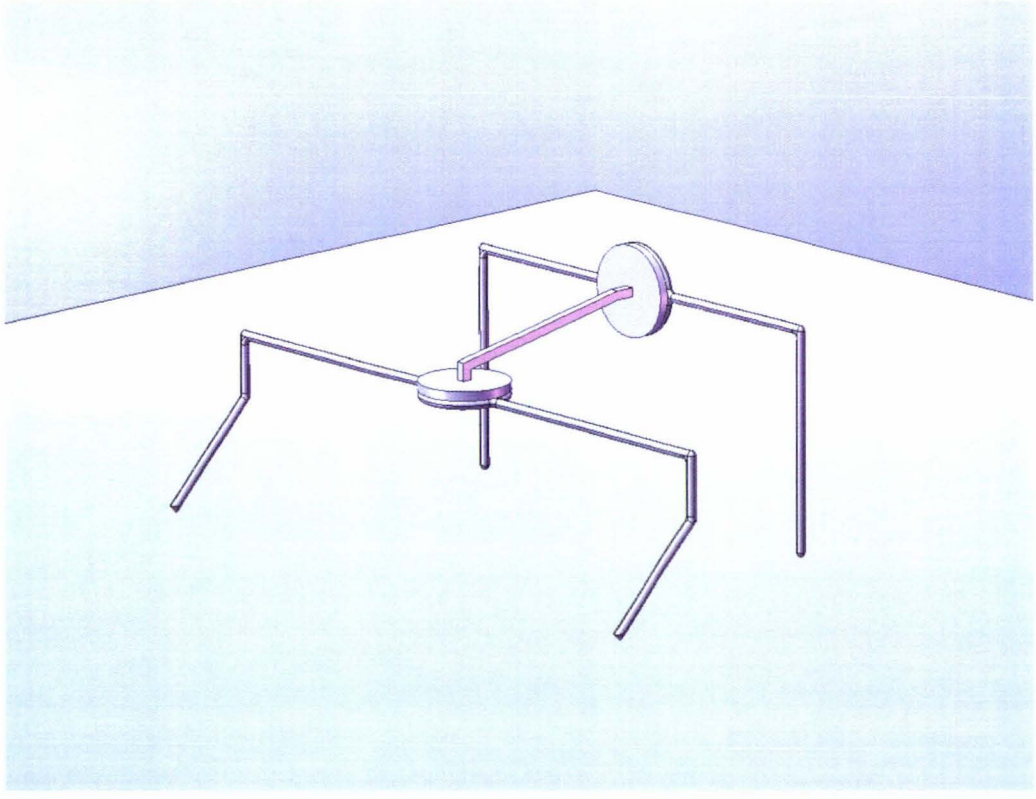

Figure 70: Simulation Run Alpha Robot, Square Stance, H-Leg Width = 140 mm

The model is driven only at the v-leg join, and the input function is:

displacement = step(f(t = 0s) = 0 deg, f(t = 10s) = 90 deg)

When simulated, the following graphs were produced of the joint angles throughout the simulation:

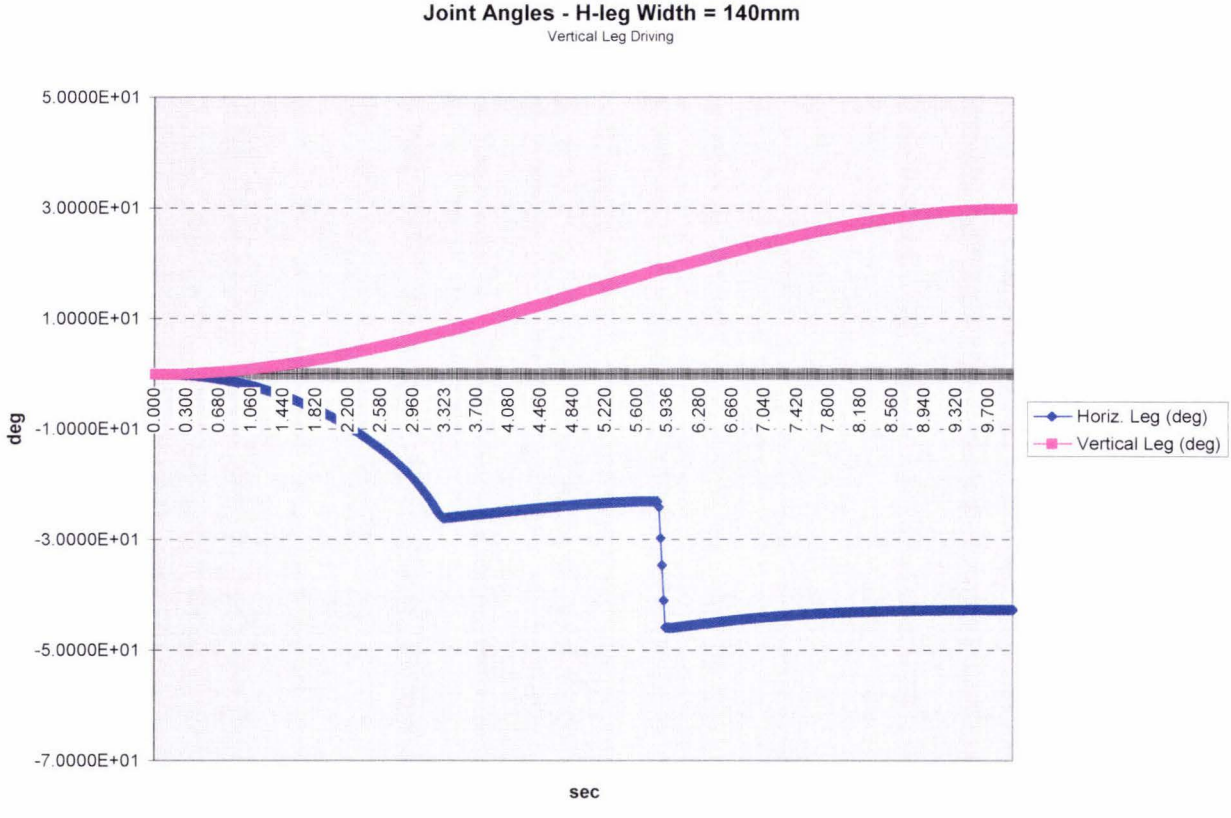

Figure 71: Joint angles throughout Alpha Run Simulation, H-leg Width = 130 mm

Clearly seen on this graph of the joint angles is the zones of behaviour, and in particular, the -sharp "elbow" in the graph which signals the end of the sub-critical region and the beginning of foot slippage in the critical region.

For example, in the case showing in Figure 71, the critical angle (i.e. the v-leg angle at which the transition from sub-critical to critical zones behaviours in the h-leg) is 8 degrees, occurring at around 3.3 seconds. As this point, the exhibited h-leg joint angle is -26 degrees.

These values were noted, for each of the simulations executed, and saved for use in later simulation runs.

6.1.2 Simulation Run Beta

For simulation run beta, the goal was to produce an entire stepping cycle for the simulated robot, and determine whether the "critical angles" measured in the alpha simulation resulted in viable walking gaits for various widths of h-leg. This simulation also attempts to determine whether the measured angles were the "optimum parameters" for walking.

The major difference between the alpha and beta simulations is that in the beta simulation, both legs are now "fixed" - both h-leg and v-leg joints are now driven by displacement type functions, meaning that they will remain locked even when the joint is not "active".

To implement an entire step-cycle in simulation, starting from "square" stance, the following steps needed to be performed, simulating the pattern of movement "commands" that would be produced by the Nv-net controller.

Step 1: The v-leg joint must be rotated counter-clockwise to the relevant critical angle, thus rear left foot of the h-leg off the ground.

Step 2: The h-leg joint must now rotate, also counter-clockwise, to the appropriate critical angle, thus beginning the significant forward movement of the centre of mass.

Step 3: The v-leg now rotates clockwise by twice the critical angle; this will cause the robot to rock forward from one tripodal stance to another, significantly shifting the centre of mass forwards.

Step 4: The h-leg also rotates clockwise by twice the critical angle, and steps 3 and 4 are then repeated *ad inifinitum*, changing direction each iteration.

The following driver functions were implemented in COSMOS/Motion to follow these steps, for v-leg and h-leg joints respectively (example shown for W_h = 150 mm):

Vertical stepping leg:

STEP(TIME,0,0D,2.00,8.62D)
+STEP(TIME,5.00,0D,7.00,-17.23D)
+STEP(TIME,10.00,0D,12.00,17.23D)
+STEP(TIME,15.00,0D,17.00,-17.23D)

Horizontal stepping leg:

STEP(TIME,2.50,0D,4.50,-28.34D)
+STEP(TIME,7.50,0D,9.50,56.68D)
+STEP(TIME,12.50,0D,14.50,-56.68D)
+STEP(TIME,17.50,0D,19.50,56.68D)

Equation 5: Driving Equations for H-leg Width Wh = 150 mm

The parameters represented above were determined from the alpha simulation with the same h-leg width as the beta simulation to be run. The steps followed to produce beta run parameters from alpha run results is outlined below:

1. On the Angles plot (e.g. Figure 71), the elbow point is identified, as shown in Figure 72:

Time	v-angle	h-angle	dh-angle
3.4200E+00	8.1267E+00	-2.5816E+01	-4.8653E-01
3.4400E+00	8.2078E+00	-2.6322E+01	-5.0637E-01
3.4600E+00	8.2891E+00	-2.6843E+01	-5.2083E-01
3.4800E+00	8.3707E+00	-2.7343E+01	-5.0023E-01
3.5000E+00	8.4525E+00	-2.7772E+01	-4.2897E-01
3.5200E+00	8.5345E+00	-2.8103E+01	-3.3064E-01
3.5400E+00	8.6167E+00	-2.8340E+01	-2.3702E-01
3.5600E+00	8.6992E+00	-2.8340E+01	6.2897E-05
3.5800E+00	8.7818E+00	-2.8306E+01	3.3124E-02
3.6000E+00	8.8646E+00	-2.8273E+01	3.3116E-02
3.6200E+00	8.9477E+00	-2.8240E+01	3.3102E-02

Figure 72: "Elbow Point" Determination

a. This has been performed by comparing the difference between consecutive data points in the h-leg's joint angle series.

b. The "elbow point" is taken as the last point before the difference changes sign.

c. For particularly "noisy" data, some discretion may be necessary.

2. The h-leg and v-leg angles at this point are used as the maximum rotation (in either direction) of their respective legs. Parameters for the other four simulations at this h-leg width are created by multiplying the v-leg parameter from the "elbow point" by 0.9, 0.95, 1.05 and 1.1, as shown in Figure 73.

a. The v-leg parameter is used as this is the driven variable in the alpha run simulations; the h-leg angle is the output.

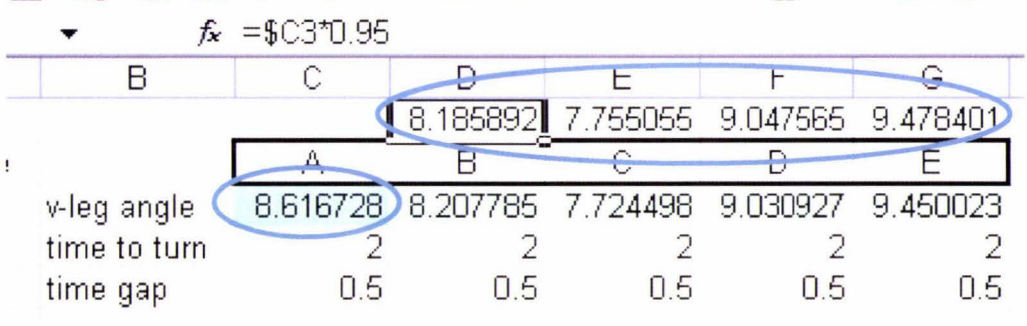

Figure 73: Estimated V-leg Parameters for determining "elbow point" optimality

3. The closest data point "matches" to these projected values is then selected. The projected values are copied from the sheet shown in Figure 73; a manual search is then made, and the corresponding "closest match" is entered in cells highlighted blue in Figure 74:

sim run:	A	B	C	D	E
estimated v-leg angle:		8.18589175	7.75505534	9.04756457	9.47840098
v-leg angle:	8.6167E+00	8.2078E+00	7.7245E+00	9.0309E+00	9.4500E+00
h-leg angle:	-2.8340E+01	-2.6322E+01	-2.3548E+01	-2.8207E+01	-2.8042E+01

Figure 74: "Closest Match" Selection to Provide Parameters surrounding "Elbow Point"

4. These maximum angles are then processed by a very long Excel cell equation, shown in Equation 6, to produce equations (e.g. Equation 5) that can be copy-pasted into the "Joint Angle Driving Equation" field in COSMOS/Motion.

```
="STEP(TIME,0,0D,"&TEXT(C4,"0.00")&","&TEXT(C3,"0.00")&"D)
+STEP(TIME,"&TEXT(2*C5+2*C4,"0.00")&",0D,"
 &TEXT(3*C4+2*C5,"0.00")&","&TEXT(-2*C3, "0.00")&"D)
+STEP(TIME,"&TEXT(4*C5+4*C4,"0.00")&",0D,"
 &TEXT(5*C4+4*C5,"0.00")&","&TEXT(2*C3, "0.00")&"D)
+STEP(TIME,"&TEXT(6*C5+6*C4,"0.00")&",0D,"
 &TEXT(7*C4+6*C5,"0.00")&","&TEXT(-2*C3, "0.00")&"D)"
```

Equation 6: Driving Function Generator Equation (Excel)

The actual parameters derived from the alpha run simulations are shown in Table 13 below. Here, and in other similar tables, the trials have been ordered to represent the spread of input data (minimum values to greatest values), rather than the letter of the trial. As such, the ordering is centered about trial A, and proceeds in order "outwards" from there.

H-leg Width		Trial	V-leg Angle		H-Leg Angle	
100	mm	C	4.6875	degrees	-13.3344	degrees
		B	4.9603	degrees	-14.8860	degrees
		A	5.2388	degrees	-16.1694	degrees
		D	5.5226	degrees	-16.0237	degrees
		E	5.7389	degrees	-15.9136	degrees
120	mm	C	5.2388	degrees	-17.6570	degrees
		B	5.5226	degrees	-19.2200	degrees
		A	5.8200	degrees	-20.5600	degrees
		D	6.1057	degrees	-20.4012	degrees
		E	6.4045	degrees	-20.2679	degrees

150	mm	C	7.7245	degrees	-23.5484	degrees
		B	8.2078	degrees	-26.3219	degrees
		A	8.6167	degrees	-28.3396	degrees
		D	9.0309	degrees	-28.2071	degrees
		E	9.4500	degrees	-28.0420	degrees
170	mm	C	7.9651	degrees	-25.7770	degrees
		B	8.4525	degrees	-28.5309	degrees
		A	8.8646	degrees	-30.4607	degrees
		D	9.2818	degrees	-30.3567	degrees
		E	9.7886	degrees	-30.1730	degrees
200	mm	C	7.1717	degrees	-15.3317	degrees
		B	7.5653	degrees	-16.3440	degrees
		A	7.9651	degrees	-16.5721	degrees
		D	8.3707	degrees	-16.4264	degrees
		E	8.7818	degrees	-16.1864	degrees

Table 13: Beta Run Parameters Derived from Alpha Run Simulations

The most visible metric derived from the beta run simulations is that of "COM Displacement" – the total effective distance the simulated mechanical context traveled under the power of the derived driving functions. The collated results from each of the simulated trials are shown in Table 14, with the addition of a green highlight to show the "winning" trial for each of the geometries.

		COM Displacement		
H-leg Width	Trial	X Axis	Z Axis	Total
100 mm	C	-2.4686 mm	-37.0924 mm	37.1744 mm
	B	20.0000 mm	-40.9518 mm	45.5747 mm
	A	31.9944 mm	33.6150 mm	46.4070 mm
	D	-2.9509 mm	-44.1454 mm	44.2439 mm
	E	-5.0317 mm	-42.3326 mm	42.6306 mm
120 mm	C	-8.2116 mm	-64.1336 mm	64.6572 mm
	B	-7.4146 mm	-65.2653 mm	65.6851 mm
	A	-9.0369 mm	-83.1291 mm	83.6188 mm
	D	-10.0308 mm	-67.5728 mm	68.3133 mm
	E	-15.7582 mm	-73.9392 mm	75.5998 mm
150 mm	C	-31.6513 mm	-99.7430 mm	104.6445 mm
	B	-26.7165 mm	-102.2045 mm	105.6387 mm
	A	-35.0124 mm	-105.5007 mm	111.1587 mm
	D	-43.4029 mm	-100.5937 mm	109.5578 mm
	E	-42.4620 mm	-101.5319 mm	110.0534 mm

170	mm	C	-39.9329	mm	-111.3370	mm	118.2817	mm
		B	-47.3388	mm	-117.3987	mm	126.5836	mm
		A	-43.4790	mm	-124.2407	mm	131.6289	mm
		D	-48.7241	mm	-136.2275	mm	144.6788	mm
		E	-29.1412	mm	-141.2334	mm	144.2085	mm
200	mm	C	0.9145	mm	-60.5022	mm	60.5091	mm
		B	-15.4697	mm	-64.7496	mm	66.5719	mm
		A	-15.5403	mm	-65.1589	mm	66.9865	mm
		D	-7.2435	mm	-66.7522	mm	67.1440	mm
		E	-15.9981	mm	-63.0295	mm	65.0281	mm

Table 14: Beta Run COM Displacements

At shorter H-leg widths, the pattern is as hoped – the "A" trial, based directly on the parameters extracted from the relevant alpha run trial, appears to produce the optimal walking gait for that trial, at least in the tested area about the "elbow" feature. However, as the H-leg width increases from 100 mm, two variations become apparent:

- The "E" trials appear to walk further than the "D" trials, suggesting that viable walking parameters are potentially available further into the super-critical region;
- At H-leg widths approaching 200 mm, the optimum parameters are no longer at the critical point – the "D" trial, located near the beginning of the super-critical region, shows significantly superior performance than both the "A" and "E" trials. This could suggest that results leading to the variation noted in the previous point may be some form of systematic outlier.

These results, while suggesting that the "elbow" feature observed in the alpha run simulations is significant for determining the optimal walking parameters, also indicate that the relationship is likely more complex than initially suspected, and will thus likely require a more detailed study.

One potential explanation is that the optimum operating point is located very near the "elbow" feature, and its distance from the feature is relative to the H-leg width; the Gamma run simulations, and their following Beta-type run, will likely shed light on this relationship, should it exist. This is expected due to gamma run's exploration between the respective influence of relative and absolute dimensions of *both legs* on the derived

parameters, and subsequent walking performance. This exploration should exacerbate any non-stationary relationship between the optimal operating point and the "elbow" feature, allowing it to be identified, and mathematically described.

The longer "E" run strides noted for wide H-legs comes at a price, however – the peak power dissipated by the motor's during their operation is identifiably higher for operating points derived from the super-critical region. Ideally, this would be considered by calculating the work done by the actuators; however the data exported from the simulation tool was too noisy and distorted for any meaningful analysis of this sort to take place. However significant information is still discernable from the produced graphs, and this is presented here as the peak power for each actuation of the vertical leg in Table 15, and the horizontal leg in Table 16.

		Peak Power Dissipation - Vertical Leg			
H-leg Width	Trial	Peak 1	Peak 2	Peak 3	Peak 4
100 mm	C	1.7 W	3.8 W	4.0 W	3.7 W
	B	1.8 W	4.5 W	4.6 W	4.2 W
	A	1.9 W	4.7 W	4.8 W	4.7 W
	D	2.1 W	4.9 W	4.9 W	4.9 W
	E	2.2 W	5.1 W	5.1 W	5.0 W
120 mm	C	2.1 W	4.6 W	4.7 W	4.6 W
	B	2.2 W	5.0 W	4.9 W	4.9 W
	A	2.3 W	5.0 W	5.0 W	5.0 W
	D	2.4 W	4.8 W	4.8 W	4.9 W
	E	2.4 W	5.5 W	5.3 W	5.4 W
150 mm	C	3.3 W	6.1 W	6.1 W	6.1 W
	B	3.4 W	6.2 W	6.2 W	6.2 W
	A	3.6 W	6.3 W	6.6 W	6.5 W
	D	-* W	6.8 W	6.8 W	6.8 W
	E	3.8 W	7.2 W	7.3 W	7.1 W
170 mm	C	3.7 W	6.2 W	6.8 W	6.2 W
	B	3.9 W	6.4 W	6.2 W	6.3 W
	A	4.1 W	6.8 W	6.0 W	6.5 W
	D	4.3 W	6.9 W	7.0 W	6.8 W
	E	4.4 W	7.3 W	7.2 W	7.2 W
200 mm	C	3.5 W	6.1 W	6.1 W	6.1 W
	B	3.6 W	6.3 W	6.4 W	6.4 W
	A	3.8 W	6.7 W	6.6 W	6.7 W
	D	4.0 W	6.9 W	7.0 W	7.0 W
	E	4.1 W	7.4 W	7.4 W	7.4 W

Table 15: Beta Run Peak Power Dissipation - Vertical Leg

H-leg Width	Trial	Peak Power Dissipation - Horizontal Leg							
		Peak 1		Peak 2		Peak 3		Peak 4	
100 mm	C	0.63	W	0.88	W	0.88	W	0.88	W
	B	0.63	W	0.90	W	0.90	W	0.90	W
	A	0.63	W	0.92	W	0.92	W	0.92	W
	D	0.63	W	0.89	W	0.89	W	0.89	W
	E	0.62	W	0.87	W	0.87	W	0.87	W
120 mm	C	0.59	W	1.7	W	-	W	1.7	W
	B	0.61	W	1.9	W	1.9	W	1.9	W
	A	0.62	W	2.1	W	2.6	W	2.1	W
	D	0.60	W	2.0	W	2.1	W	2.1	W
	E	0.59	W	2.1	W	2.5	W	2.1	W
150 mm	C	0.47	W	3.2	W	4.2	W	3.2	W
	B	0.53	W	4.0	W	5.2	W	4.0	W
	A	0.50	W	4.1	W	5.3	W	4.1	W
	D	-	W	4.1	W	5.6	W	4.1	W
	E	0.44	W	4.2	W	5.0	W	4.2	W
170 mm	C	3.4	W	3.9	W	5.9	W	3.9	W
	B	2.2	W	4.4	W	6.5	W	4.4	W
	A	2.5	W	4.8	W	7.1	W	4.8	W
	D	2.3	W	4.9	W	5.8	W	4.9	W
	E	2.7	W	5.0	W	5.9	W	4.9	W
200 mm	C	-	W	-	W	-	W	-	W
	B	1.5	W	2.6	W	3.3	W	2.6	W
	A	1.6	W	2.7	W	3.5	W	2.7	W
	D	1.6	W	2.7	W	3.5	W	2.7	W
	E	1.6	W	2.7	W	3.7	W	2.7	W

Table 16: Beta Run Peak Power Dissipation - Horizontal Leg

* Note. A dash "-" represents those data that were irrecoverable due to noise/distortion in the simulation output

With the exception of the 170 mm trials, the peak power consumed by each actuation generally increases as the operating point moves from the sub-critical to super-critical regions. One observed pattern is that the third and fourth peaks tend to be equal, and the third peak slightly greater – this is likely not due to any particular mechanical explanation, but rather a further quirk of the simulation technique. From observation of the running simulations, because the robot starts from "square" stance, the first step tends to slide sideways somewhat. This tended to set up an alternating pattern of further

sideways movement in the robot's further steps, likely explaining the alternating nature of the "middle" peak.

Note, of course, that the first peak is significantly smaller – this is also resultant of starting from "square" stance, and represents the first "half step" required from each actuator, before the regular walking cycle begins.

Also of note, however, is that the 170 mm geometry appears to be near some form of critical point related to H-leg width: despite stride length and power dissipation steadily increasing for H-leg widths from 100 mm to 170 mm, the trials run with W_h = 200 mm show a considerable drop in both distance covered and power dissipated. The conductance of beta run simulations of the intervening geometries should shed light on whether this is a further "critical point" phenomena, or merely due to an error in the simulator settings or execution of the 200 mm trials.

6.1.3 Other Simulation Runs

Unfortunately, due to time constraints, it was not possible to execute the Gamma, Delta or Epsilon simulation runs. The Gamma, Delta and Epsilon simulations, as well as their intended Beta-style comparative simulations, were intended to explore other geometrical and physical influences on the robot's walking behaviour, and thus provide a considerably more complete "picture" of the design considerations for developing Nv-net controlled walkers of this type.

This mechanical context is a common, simple and robust framework easily controlled by nervous networks, and the completion of these studies will significantly reduce the effort required to develop an operational implementation of a nervous network controlled walker constructed in this pattern.

6.1.4 Summary

The alpha run simulations were successful in identifying significant features in the behaviour of the mechanical context, specifically in the relationship between the v-leg and h-leg joint angles, in the case where the v-leg is driven and the h-leg joint is free to move. It was theorised in section 4 (Methodology) that a relationship would likely be apparent here, due to the ability of the v-leg's movement to influence the h-leg.

The observed relationship is shown to have several qualitative "zones", including a prominent "elbow" feature, identified as a critical point in the robot's behaviour. In order to test the validity of these observations for the development of useful walking parameters, this critical point feature was used as the operating point for the extraction of test parameters for the beta run simulations.

The beta run simulations clearly showed that parameters derived around this critical point phenomenon were useful, and perhaps even optimal for the production of low frequency walking behaviours in the targeted mechanical context. For certain leg topologies, however, the optimal parameter point for walking behaviour was no longer centered on the "elbow" feature, suggesting that the relationship may be somewhat more complex than at first it appears.

Results also suggested that the power consumed by the mechanical context in implementing these walking behaviours was not directly linked with gait optimality; rather it appeared to be a function of leg topology.

6.2 Recommendations

6.2.1 Further Model Development

In order to continue researching this type of Nv-bot, one of two approaches will need to be taken – either a real world robotic platform suitable for research measurements must be constructed, or a more detailed model must be implemented. It is likely that a combination of these two methods will prove the most successful.

In terms of more advanced simulation, not only is a higher level of detail required for the mechanical simulation, the motors, interfaces, and Nv-net controller must also be accurately modeled, and the resulting models linked. This steps into the realm of multiphysics simulation, as it requires mechanical, electromechanical, and both analogue and digital electronics modeling in order to accurately represent the behaviour of the robot's constituent technologies.

One way of implementing this type of simulation would be with a dedicated suite of multiphysics tools, such as ComSol (see http://www.comsol.com for more details). This has the advantage of allowing the researcher to work in a unified environment, and thus provide integrated output of all the results as well.

A more readily available way of implementing this would be to use an interface to allow two existing packages to interact. For example, the National Instruments LabView-Solidworks Mechatronics Toolkit (NI Website, 2007) allows LabView to provide input into a COSMOS/Motion simulation. This would allow a model of the Nv-net controller to be constructed in LabView, and also accurate modeling of the interface circuitry and motor behaviour as well. The motor outputs could then be feed into the joint driving functions in Solidworks to allow for an integrated simulation.

Even with the significant advances in computing technology and simulation techniques in recent years, however, simulations can not perfectly represent the complexity and chaotic nature of the real-world; exacerbating this is the significant simplifications required to produce a model that can practically be simulated upon. Thus the best next step in the development of these conclusions is the implementation of a functioning walker in the real world, and thus the physical verification of the results presented in this thesis.

In order for the results presented here, and those stemming from completion of the other simulation runs, to have significant meaning, they must be tested on a physical implementation of the mechanical context, under the control of a nervous network. Experiments carried out with this real-world implementation will then allow conclusions to be drawn regarding both the accuracy and shortcomings of this study, and produce results that are all the more directly applicable for researchers for having been tested and refined in their actual operating environment.

6.2.2 Further Research Development

In order to encourage further academic research into Nv-networks, their potential, their capabilities, and their computational model/"inner workings", it is necessary for more research to be done in the fundamental areas of network construction and tuning, and general network theory. As these fields become more concretely expressed, it will be more practical to undertake academic research, both in terms of the time taken to build a useful research platform, and also the provision of a stronger theoretical base, and an increased ability to express Nv-bots' operation and behaviour mathematically.

The major area of focus for near-term research into this area must thus belong to theoretical exploration and development. Until the understanding and "quantifiability" of Nv-networks is increased to a much higher level than it is currently, it will be difficult to attract high-profile academic research. Once theory has developed to a level to allow at least limited subjective mathematical modeling of Nv-bots, the potential for quantum

leaps in applications and understanding of artificial neuronal systems becomes a very real prospect.

6.3 Research Application

Currently, there are two primary ways of "programming" Nv-nets: a) altering the characteristics of the neurons and their communications and linkages to each other, and b) altering their linkage to their environment (i.e. the level, sensitivity & type of stimulus received from the environment via sensors, and the level & type of influence the robot exerts on the environment via actuators). Once understanding has been developed regarding the way in which Nv-nets perceives and responds to stimuli representing their environment it will be possible to use a single diverse network in many diverse applications, merely by using a "plug-and-play" approach, and a diverse array of developed interfaces.

Considered in this way, a walking robot is not so very different from a cleaning robot, a swimming robot, a social robot, or even a robot whose environment is the information realm - cyberspace. If the task or problem is of a type the can be mapped onto the behaviour of a particular Nv-net, potentially the only barrier that stands in the way of creating a robot to satisfy the need is the design of an interface that suitably integrates the task/problem into the robot's environment - its survival domain.

By developing a knowledge of the mechanical behaviours of a common robotic platform, this thesis provides progress towards this application, providing researches with a known context in which to explore the mapping of an Nv-net's "world view".

6.4 Conclusion

This thesis set out to examine, via simulation, a typical mechanical context from neuromorphic walking controlled by Nv-nets, with the intent of improving knowledge of the mechanical context, and thus providing a more complete platform for future research into Nervous Networks.

Simulations were devised to investigate the extraction of capable walking parameters from a given robot geometry, and to determine how they changed relative to the leg topology of the robot. Simulations were also developed to evaluate the proficiency of such derived parameters in generating walking behaviours, and produce metrics to allow the comparison of these behaviours.

As a result of these simulations, a critical point phenomenon, observable as an elbow-like feature discussed in section 6.1.1 (Simulation Run Alpha), was identified in the robot's geometry, from which an operating point was extracted. A parameter vector was developed about this operating point, to provide test parameters around the identified feature to determine whether parameters derived from this critical point produced optimal walking patterns.

These parameters were then tested in a series of simulations across a range of robot geometries, and provided significant evidence to suggest that the identified critical point phenomena is useful in producing parameters for near optimal robotic gaits for low frequency walking in the studied mechanical context.

In order to fully validate these parameters, however, these findings will need to be applied to a "real world", Nv-net controlled implementation of the robot, such as that outlined in section 2 (Robotic Platform). Results from such experimentation will provide useful data for further refinement and development of the conclusions of this thesis.

This Masters thesis set out to explore and investigate the influence of various factors in the design of a neuromorphic walker's mechanical context on the robot's walking behaviour. This goal was pursued by identifying multiple factors capable of influencing the walking gait of such a robot, quantifying the influence of several such factors, and providing empirical tools for the development of future implementations of Nv-net powered walkers using this elegant mechanical template.

7 References

Allen, T. J., Quinn, R. D., Bachmann, R. J., Ritzmann, R. E., "Abstracted Biological Principles Applied with Reduced Actuation to Improve Mobility of Legged Vehicles", *Proceedings of the 2003 IEEE/RSJ Conference on Intelligent Robots & Systems*, pp. 1370-1375, October, 2003

Aoi, S., Tsuchiya, K, "Stability Analysis of a Simple Walking Model Driven by an Oscillator With a Phase Reset Using Sensory Feedback", *IEEE Transactions on Robotics*, Vol. 22, No. 2, April, 2006

Bains, Sunny, "Extending Neuromorphic Engineering Beyond Electronics", *Proceedings of Brain Inspired Cognitive Systems*, 2004

Chestnutt, J., Lau, M., Cheung, G., Kuffner, J., Hodgins, J., Kande, T., "Footstep Planning for the Honda ASIMO Humanoid", *Proceedings of the 2005 IEEE International Conference on Robotics & Automation*, April, 2005

Comerford, Richard, "Mecha...what? ", *IEEE Spectrum*, pp 46-49, August, 1994

Corke, Peter I., "Visual Control of Robot Manipulators – A Review", 1994, Available: http://citeseer.ist.psu.edu/cache/papers/cs/6625/ftp:ZSzZSzftp.cat.csiro.auzSzpubzSzpiczSzrcv.pdf/corke94visual.pdf

Cruse, H., "The Functional Sense of Central Oscillations in Walking", *Biological Cybernetics*, Vol. 86, pp. 271-280, 2002

Duysens, J., Clarac, F., Cruse, H., "Load-Regulating Mechanisms in Gait and Posture: Comparative Aspects", *Physiological Reviews*, Vol. 80, No. 1, pp:83-133, January, 2000

Duysens, J., Van de Crommert, H.W.A.A., Smits-Engelsman, B.C.M., Van der Helm, F.C.T., "A walking robot called human: lessons to be learned from neural control of locomotion", *Journal of Biomechanics*, 35, pp. 447-453, 2002

Ferretti, G., Magnani, G., Rocco, P., Vigano, L.., Gritti, M., Rusconi, A., "Object-Oriented Modeling of a Space Robotic Manipulator", *Proceedings of the 8th ESA Workshop on Advanced Space Technologies for Robotics & Automation*, November, 2004

Geng, T., "Fast Biped Walking with a Neuronal Controller & Physical Computation", Doctoral Thesis, Faculty of Human Sciences, Department of Psychology, University of Stirling, April, 2007

Goldenberg, Andrew A., Behabibi, B., Fenton, Robert G., "A Complete Generalized Solution to Inverse Kinematics of Robots", *IEEE Journal of Robotics and Automation*,

Vol. RA-1, No. 1, March 1985

Guralnik, Jack M., "Progressive versus Catastrophic Loss of the Ability to Walk: Implications for the Prevention of Mobility Loss", *Journal of the American Geriatrics Society*, 49:1463-1470, 2001

Haefliger, P. D., "Neuromorphic Electronics – Lecture Notes", *Department of Informatics*, University of Oslo, 2005

Hasslacher, B., Tilden, M. W., "Theoretical Foundations for Nervous Nets and the Design of Living Machines", November, 1995

Hirai, K., Hirose, M., Haikawa, Y, Takenaka, T., "*The* Development of Honda Humanoid Robot", *Proceedings of the 1998 IEEE International Conference on Robotics & Automation*, pp. 1321-1326, May, 1998

Hyrnkiw, D., Tilden, M. W., "Junkbots, Bugbots, and Bots on Wheels: Building Simple Robots with BEAM Technology", McGraw-Hill Professional, ISBN:0072226013, 2002

Ikuta, K., Ishii, H., Nokata, M., "Safety Evaluation Method of Design and Control for Human-Care Robots", *The International Journal of Robotics Research*, Vol. 22, No. 5, May, 2003

Lewis, M. A., Etienne-Cummings, R., Hartmann, M., Xu, Z. R., Cohen, A. H., "An in silico central pattern generator: Silicon oscillator, coupling, entrainment, and physical computation", *Biological Cybernetics*, vol. 88, pp. 137–151, 2003

Liu, J., Tan, M., Zhao, X., "Legged Robots - an Overview", *Transactions of the Institute of Measurement and Control*, pp:185-202, 29, 2, 2007

Long, Jr., J. H., "Biomimetic Robotics: Self-propelled physical models test hypotheses about the mechanics and evolution of swimming vertebrates", Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, pp 1193-1200, Vol. 221, No. 10, 2007

McGeer, T., "Passive Dynamic Walking", The International Journal of Robotics Research, pp:62-82, Vol. 9, No. 2, April, 1990

Mead, C., "Neuromorphic Electronic Systems", *Proceedings of the IEEE*, Vol. 78, No. 10, October, 1990

Moore, K. R., Frigo, J. R., Tilden, M. W., "A Novel Microsatellite Control System", Los Alamos National Laboratory, Feb 1, 1998, available: http://www.osti.gov/energycitations/product.biblio.jsp?osti=642815

Netter, T., Francesschini, N., "Neuromorphic Motion Detection for Robotic Flight

Guidance", *The Neuromorphic Engineer*, Vol. 1, No. 1, pp. 8-12, 2004

NI LabView-Solidworks Mechatronics Toolkit (Alpha Version 01), National Instruments Developer Zone, June 6, 2007, available: http://zone.ni.com/devzone/cda/tut/p/id/6183

Nicoud, J.-D., "Vehicles & Robots for Humanitarian Demining", *Industrial Robot*, Vol. 24, No. 2, 1997, pp. 164-168

Paluksa, D. J., "Design of a Humanoid Biped for Walking Research", Thesis, Massachusetts Institute of Technology (September, 2000), available: http://www.ai.mit.edu/projects/leglab/robots/m2/docs/paluska-masters-m2.pdf

Paulsen, L. D., "Biomimetic Robots", Computer, IEEE Computer Society, pp:48-53, September, 2004

Pratt, J., "Exploiting Inherent Robustness and Natural Dynamics in the Control of Bipedal Walking Robots", PhD thesis, Computer Science Department, Massachusetts Institute of Technology, 2000

Rietman, E. A., et. al., "Analog Computation with rings of quasiperiodic oscillators: the microdynamics of cognition in living machines", *Robotics and Autonomous Systems*, 45, pp 249-263, 2003

Ritger, W., "A Phototropic 2-Motor Walker with Sensors", November, 2002, available: http://www.solarbotics.net/library/circuits/bot_walker_phototrope.html

Rodriguez, N. J., Garzon, J. H., Soriano, J. J., "Applying Scarab Observations in Design of Robots", *Proceedings of the 3rd IEEE Latin American Robotics Symposium*, pp 60-65, 2006

Rodriguez, N. J., Garzon, J. H., Soriano, J. J., "Morphology, Kinematics & Stability of a Scarabaeus as Bases for Design of Robots", *Proceedings of the Electronics, Robotics & Automotive Mechanics Conference*, 2006

Shiller, Z., Dubowsky, S., "Robot Path Planning with Obstacles, Actuator, Gripper & Payload Constraints", *The International Journal of Robotics Research*, Vol. 8, No.6, December, 1989

Smith, C. M., "Review & Identification of DOE Laboratory Technologies for Countermine/UXO Detection", U.S. Department of Energy, 2001
Still, S., Hepp, K., Douglas, R. J., "Neuromorphic Walking Gait Control", *IEEE Transactions on Neural Networks*, Vol. 17, No. 2, March 2006

Tilden, M. W., "Biomorphic Robots as a Persistent Means for Removing Explosive Mines", *Autonomous Vehicles in Mine Countermeasures Symposium*, Naval Post Graduate School, Monterey, CA, April 5, 1995

Tilden, M. W., "Neuromorphic Motion Detection for Robotic Flight Guidance", *The Neuromorphic Engineer*, Vol. 1, No. 1, pp. 12, 2004

Tilden, M. W., Haslacher, B., "The Design of "Living" Biomech Machines: How low can one go?", Physics Division, Los Alamos Research Laboratory, July, 1997

Tilden, M. W., *Personal Correspondence*, September, 2006

Vogelstein, R. J., Mallik, U., Cauwenberghs, G., Culurciello, E., Etienne-Cummings, R., "Saliency-Driven Image Acuity Modulation on a Reconfigurable Silicon Array of Spiking Neurons", *Advances in Neural Information Processing Systems* 16, 2005

Wahde, M., Pettersson, J., "A Brief Review of Bipedal Robotics Research", *Proceedings of the 8th UK Mechatronics Forum International Conference*, pp 480-488, 2002

7.1 Images

Figure 1: Mechatronics Overview http://upload.wikimedia.org/wikipedia/commons/9/99/Mecha.gif