Bio-Inspired Adaption and Navigation of a Quadruped Robot - Final Report

Joseph Ingham

Department of Electronic Engineering, University of York

4th Year Project Final Report for the Degree of MEng in Electronic Engineering

May 2023

Contents

1 Abstract

Quadrupeds are four legged robots that can have either active or passive spines, with a multitude of control schemes. One such control scheme is the use of Central Pattern Generators (CPGs). CPGs are groups of Neural Oscillators that act as biological circuits that produce rhythmic output from non-rhythmic input, they can be used in a robot to control joint angles. Combining these with an active spine introduce the possibility of exploiting the benefits an active spine may provide. The effect of an active spine controlled by CPGs on this project's quadruped (dubbed "TQBot") is unknown. Here it's proven that the spine is able to provide a benefit to the robots stability and speed when it is utilised correctly. Furthermore, parametric sweeps of the CPG are also performed, completing a sensitivity analysis of the CPG network of TQBot, providing necessary data to truly understand TQBot and further its development. These results are able to show that having an active spine is useful for a quadruped robot, and what sort of modifications you can do to the control scheme to produce desired results. This work will lay the foundation for the introduction of machine learning techniques to improve the gaits that TQBot utilises by providing an extensive sensitivity analysis of the robot and its parameters, which can inform future development with upper and lower bounds of parameters that produce valid locomotion.

2 Acknowledgements

I would like to thank my first project supervisor Dr Mark Post, for his constant enthusiasm and endless support always providing knowledge and ideas without hesitation, I also want to thank my second project supervisor Dr Jihong Zhu for his academic support and project encouragement. I would also like to thank Yunlong Lian, who helped define some of my project, and willingly let me work alongside him for his PhD project. Another thank you for my Academic Supervisor Dr Eugene Avrutin who has supported me throughout the entirety of university, making sure that things went smoothly and for providing a kind ear when I needed someone to speak to.

A special thank you to everyone who believed in me throughout my university career.

3 Introduction

Robotic Locomotion is a rich and complex field, there are many different robot forms, with many different control schemes. It is widely considered that cursorial animals (animals adapted specifically to run) possess far greater off-road capabilities than typical tracked or wheeled vehicles [1]. Furthermore, multi-legged robots are deemed to be the most stable in unstable environments [2]. Hence, when developing a robot that you want to walk on natural (unpredictable) terrain, it would make sense to design it with quadruped cursorial animals in mind (like a dog). We can take more from biology than just the form of the robot, we can also use Central Pattern Generators (CPGs) to generate motor torque. There has been interest in CPGs in quadrupeds for quite some time [3] due to their multiple advantages over other systems, which will be discussed later. The overall goal of this research is to investigate the relationship between CPG parameters and gaits, and find the correlation between the two. The project also aimed to produce a way to set stable parameters for desired gaits. One of the main aims that evolved out of the direction of the project was an investigation into the effect of the active and flexible spine on the robots locomotion. The research desires to prove that the spine has a overall positive effect on the robot.

The work was performed alongside the work being completed by University of York's Intelligent Systems and Robotics Group, and a number of PhD students, particularly Yunlong Lian, supervised by Dr Mark Post on TQBot.

The work starts out with a review of work completed in this area already, then discusses the methodology and experimentation completed in the project. The paper then discusses the conclusions of the results, and of the project as a whole. It then discusses possible future work.

4 Literature Review

This section is an updated and expanded version from the Initial Report. The literature review aims to investigate the background behind TQBot. TQBot is a quadruped robot with a tensegrity spine and is controlled by CPGs, which is why these topics were investigated. There is a large focus on CPGs over the other two, because this project focuses mainly on the CPGs. Although there is investigation into the tensegrity spine, this project is more concerned with its effect, as opposed to how it works.

4.1 Quadrupeds

TQBot is a quadruped robot, this means it has four legs, four-legged animals have evolved to counteract the effect of gravity, negotiate terrestrial ground and locomote more efficiently for predation and survival [4].

Furthermore quadruped robots have been studied for quite some time [5] and the research into quadrupeds is only becoming more and more prominent as they provide effective solutions for patrolling/rescuing tasks in hazardous areas since they can overcome obstacles [6].

4.2 Tensegrity

TQBot has a tensegrity spine, meaning its spine is connected without rigid parts. Tensegrity is a structural principle arising from architecture/sculpting, it describes structures that are under "continuous tension [and] discontinuous compression"[7]. Locomotion in tensegrity robots has been researched since at least 2005 [8]. Tensegrity robots have also been controlled by CPGs previously as well [9][10].

Tensegrity spines provide advantages over regular rigid spines due to their inherent flexibility [11].

4.3 Central Pattern Generators

The focus of this research is on CPGs, these will be used to control the locomotion of TQBot. Previous research on a dead cat suggests that locomotion patterns are somewhat controlled by CPGs in the spinal cord [12]. CPGs are biologically inspired circuits that produce rhythmic output from non-rhythmic input [13].

CPGs can be found everywhere in biology, such as walking or breathing [14]. It only makes sense that we would try and exploit these circuits for robotic locomotion.

There are multiple examples of CPG networks for different robotic structures, such as a quadruped [15], a salamander-type robot [16], a fish-type robot [17], a hexapod [18] and more recently an octopod [19].

4.3.1 CPG Architecture

There are also a few different types of CPG architectures [20].

The circuits consist of groups of interlinked neural oscillators (NOs), which have also been studied for the use of quadruped locomotion [21]. These grouped NOs form oscillatory networks that can produce phase angle outputs for groups of motors.

An NO can usually be described as a set of differential equations. For example, an oscillator i described by A.Ijspeert et al [16] is expressed in the equations 1 - 4 below.

$$\dot{\phi}_i = \omega_i + \sum_j (w_{i,j} r_j \sin(\phi_j - \phi_i - \varphi_{i,j})) \tag{1}$$

$$\ddot{r}_i = a_r(\frac{a_r}{4}(R_i - r_i) - \dot{r}_i) \tag{2}$$

$$\ddot{x}_i = a_x(\frac{a_x}{4}(X_i - x_i) - \dot{x}_i) \tag{3}$$

$$\theta_i = x_i + r_i \cos(\phi_i) \tag{4}$$

θ_i is the oscillating set-point taken from the NO, ϕ_i, r_i, and x_i are state variables that represent the phase, the amplitude and the offset of the oscillations respectively. Each state

variable contains a parameter which can be used to determine the desired output. These are w_i, R_i and X_i which determine the frequency, amplitude and offset of the oscillations. The parameters that connect the NOs to one another are $w_{i,j}$ and $\varphi_{i,j}$ which represent the coupling weight and the phase-bias respectively, they describe how oscillator j influences oscillator i. Using these parameters it is possible to alter the behaviour of a system entirely, this complete customisation allows room for lots of optimisation/experimentation.

In the NO networks, connections between said NOs can be "Excitatory" or "Inhibitory".

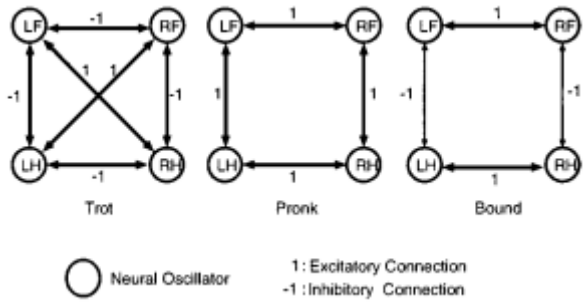


Figure 1: Neural Oscillator Networks for different gaits of a simple quadruped [21]

In Fig. 1 an excitatory connection makes the phase difference between the legs zero, whereas an inhibitory connection makes the phase difference 180 degrees. Having the control over this phase difference between the oscillators can produce wildly different behaviour, and is important to investigate when it comes to designing CPGs for quadruped gaits.

4.3.2 CPG Parameters

The parameters within the NOs also provide an interesting basis for investigation. The difficulty of the investigation comes from determining which parameters to change. Xie et al [22] suggest having four "high-level control command" parameters, these include amplitude, angular velocity, offset, and a time ratio between two phases forming a flapping cycle. However, this paper is discussing a fish-like robot, and so it's worth noting that not every parameter is as particularly relevant to a quadruped robot. For example, a quadruped robot would not have a "flapping cycle", it however would have a stance/swing cycle. Amplitude is also not particularly relevant for a quadruped, where a fin will move further with a higher amplitude thus producing more forward thrust (as it displaces more water), whereas with a quadrupeds leg, having it move further up would not be beneficial as the locomotion comes from the legs contact with the ground. However phase-lag provides a very interesting basis for investigation, the phase lag between NOs can produce wildly different behaviours in the legs of quadrupeds (especially when they have higher DOFs, like TQ-Bot). Phase lag is important because it coordinates all of the joints with one another in a CPG system[23].

This was proven by Liu et al [24], in Fig. 2 they were able to produce four different gait patterns, a walk, a trot, a pace and a bounding pattern, all just by changing the phase difference in the oscillators.

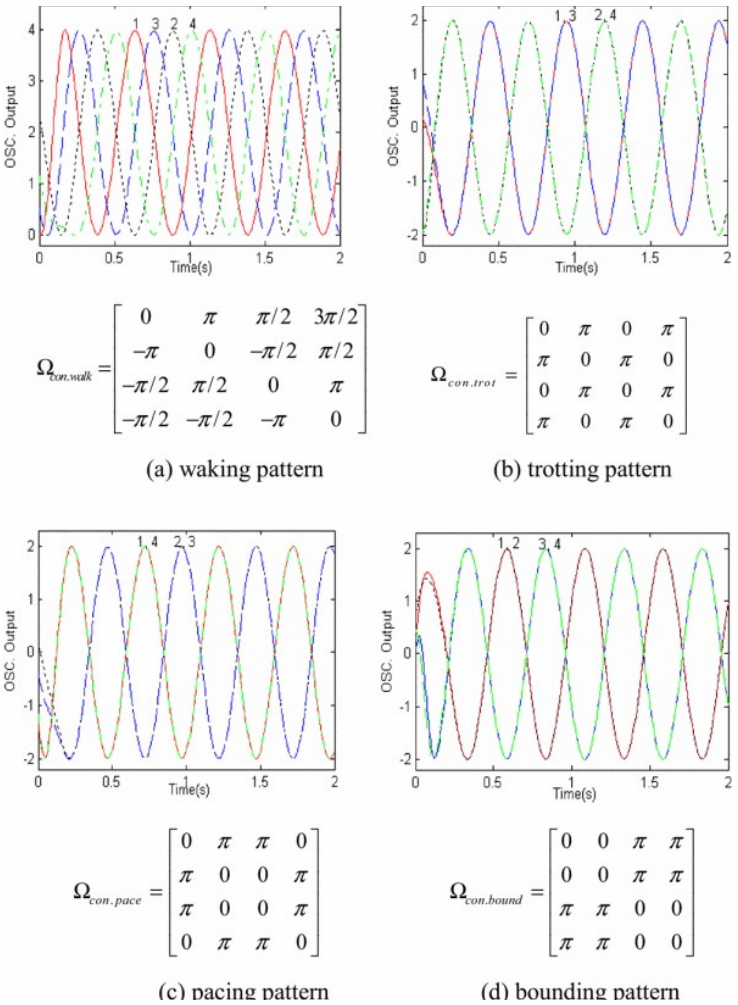


Figure 2: Gait patterns and their phase connection matrices [24]

Frequency is also an interesting parameter because with a higher frequency you can see a linear increase in the movement speed of a robot [25].

Multiple papers agree that CPGs provide an excellent basis for designing robot locomotion controllers [13] [16] [17]. This is for a multitude of reasons; CPG systems are able to exhibit "limit cycle behaviour", meaning that oscillations rapidly return to the steady-state oscillations after any deviations in the system. This is useful because it means that the system can stay relatively stable even when influenced by external sources, making the system quite robust [16].

The previous benefit also leads to another benefit, the parameters of the CPG can be changed at anytime, either abruptly or continuously, and the output will remain smooth. This is incredibly useful for robot locomotion, because it means that parameters can be changed *during* travel to change a robots gait (i.e. from walk to trot), while still maintaining a smooth transition between said gaits, increasing stability and energy-efficiency [16].

The systems also tend to have few control parameters, which lowers the dimensionality of the problem when compared to other controllers. This makes the problem easier to implement physically, as a controller only needs to produce a higher-level control signal rather than a multidimensional motor command [13].

4.3.3 Using sensors with CPGs

Feedback in robot locomotion is important , it provides a controller with the context of its state and allows a robot to react to external stimuli from the environment [26].

L.Righetti et al [15] found that when feedback is included within CPGs, there is a visible improvement in the robots behaviour, speed, *and* stability. This was seen in simulation, however that is fine for the purposes of this research, as most (if not all) of the work will be done inside simulation as well.

Other papers also suggest that sensory feedback plays an important role in robotic locomotion [27], for example it can also be used to coordinate motors [28].

4.4 CPGs Parameter Optimisation

CPG parameter optimisation is a widely researched problem that has been tackled in various ways, such as PSO[29], "the bat-algorithm"[30], and various others [31][32].

Oliveira et Al[31] performed sensitivity analysis of the various parameters and their effect on the gait of the robot. However this was done on a bipedal robot, and so their results may differ than that of a quadruped robot. Their robot also had 20 DOFs (with 6 in each leg) and so their work may differ in that respect as well.

More notable is the research done by Hustig_Schultz et al [33], as this was done on a tensegrity quadruped, it makes it quite suitable to discuss when discussing TQBot (although it differs in the sense that TQBot has a tensegrity spine, whereas "MountainGoat" was all tensegrity and also does not have knees). This research concerns the optimization of the CPG using a neural network, "machine learning techniques involving the Monte Carlo technique as well as genetic evolution for parameter optimization". The research concludes that the learning worked and was able to effectively optimise the CPG parameters. However when looking at the data you can see it took upwards of 30,000 Monte Carlo trials, which is quite a long process, the research being conducted in this project does not have the time to achieve this number of simulations.

Another notable piece of research is more work achieved by Oliveira et Al[34], where they study multi-objective optimization (MOO) for CPG parameters. It is especially relevant to this research because the work is performed on a quadruped. Although their quadruped of choice has 18 DOFs and only 2 DOFs per leg, which differs from TQBot. This work suggests that "Bio-Inspired Evolutionary Computation" is a good choice for gait optimization, as it is good for large dimension MOO problems, it is also model-free, has a low risk of getting stuck in a local minimum and has a strong global search capability[34]. The work focused on having a specific gait generated "a slow crawl", which is important for navigating uneven terrain. They used a genetic algorithm called "NSGA-II"[35], which the research says was effective at optimizing and generating the gait. The work is interesting and shows the possibility of doing MOO with quadrupeds with high DOFs.

Another interesting (and very recent) piece of research has been completed by A.Ijspeert

and G.Bellegarda [36] [37].

Another recent piece of work shows the possibility of using machine learning techniques on central pattern generators. [38]

Overall hand-tuning (whether that be for gait generation or optimization) is not very effective[34], however this research is not particularly interested in the optimization of TQBot's parameters, and more a study of the effect of changing said parameters like the aforementioned sensitivity analyses. However, the literature suggests that the research may continue and start to look at the optimization of TQBot's CPG parameters after this project is completed.

4.5 Measuring Performance of CPGs

To determine how successful a CPG is, the measures of success first need to be determined.

T.Ishii et al [3] suggests using a metric called the "Wide Stability Margin" (WSM). WSM is measured as "the shortest distance from the projected point of the centre of gravity to the edges of the polygon constructed by the projected points of legs"

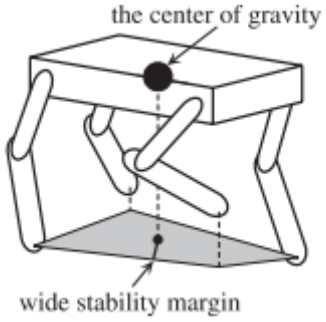


Figure 3: Wide Stability Margin Calculation [3]

The amount this WSM moves as the quadruped travels can be converted into a graph, and compared against other gaits to determine how effective the gait is at keeping stable. Stability is an important measure of performance with TQBot because the spine is free standing (because of the tensegrity), and so it would be interesting to see how it moves with specific gaits.

However, stability may not be the greatest measure of performance, speed may be a bigger factor, or something else entirely. It depends on the context that the robot is being used in. For example in disaster recovery and aid a stable robot would be required, as the ground would be unpredictable.

Another possibility is to look at the energy consumption of the system, this has been done before [22] [39]. Energy consumption is positively correlated with amplitude and frequency, whereas there is no correlation with phase lag [22] because of this energy consumption would be best paired with another measurable and to treat them like an optimisation problem (e.g. lowest energy consumption vs fastest gait). The two examples mentioned measured the energy used when the CPGs were implemented on their hardware, and to the authors knowledge there has not been realistic energy analysis achieved in simulation. Energy analysis may not be suitable for this research, as it will take place primarily in simulation.

Oliveira et Al also did a sensitivity analysis of their quadruped MOO work [34][40].

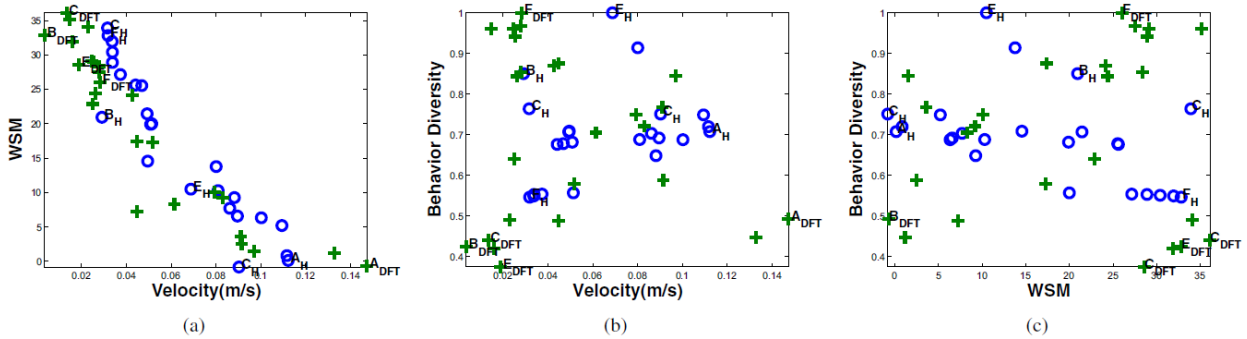


Figure 4: MOO Pareto graph [40]

In Fig. 4, the graph on the left shows that velocity and WSM are hard to optimize together, as they seem to have an opposite effect on each other. This means it would be worth considering which one you want to optimize based on your problem, rather than optimizing for both of them.

Vibration is also a metric that is worth looking at, and is achievable by using acceloremeters in Isaac Sim[31]. Vibration is a useful metric because the lower the vibration, the less stress exerted on the robot.

4.6 TQBot

TQBot is an experimental tensegrity quadruped robot that is currently being developed by the research group. It has 15 degrees of freedom, 3 in its spine (Pitch, Yaw, Roll), 3 in each leg (Hip, Shoulder, Knee). Each degree of freedom has its own oscillator in the CPG.

TQBot's most novel feature is its spine, it is a tensegrity structure, allowing for more flexibility compared to traditional quadrupeds with rigid spines. The spine can be active, or remain passive allowing the research to experiment with both.

Previous work completed suggests that a passive spine solution is more energy efficient and self stable than active ones [41]. However this work was completed on a 2D robot, and so may not necessarily pertain to the results we will see from TQBot.

5 Main Body

5.1 Approach

The research was completed in a logical order, with the gaits to be experimented on being created before any sort of experimentation begun. Yunlong Lian assisted in defining some of the tasks for this project, the author then narrowed down the scope, implemented and gathered results from these tasks. Yunlong Lian aided in defining the tasks, so that this project could work alongside his PhD project. The tools were learnt, they were used to create the gaits, then

they were used to modify and experiment with said gaits, the results from these experiments were then plotted and analysed.

5.2 Setup

To learn Isaac Sim for the research it was important to understand the basics of the GUI and how the API worked, to do this, the tutorials provided incredibly useful background for the topic. The tutorials can be found on NVIDIA's website, and provide ample background into how to use the tool [42]. Any other information that was required was easily found in the API documentation, or even in the Universal Scene Description (USD) API documentation [43].

5.2.1 TQBot Model

The model that the research focused on was created by Yunlong Lian [44], this includes the 3D model and the CPG model itself.

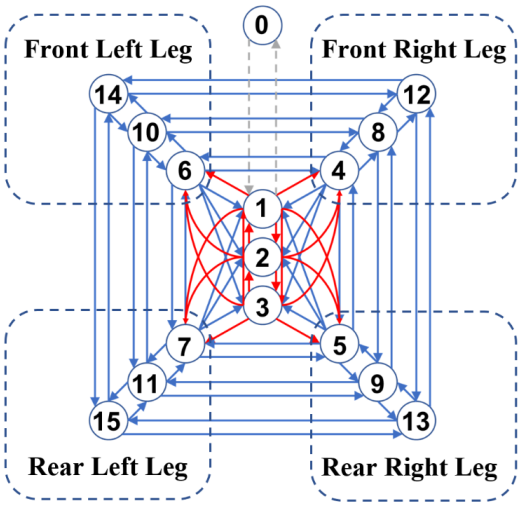


Figure 5: TQBot's CPG Structure [44] [45]

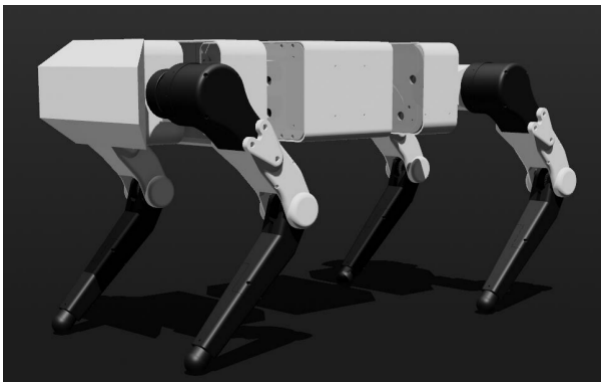


Figure 6: TQBot's 3D model [44]

Fig. 5 displays the layout and the relevant indexes of TQBot's CPG. Oscillator 0 is a reference oscillator and oscillators 1-3 are spine joints (roll, pitch and yaw respectively).

Fig. 6 shows the 3D model that is used to represent TQBot in Isaac Sim, it is constructed from the models that were used to generate the 3D printed parts, this is to make the reality gap as small as possible. More details about TQBot can be found in the conference paper that is mentioned in the Bibliography [45].

It is very important to note that all of the first work was done on a build of Isaac Sim that had a critical bug which affected the friction of the ground plane, meaning that all of the initial analysis was completed on a friction-less build of the robot. However, extra work has been completed so that the research can compare the friction-less results with the results with friction, providing an interesting avenue to investigate the effect of an active spine on a slippery surface, and how it compares with a typical surface.

All of the gaits were checked after the bug was found, and none were found to work in the environment with friction, however it provides a good opportunity to investigate into TQBot's behaviour in a friction-less environment, and how the same gaits would not work in an environment with friction. The gaits were modified to work in the new environment with friction, and experiments were reran to compare the effect of friction on the performance of the robot.

5.3 Gait Generation

Before any of the parameter experimentation could be completed, there first needed to be a baseline to compare them against. These baselines came in the form of gaits/behaviour that were generated for TQBot, they don't need to be perfect examples because as the experimentation occurs the research aims to investigate what affect changing the parameters has on the gait, and so having a middle ground allows for improvement and deterioration of performance which helps garner proper understanding. The plotting tool which displays the output of the CPGs was based on work by Yunlong Lian. The code was modified for this project, and everything was graphed accordingly.

To generate the gaits a spreadsheet was created with each iteration of the gaits being noted down with the parameter changes and their effect. Each gait was iterated through until it was deemed suitable for experimentation, with some being iterated over at least 50 times to produce the desired locomotion. A gait was deemed suitable when it moved a reasonable amount and remained completely upright.

The main difference between the 4 major gaits (walk, trot, pace and bound) is the phase-lag connections as shown by Liu et Al [24]. The other gaits produced were all modified versions of the trot.

The parameters for the fricton-less gaits are in the appendix.

5.3.1 Walk

In Fig. 7, all of the leg oscillators are out of phase with one another, this is the intended effect as traditional quadruped walk gaits move all the legs separately (like in Fig. 2).

This walk is based on work completed by Yunlong Lian, he was able to produce a walk gait, and the project aimed to improve it so that it looked more natural. The original walk dragged

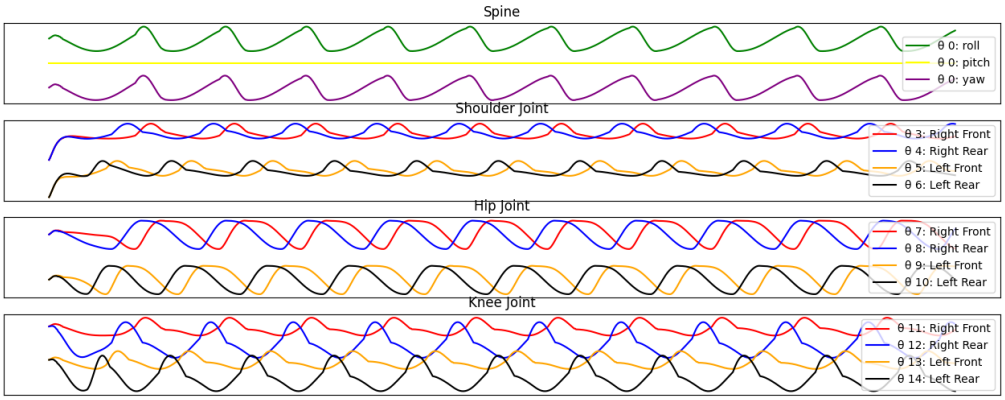


Figure 7: Output for friction-less walk gait

its hind legs when it walked, this was unnatural and also undesired because increased contact with the ground would increase the friction parallel to the ground and slow down TQBot.

5.3.2 Trot

The trot is the most versatile gait used by quadrupeds, they are commonly used as a baseline for most behaviours, the same is true for this project.

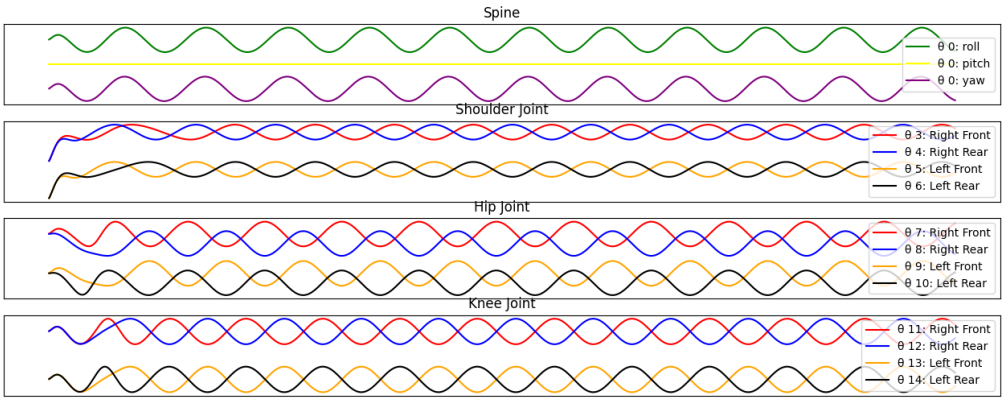


Figure 8: Output for friction-less trot gait

5.3.3 Trotting on the Spot

In Fig. 8 and Fig. 9 the main difference between the trot gait that generates forward movement and the trot gait that doesn't is that for the trot on the spot, there is no shoulder or spine movement. The knee movement is also a higher amplitude and similar side legs move the same.

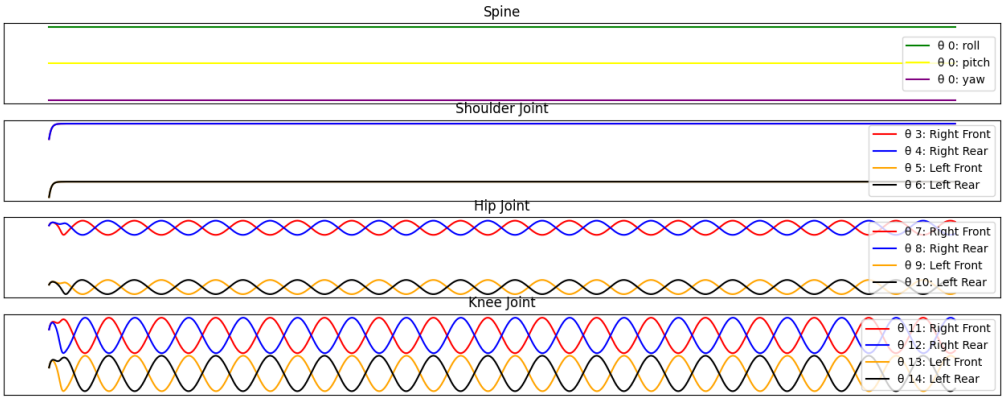


Figure 9: Output for friction-less trotting on the spot behaviour

5.3.4 Turning on the Spot

Although the flexible spine allows TQBot to turn with ease, the research also wanted to show that it could turn on the spot like traditional quadrupeds such as Spot [46]. There was a big consideration of the phase-lag for the shoulders of the robot to achieve the desired behaviour. It was thought that the peak of the shoulder movement should occur at the same time as the peak of the knee movement, so the knee and shoulders had to be in phase with one another. It is shown that they are for this behaviour in Fig. 10

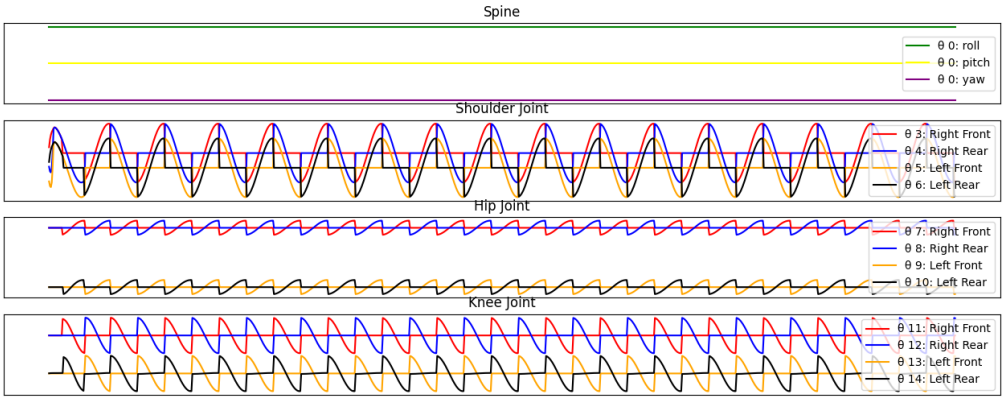


Figure 10: Output for friction-less Turning on the Spot

This produced a gait that turned on the spot, using similar movement and couplings to the trot on the spot.

5.3.5 Pace

The pace gait is a locomotion designed to be faster (although less stable) than the trot. It works by moving the legs on the same side at the same time (i.e. both the left legs are in phase with one another in Fig. 11).

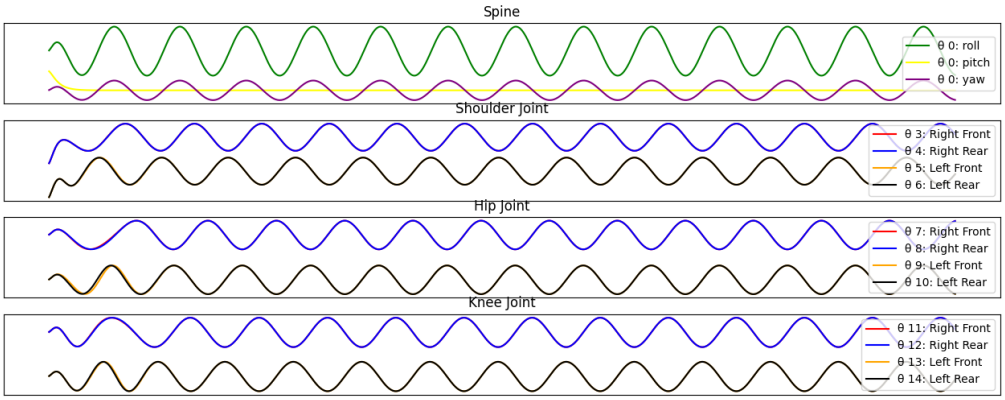


Figure 11: Output for friction-less pace gait

5.3.6 Bound

The bound gait is a locomotion designed to be faster than the pace gait, it works by moving the front legs in tandem, then the back legs in tandem (Fig. 12 shows this).

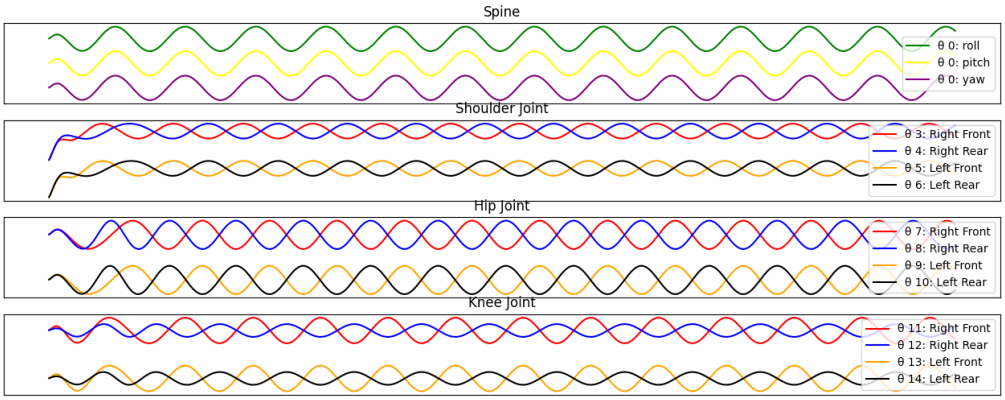


Figure 12: Output for friction-less bound gait

In simulation it is not as effective however, due to simulation inaccuracies and the complications of maintaining high speed with high stability.

5.3.7 Bending the Spine

One of the main advantages of TQBot over traditional quadruped robots with rigid spines, such as Unitree Go1 [47], is the tensegrity spine which provides an additional level of flexibility and dexterity compared to rigid spines. To show that the spine is flexible, and to show that it was incorporated into the CPG, a gait was generated which moved the spine on each of its 3 DOFs, without moving the rest of the body.

This was the simplest behaviour to generate, as it didn't require any sort of timing with the phase-lag matrix, other than to emphasise the movements, this is because the desired amplitudes of each joint in the leg was set to 0, so they don't move.

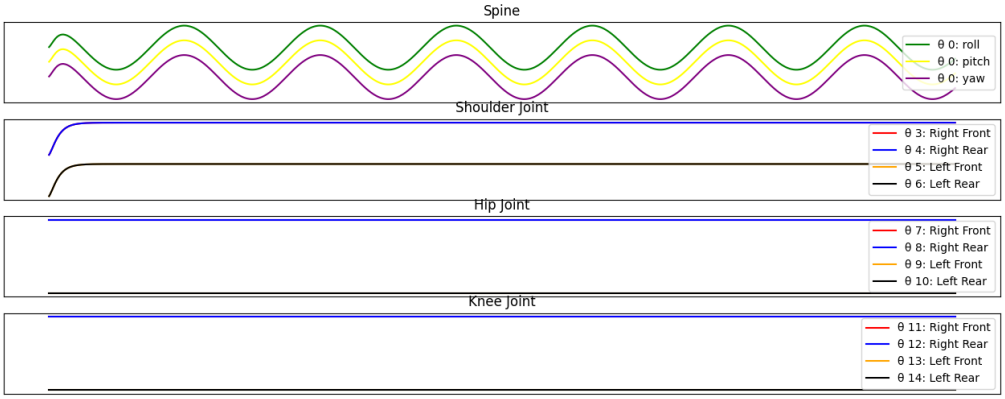


Figure 13: Output for the spine bending behaviour

In Fig. 13, the spine joints are moving independently of any other sort of movement within the robot. This is the same whether the ground has friction or not, as only the spine moves.

5.3.8 Limited Height Movement

There were two gaits generated for the limited height movement, one of which is displayed in Fig. 14, has splayed legs, which decreases its body height but increases its width profile. The splayed legs are described with the increased shoulder joint offsets, which spread the legs out.

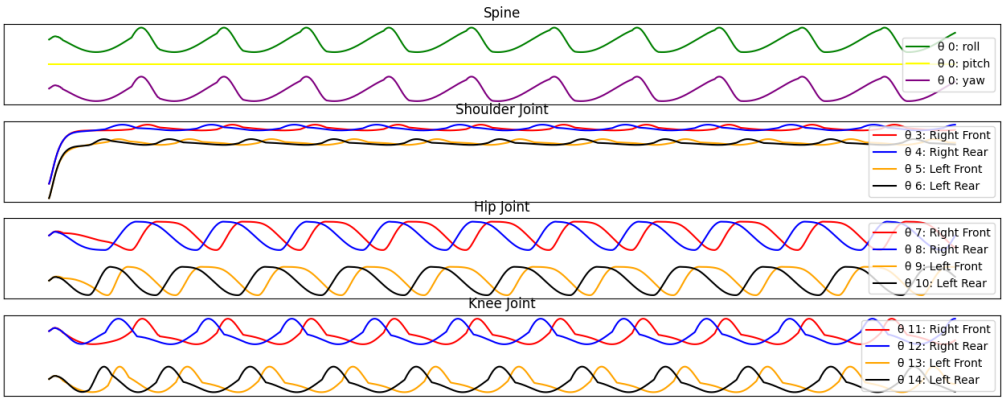


Figure 14: Output for friction-less limited height gait

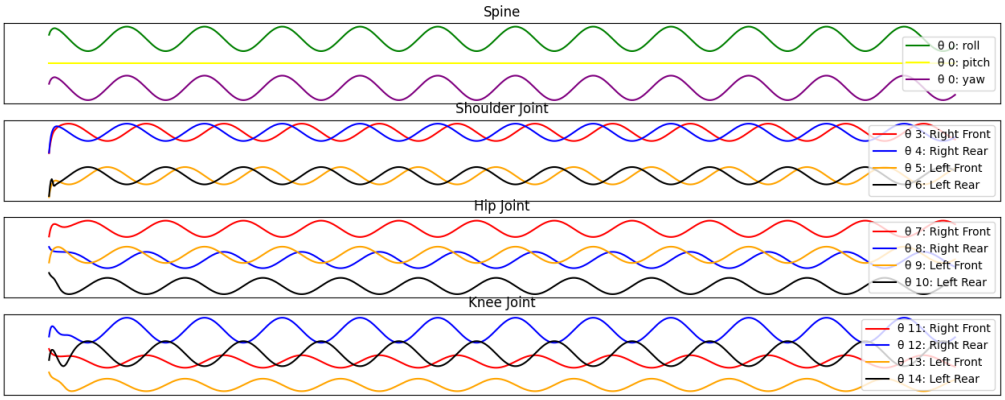


Figure 15: Output for friction-less alternate limited height gait

In Fig. 15, the offsets of the hips and the knees allowed for a lower body stance, while still allowing for relatively good forward movement.

5.3.9 Turning

The additional flexibility provided by the spine allows for TQBot to forgo the traditional method of turning used by quadrupeds [48] (using body trajectory algorithms), and can instead change the yaw of its spine, to direct the forward legs motion. This is what was achieved with TQBot, which allowed it to turn.

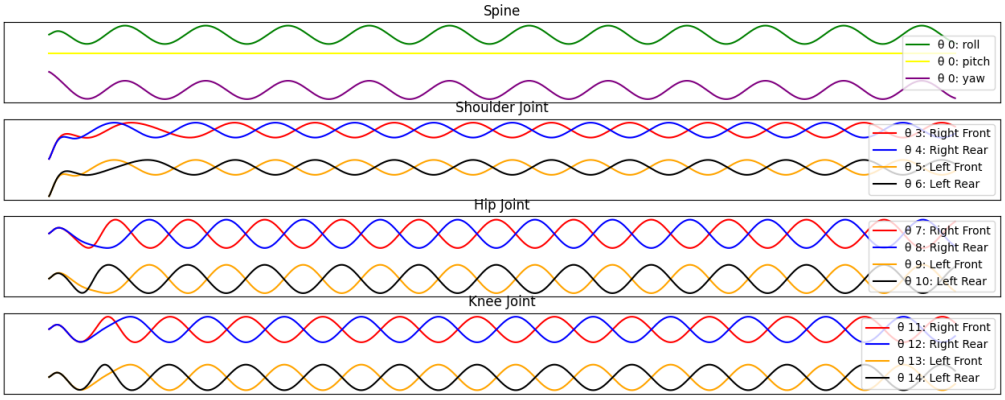


Figure 16: CPG Output for the turning left gait

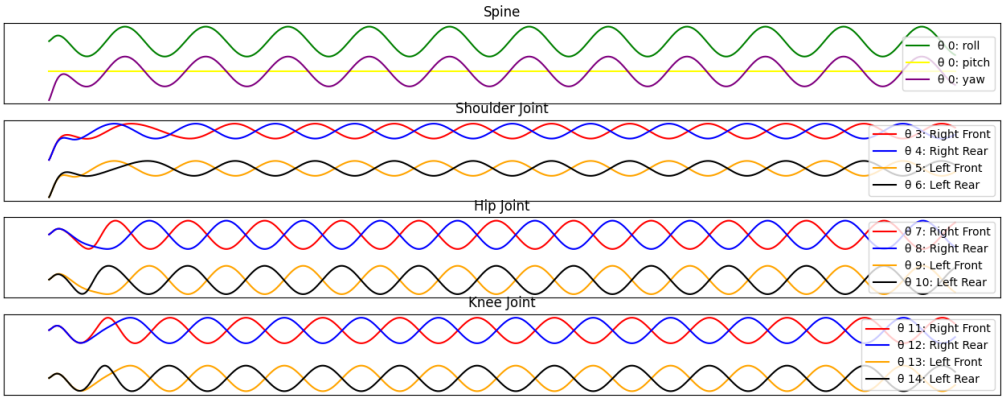


Figure 17: CPG Output for the turning right gait

In Fig. 16 and Fig. 17 the main difference that determines the direction of the turning, is the offset of the spines "Yaw", with a negative offset turning TQBot right and a positive offset turning TQBot left. This can also be seen in the parameters in Fig. 66 and Fig. 67.

The magnitude of the offset determines the turning rate and also affects stability, the details of which are explained in an experiment later in the paper.

5.4 Results Collection Methodology

5.4.1 Results Handling

This sections cover how the results were collected, saved and then plotted.

5.4.1.1 Saving The Results

All of the data was collected during the runtime of each experiment and saved into a CSV file. These CSV files were stored locally, before being uploaded to google drive. All of the CSV files saved during this project are still available in the drive, and have been submitted alongside this report.

5.4.1.2 Plotting The Results

Most of the results for this research were plotted from CSV files using GNUplot. This was chosen because it is free to use, and also because when you run the simulation tool in headless mode (which is how all of the simulations were run), PyPlot is unable to render any of the graphics for the plotting tool, and so you cannot visualise the data. It is also a good experience to learn new tools and skills. The scripts for GNUplot have been handed in alongside this paper.

Some of the results that produced more complex graphs (such as the 3D graphs) were plotted using PyPlot, this is because there is a lot more documentation and help available online for

this tool compared to GNUplot. PyPlot is also a more familiar tool than GNUplot. The code for the results plotting was also handed in alongside this report.

5.5 CPG Parameter Experimentation

All of the experiments were programmed in Python using Isaac Sim's API to control the simulation. All of the experiment code was written and tested by the author, with Yunlong Lian providing the code that moves TQBot. The experiment functions started out with a control block at the start, which determined the parameters of the experiment, such as number of gait cycles or number of steps. Each experiment differs in which parameters it changes and what metric it records.

Most experiments consisted of a number of loops, which iterated through multiple different parameters of TQBot, the general function flowchart is displayed in Fig. 54. A gait cycle loop consisted of running a particular parameter setting multiple times to take an average of the runs. Most of these varied between 5-10 cycles, as it allowed the research to maintain results that relied on average performance while still having short runtimes.

The three main parameters that were focused on were amplitude, offset and frequency. Amplitude determines how far the joints move, offset determines where the center point of said movement is, and frequency determines how fast the oscillation of the movement occurs. Editing these 3 parameters allows for drastic change in the performance and behaviour of TQBot.

5.5.1 Parameter Editing

To investigate the relationship between the parameters and the robots behaviour, it was necessary to make it possible to change the CPG parameters while the simulation was live. This required editing of the code to allow the NOs to be edited, and then allowing the CPGs to also be edited.

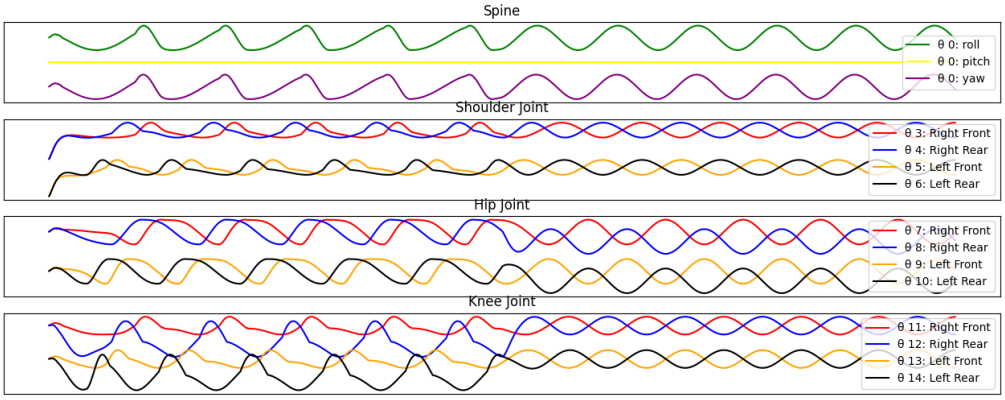


Figure 18: Output for friction-less transitioning between a walk and a trot

In Fig. 18 you can see the smooth transition between the two different cycles for the different

gaits, there is no jerky movement, and shows the limit cycle behaviour discussed earlier in the paper.

It also allows for online parameter change investigation, as well as full gait changes during run-time. This provides useful benefits for controlling TQBot, because it allows you to change its behaviour (such as speed/direction) while it is running.

5.5.2 The Effect of an Active Spine on Gait Stability

As one of TQBot's most unique features is the flexible spine, it was important to characterise and investigate its influence on the performance of TQBot. This experimentation aimed to see whether that including the flexible spine during locomotion would increase the stability of TQBot, determining what the flexible spine's benefit is to TQBot.

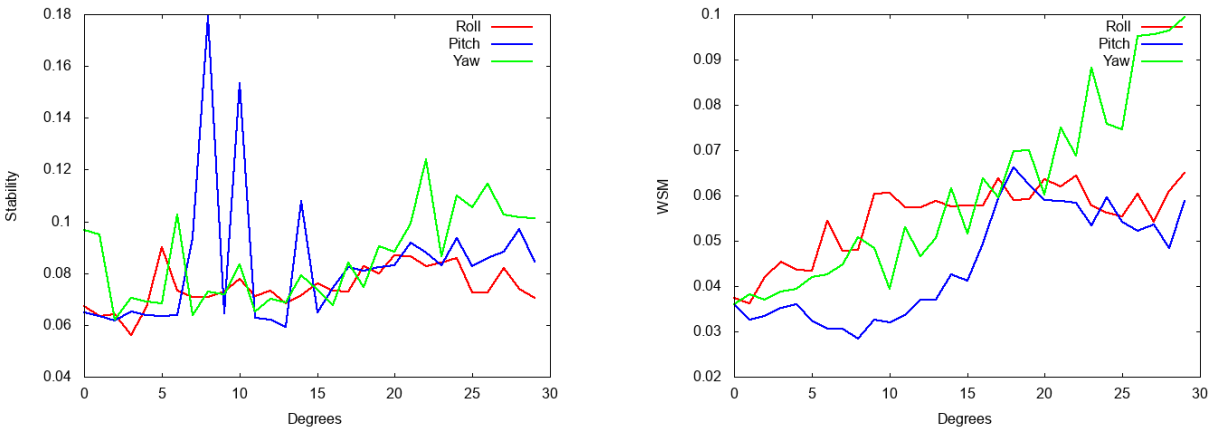
The experiment first consisted of going through each DOF in the fleixble spine one by one, and increasing its maximum turning angle (by increasing the amplitude) and seeing the change in performance.

The experiment was completed with two different gaits, the walk and the trot, this is because they are the most common gait used during the robots operation.

The first metric taken from this experiment was the WSM of TQBot as it ran. These were the first experiments completed in the research.

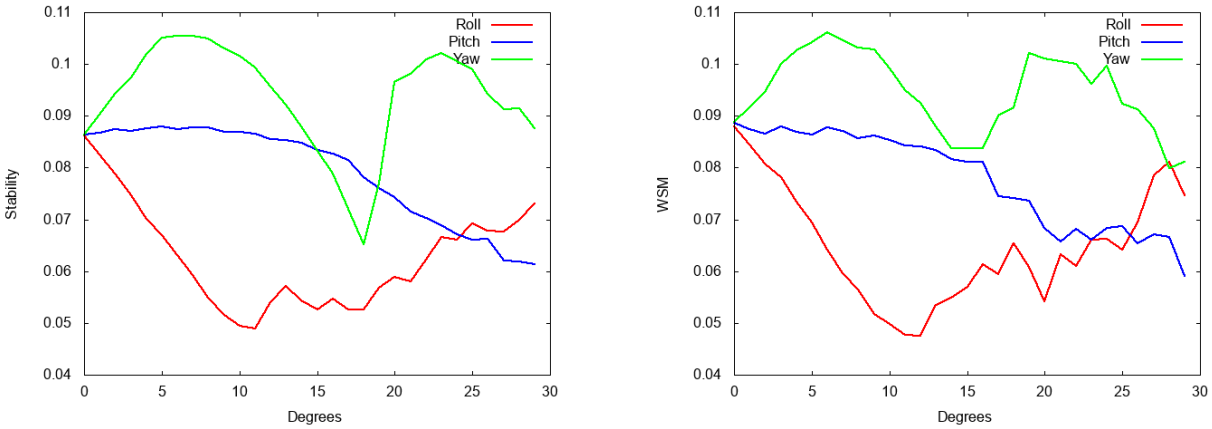
5.5.2.1 Wide Stability Margin

One of the metrics used to measure the performance of the gaits and the effect of parameters was Wide Stability Margin, which was discussed in the literature review. All of the code to calculate the WSM was custom written to work with TQBot and Isaac Sim. First the polygon is created based on the position of the contact meshes on TQBot's feet (which are the contact points with the ground), the centre point of TQBot is found using Isaac Sims inbuilt "`get_world_pose()`" function, this is then projected onto the polygon created earlier. To calculate the closest distance from the point in the polygon to a vertex of the polygon, the research followed the mathematics provided by D. Sunday [49]. The pseudocode to calculate the WSM can be found in the appendix (Fig. 55).



(a) Friction-less (b) Friction

Figure 19: WSM Walking Spine Experiment



(a) Friction-less (b) Friction

Figure 20: WSM Trotting Spine Experiment

Fig. 19a suggests that there is a slight trend upwards in WSM when the angle in the any of the DOFs is higher. Comparing this with Fig. 19b, the WSM is lower when friction is present, but the positive trend is much stronger for the first 15 degrees, and remains strong for the Yaw but not the other DOFs.

Fig. 20a indicates that there is an increase in WSM for the first 6-7 degrees of Yaw, with another increase after around 17 degrees. However for the rest of the DOFs (Pitch and Roll), there was only a decrease in WSM. It makes sense that yaw increases the WSM, as the body turns the distance between the legs will be higher, and so the WSM will be higher. Fig. 20b shows that adding friction doesn't change the effect of the spine on the trots WSM all that much, with similar shapes. This suggests that the trot is a stable gait and also works sufficiently on a slippery surface as well.

5.5.2.2 The Stability Value

While the experiment was running, the stability value was also calculated and recorded. The stability value was devised by Yunlong Lian to measure the stability of TQBot with respect to

the distance it travels, this means that it is quite a useful metric because it also measures the general performance of the robot as well as how stable it is. If the robot travels far and has low movement in the spine, it has a high stability value. If the robot travels a short distance and has lots of movement in the spine, it has a low stability value.

$$d = \sqrt{\Delta x^2 + \Delta y^2} \tag{5}$$

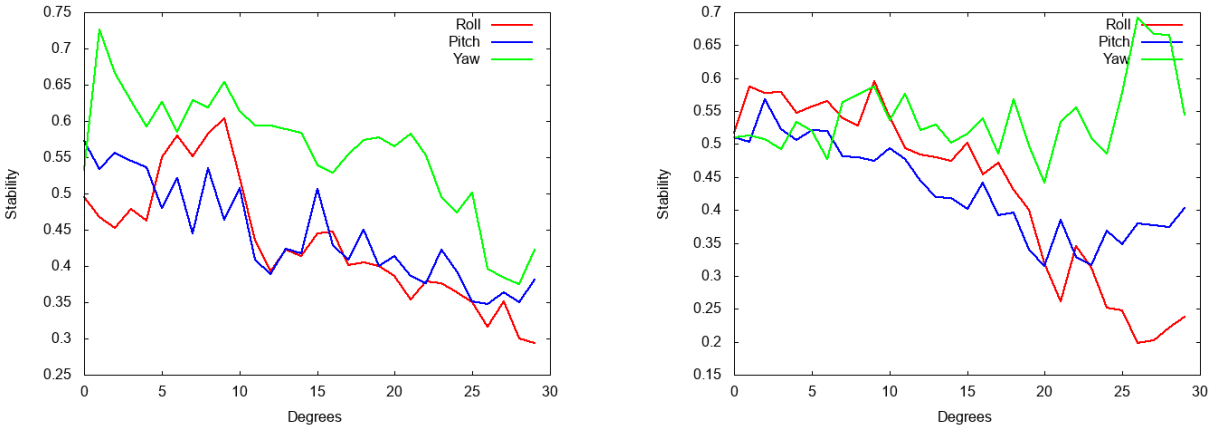
$$\theta(t) = \frac{d}{|\Delta\alpha| + |\Delta\beta|} \tag{6}$$

$$c = \frac{1}{N}\sum_{t=0}^{N}\theta(t) \tag{7}$$

$$F = \frac{2}{1 + e^{-2c}} - 1 \tag{8}$$

Figure 21: The Stability Value [50]

Equations 5 - 8 describe how the stability results are calculated. x and y are Cartesian coordinates. d is the distance travelled, α is the angle of the pitch, β is the angle of the roll, t is the number of iterations in a gait cycle, N is a gait cycle and F is the stability value, because F is calculated using a tanh function, and equation 6 always returns a number larger than 0, the stability value is always larger than 0. The closer the stability value is to 1, the more stable the gait is.

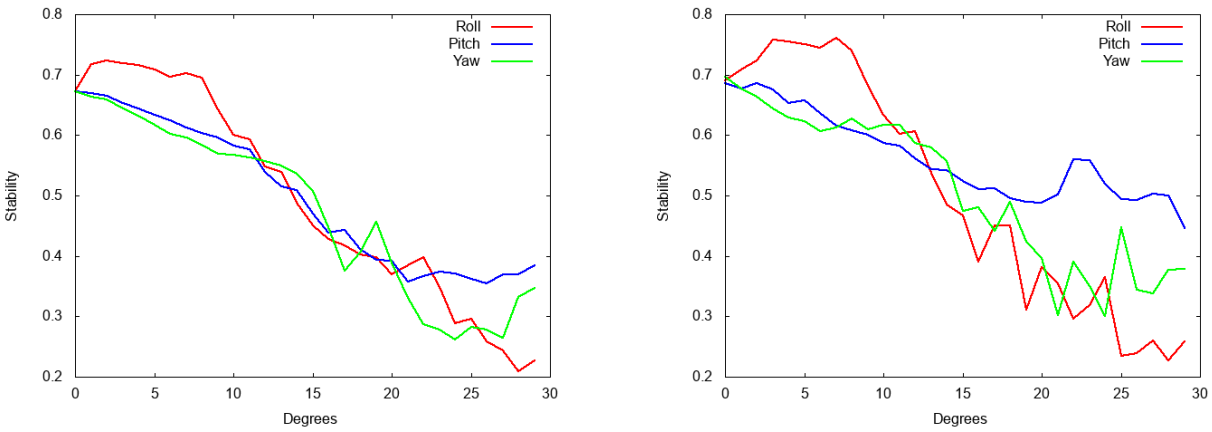


(a) Friction-less

(b) Friction

Figure 22: Stability Walking Spine Experiment

Fig. 22a suggests that using the yaw of the robot increases the stability for values upwards of 20 degrees of movement, roll increases stability between 5-10 degrees and pitch only decreases the stability. Looking at Fig. 22b when friction is introduced, yaw is now more stable towards the higher angles instead, roll is similar when there is no friction and pitch has a similar trend, but is now more stable.



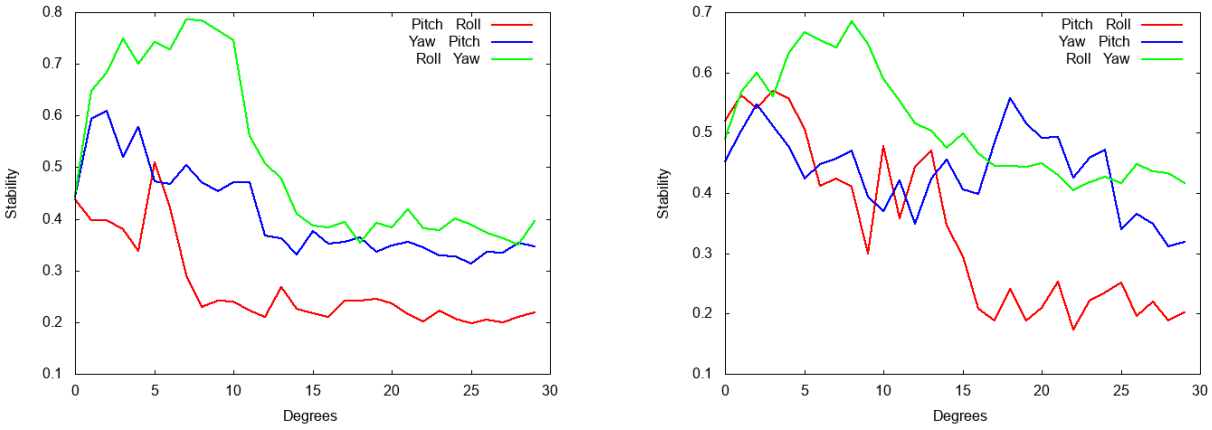
(a) Friction-less (b) Friction

Figure 23: Stability Trotting Spine Experiment

Fig. 23a suggests that roll increases stability between 0-12 degrees, whereas the other DOFs decrease the stability when there is no friction. When there is friction (Fig. 23b), the trends remain similar for each DOF but overall the stability is higher, notably by about 0.5 for the first 10 degrees for the Roll.

These results suggest that different gaits require different spine movements to make them more stable.

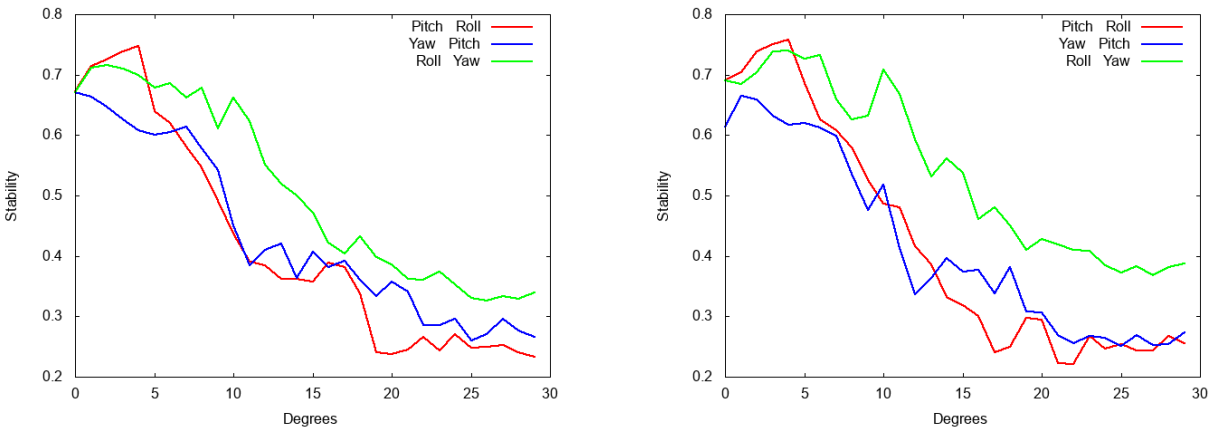
The experiment was then done again controlling 2 DOFs at a time as opposed to just one, the DOFs had the same amplitude applied to each of them. Similar to the previous experiment, the research wanted to see what affect changing two DOFs had on the stability.



(a) Friction-less (b) Friction

Figure 24: Stability Walking Spine Experiment Part 2

Fig. 24a suggests that combining roll and yaw gave a drastic increase in stability between 0-10 degrees, combining yaw and pitch gives an increase in stability between 0-4 degrees and combining pitch and roll reduces the stability. The results with friction (Fig. 24b) suggest that when friction is introduced overall stability is lower for the walk gait, also Roll and Yaw remain the most effective at increasing stability.



(a) Friction-less (b) Friction

Figure 25: Stability Trotting Spine Experiment Part 2

Fig. 25a suggests that there is a slight increase in stability when combining pitch and roll between 0-5 degrees, whereas anything else decreases stability. Similar to part 1 of this experiment, the shape for Fig. 25b is very similar to the friction-less results.

This experiment was then done again, with finer control, looking at the areas of stability increase with smaller step sizes. In essence this experiment allowed the research to take a closer look at areas of increased stability and allowed the research to figure out why the spine was increasing stability. Initial thought suggested that because the stability value was based on speed, that the spine was increasing the speed and not the actual stability.

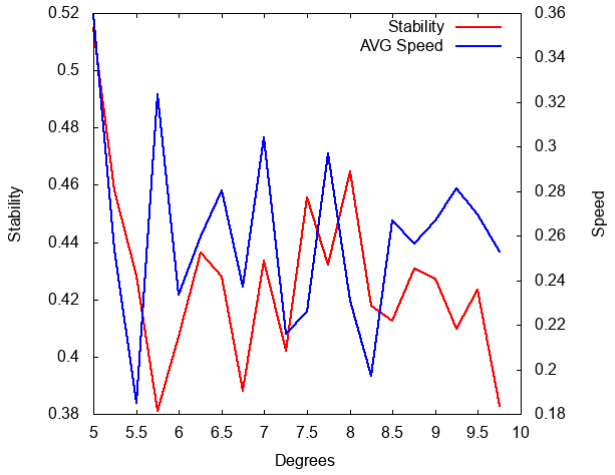


Figure 26: Part 3 - Walk - Roll - friction-less

Fig. 26 suggests that the roll was increasing the stability because the speed was increasing as well. In the middle of the graph the speed and stability follow a similar pattern, increasing and decreasing at similar times.

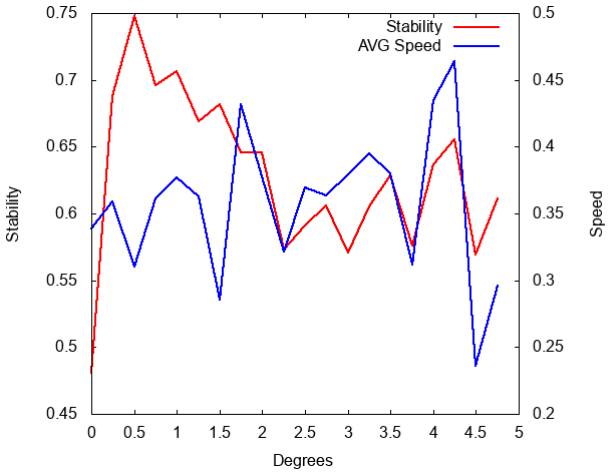


Figure 27: Part 3 - Walk - Yaw - friction-less

Fig. 27 suggests that between 0-1.5 degrees the yaw actually increases the stability as opposed to the speed, whereas between 2.5-5 degrees the yaw was affecting the speed.

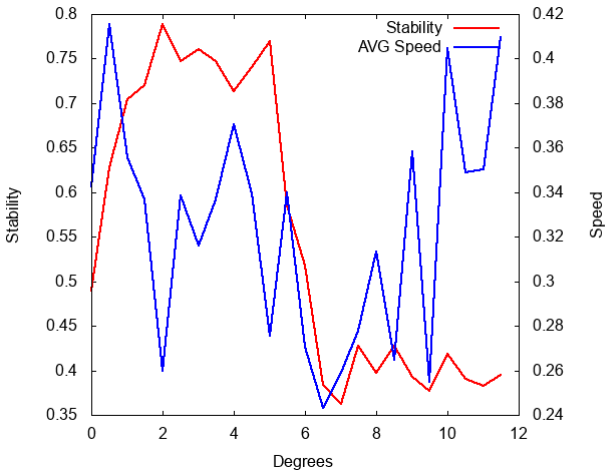


Figure 28: Part 3 - Walk - Roll & Yaw - friction-less

Fig. 28 suggests that when combining roll and yaw they increase stability instead of just increasing the speed, as the speed does not increase when the stability increases.

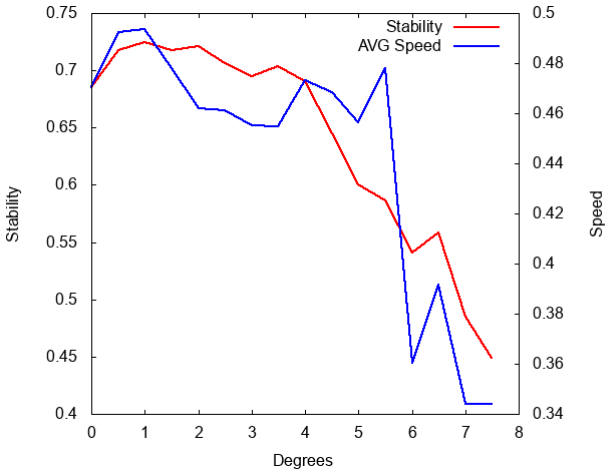


Figure 29: Part 3 - Trot - Roll - friction-less

Fig. 29 suggests that the roll of the spine affects the trotting speed of the robot, which changes the stability as opposed to increasing the stability on its own.

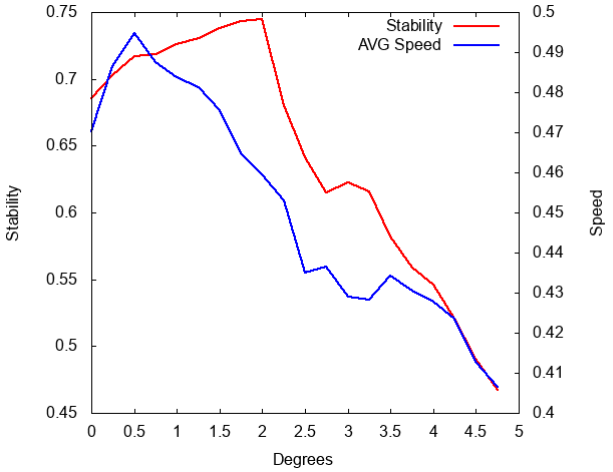


Figure 30: Part 3 - Trot - Roll & Pitch - friction-less

Fig. 30 suggests that, similar to Fig. 29, the roll and the pitch affect the speed of the robot and not the stability.

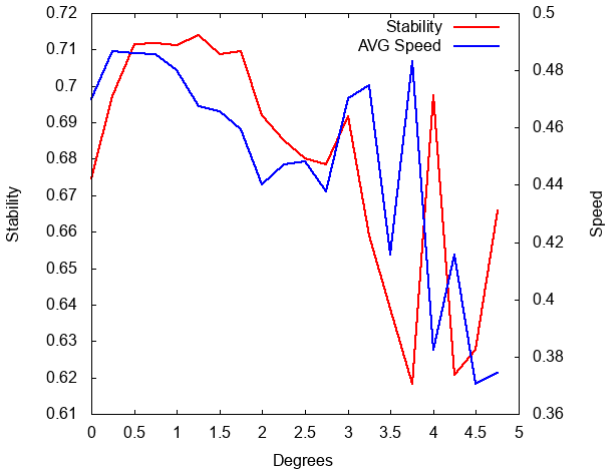


Figure 31: Part 3 - Trot - Roll & Yaw - friction-less

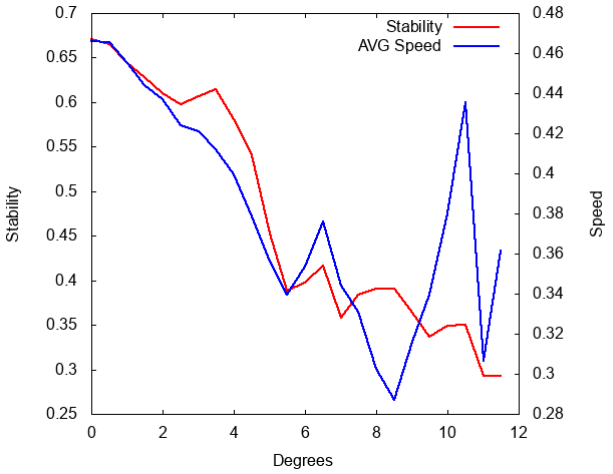
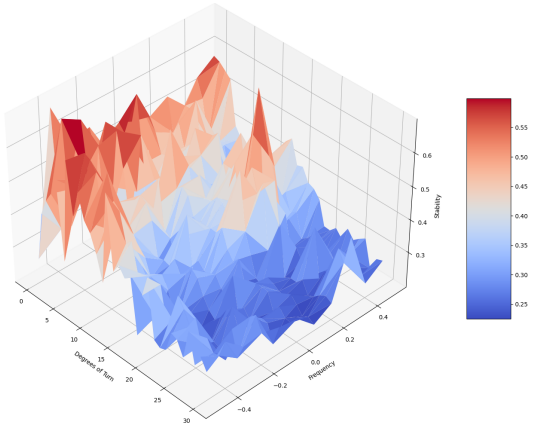


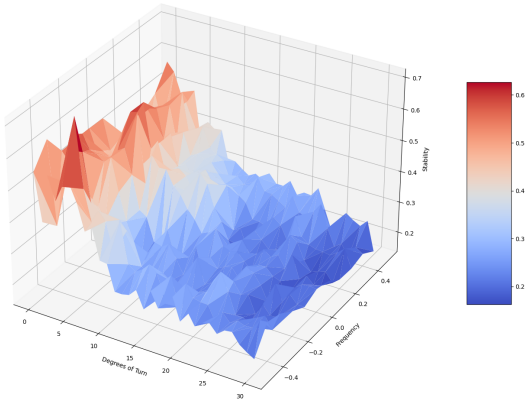
Figure 32: Part 3 - Trot - Yaw & Pitch - friction-less

Fig. 31 suggests that, similar to the previous two graphs, the DOFs here are changing the speed of the robot, which affects the stability, not reducing the amount of unwanted turn in the body. Fig. 32 suggests the same, except the stability is on a downwards trend as the angle increases.

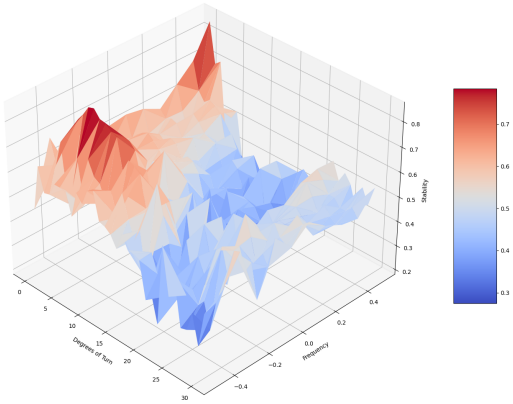
The results led into another experiment, the research wanted to look at the relationship between the frequency of the robots movement, the total spine movement and the stability. The main hypothesis was to suggest that if TQBot is moving faster, would a higher degree of freedom in the spine increase stability? (I.e. is there a correlation between spine movement distance and spine movement speed with the stability of the robot?). Each experiment got 300 points of data for each DOF of the spine, and took approximately 14 hours to run.



(a) Pitch Angle Vs Frequency Vs Stability



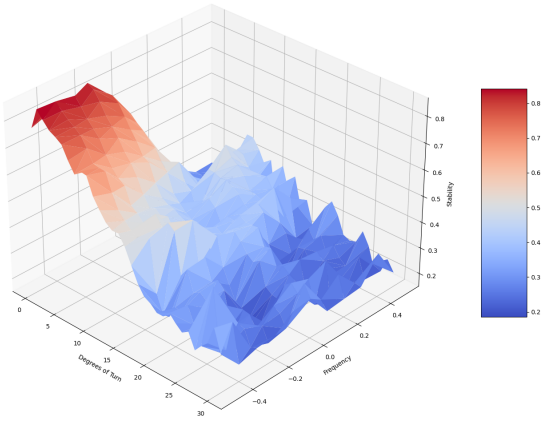
(b) Roll Angle Vs Frequency Vs Stability



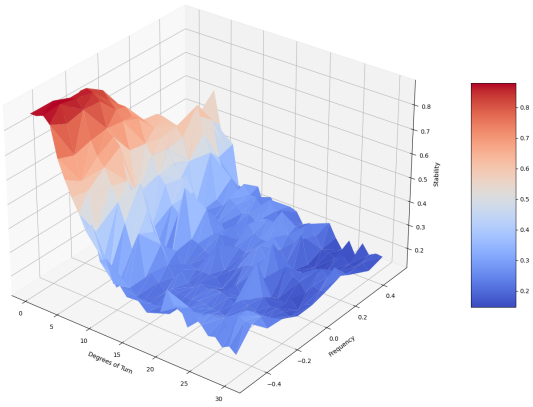
(c) Yaw Angle Vs Frequency Vs Stability

Figure 33: Walk Spine Frequency Experiment Friction-less Results

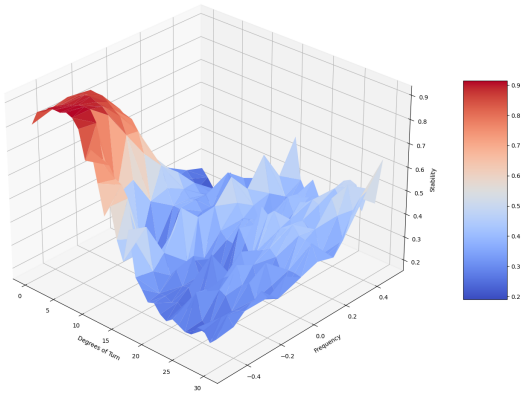
Fig. 33 shows that there is no correlation between angle and frequency for any of the DOFs for the walk gait.



(a) Pitch Angle Vs Frequency Vs Stability



(b) Roll Angle Vs Frequency Vs Stability

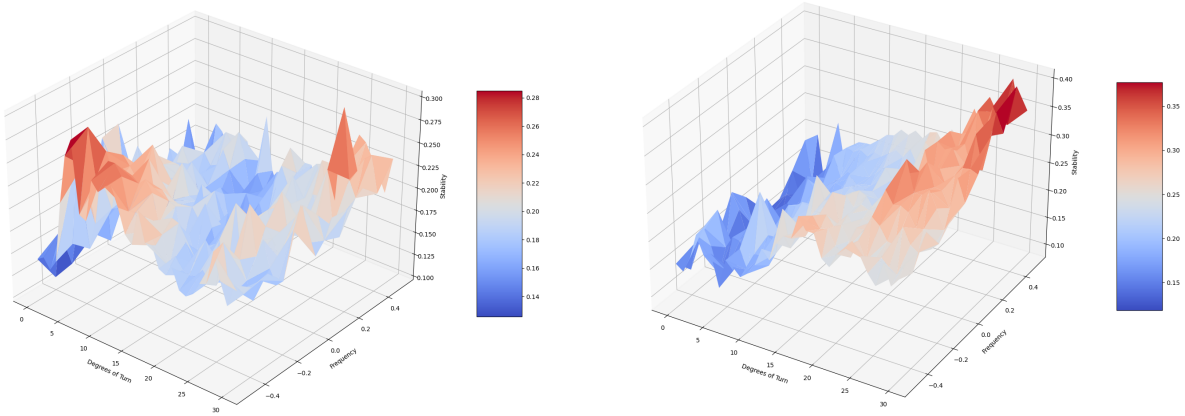


(c) Yaw Angle Vs Frequency Vs Stability

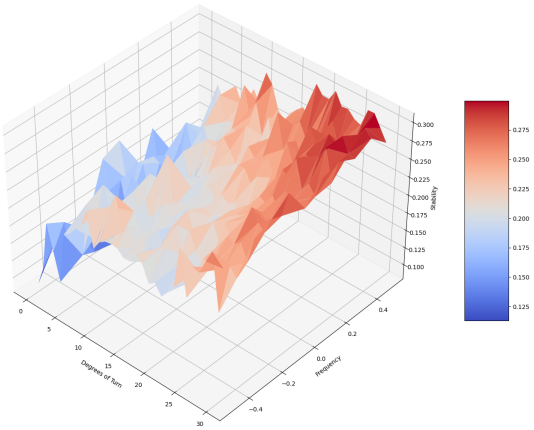
Figure 34: Trot Spine Frequency Experiment Friction-less Results

Fig. 34a however, suggests that for the yaw (Fig. 34c), a higher frequency increases the stability when the angle is above 13 degrees. The rest of the DOFs have no correlation however.

This experiment was then expanded to match the first one, where it was run again, this time pairing DOFs together to see what effect that had. As pairing DOFs in the previous experiment produced different results, it was thought that the same would happen in this experiment.



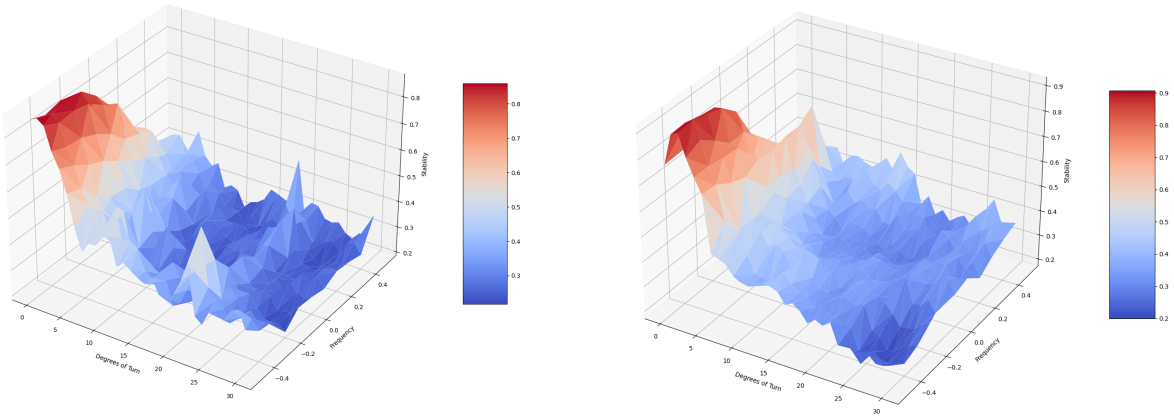
(a) Pitch & Roll Angle Vs Frequency Vs Stability (b) Roll & Yaw Angle Vs Frequency Vs Stability



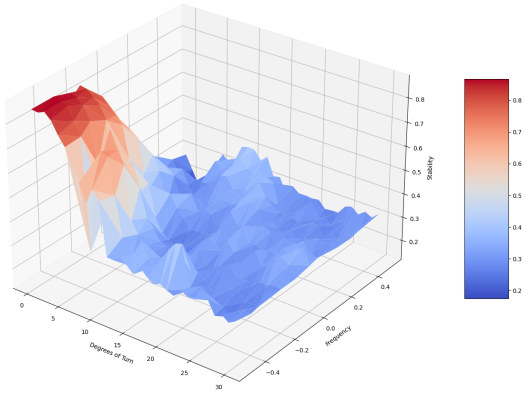
(c) Yaw & Pitch Angle Vs Frequency Vs Stability

Figure 35: Walk Spine Frequency Experiment Friction-less Results

Fig. 35 shows that for pitch and roll (Fig. 35a) there is no correlation, for roll and yaw (Fig. 35b) there seems to be a slight positive correlation between angle and frequency with stability and that with yaw and pitch (Fig. 35c) there is a stronger correlation between increasing the frequency and the angle which increases the stability.



(a) Pitch & Roll Angle Vs Frequency Vs Stability (b) Roll & Yaw Angle Vs Frequency Vs Stability



(c) Yaw & Pitch Angle Vs Frequency Vs Stability

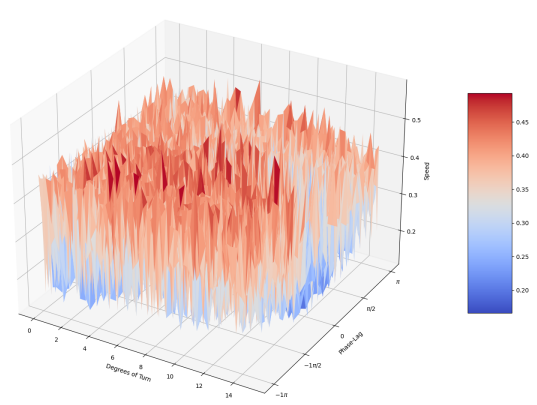
Figure 36: Walk Spine Frequency Experiment Friction-less Results

Fig. 36 shows that there is no correlation between increasing frequency with angle to produce a higher stability in the robot.

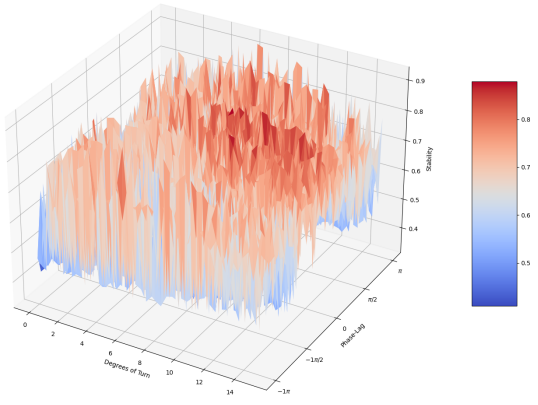
As the research was showing that the active spine wasn't having a positive effect on the stability for the trot gait (or at least that the stability was only increasing because the spine was making the robot move faster). The project also wanted to investigate the effect of changing the phase-lag of the spines DOF that was moving.

Before this experiment was run, the research had to verify that with increasing phase-lags that the spine was still able to couple and able to produce effective gaits. This verification produced 16 different gaits and CGP output graphs (the same as displayed earlier during the discussion of gait generation), these were omitted from the paper, as they do not show anything new. The phase-lag did not affect the robot's ability to locomote effectively, and the CPG still coupled properly. The experiment did not look at the pitch of the spine, because overall it did not contribute to stability when moved.

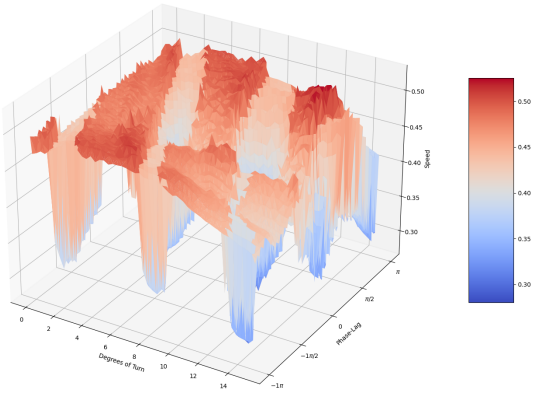
The verification spotted a critical bug in the CPG that meant any changes to the phase-lag of the roll, did not occur. The bug was fixed by including an extra oscillator in the CPG model that acted as a reference oscillator.



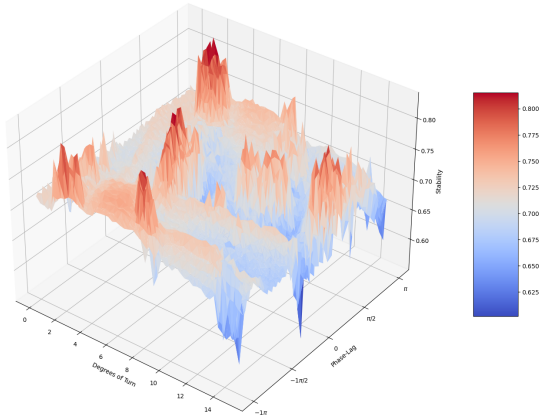
(a) Walk Yaw: Phase vs Angle vs Speed



(b) Walk Yaw: Phase vs Angle vs Stability



(c) Trot Yaw: Phase vs Angle vs Speed



(d) Trot Yaw: Phase vs Angle vs Stability

Figure 37: Phase-Lag vs Spine Angle Friction-less Results

Fig. 37 shows that for the yaw of the trot, there is a clear pattern in the speed of the robot, with large trenches going at a diagonal angle through the graph and looking at the stability there is a clear effect caused by the phase-lag, however it is seemingly random and quite difficult to analyse. We can however use these results to help us increase the stability of the trot gait. For example, the highest point of stability seems to be at a low angle (around 2 degrees) and a phase lag of π.

Looking at the walk results, there is no indication of any sort of pattern caused by the phase lag of the yaw. This further strengthens the idea that different gaits are affected differently by the spine.

This shows that the phase-lag should be taken into consideration when determining how to use the spine.

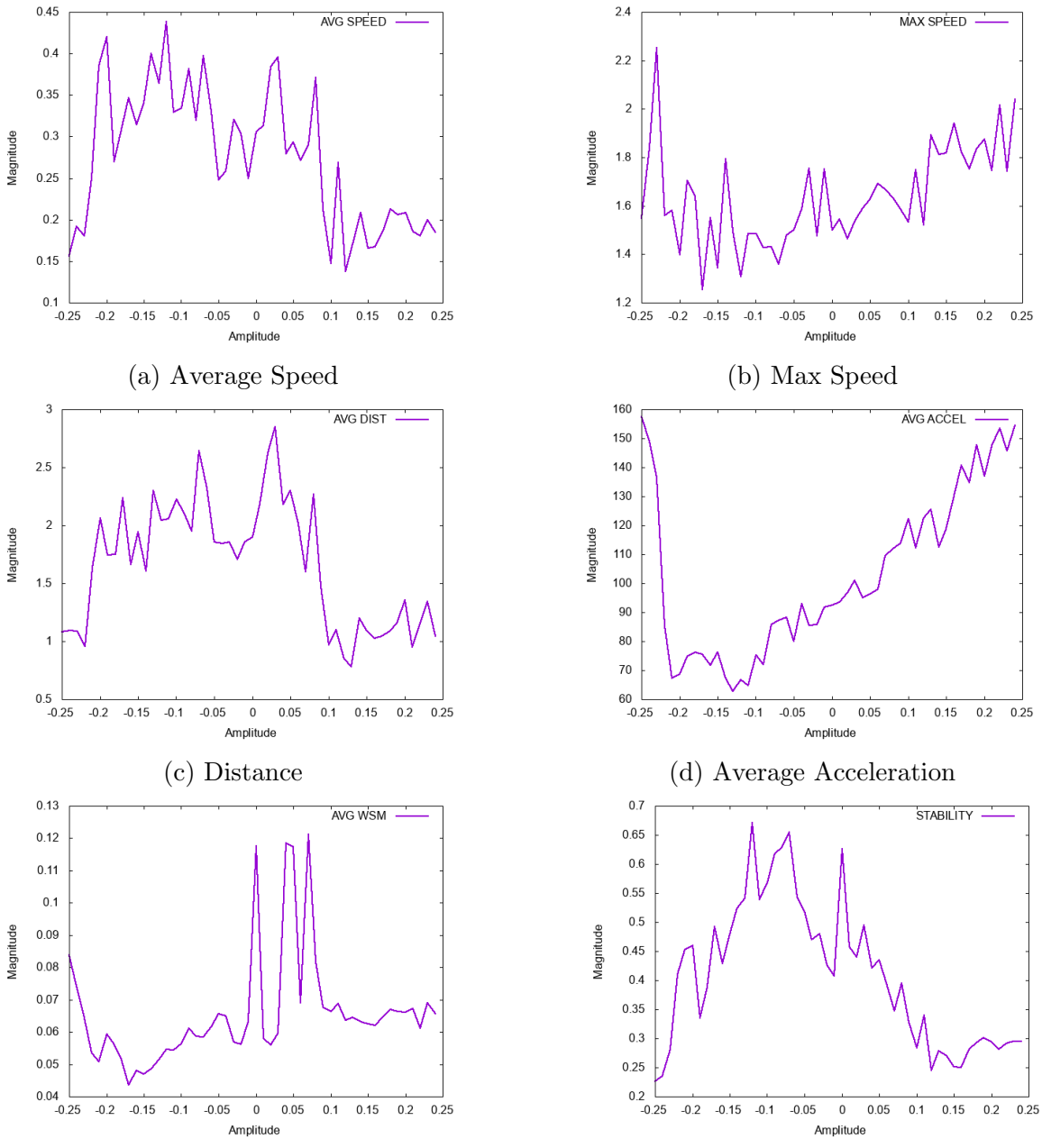
5.5.3 Full Parameter Sweeps

Following on from the previous experiment, not only was it important to characterise the spine's effect, but also the effect of changing the rest of the parameters in the CPG, this formed the basis of the sensitivity analysis of the research. Parametric sweeps were completed, with the center value being the default value of the current gait as described in the appendix. The first sweep was a full body sweep, where each oscillator was swept over, including the spine.

When moving it is important that the robot remains above a certain stability, to ensure that it doesn't fall over, and remains as upright as possible with minimum pertubation. With deeper investigation it could be possible to explore what sensitivity value is required at a minimum.

The pseudocode for a generic sweep can be found in the appendix Fig. 56, the relevant joints change depending on the type of sweep.

5.5.3.1 Walk - Amplitude



(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

(e) Wide Stability Margin

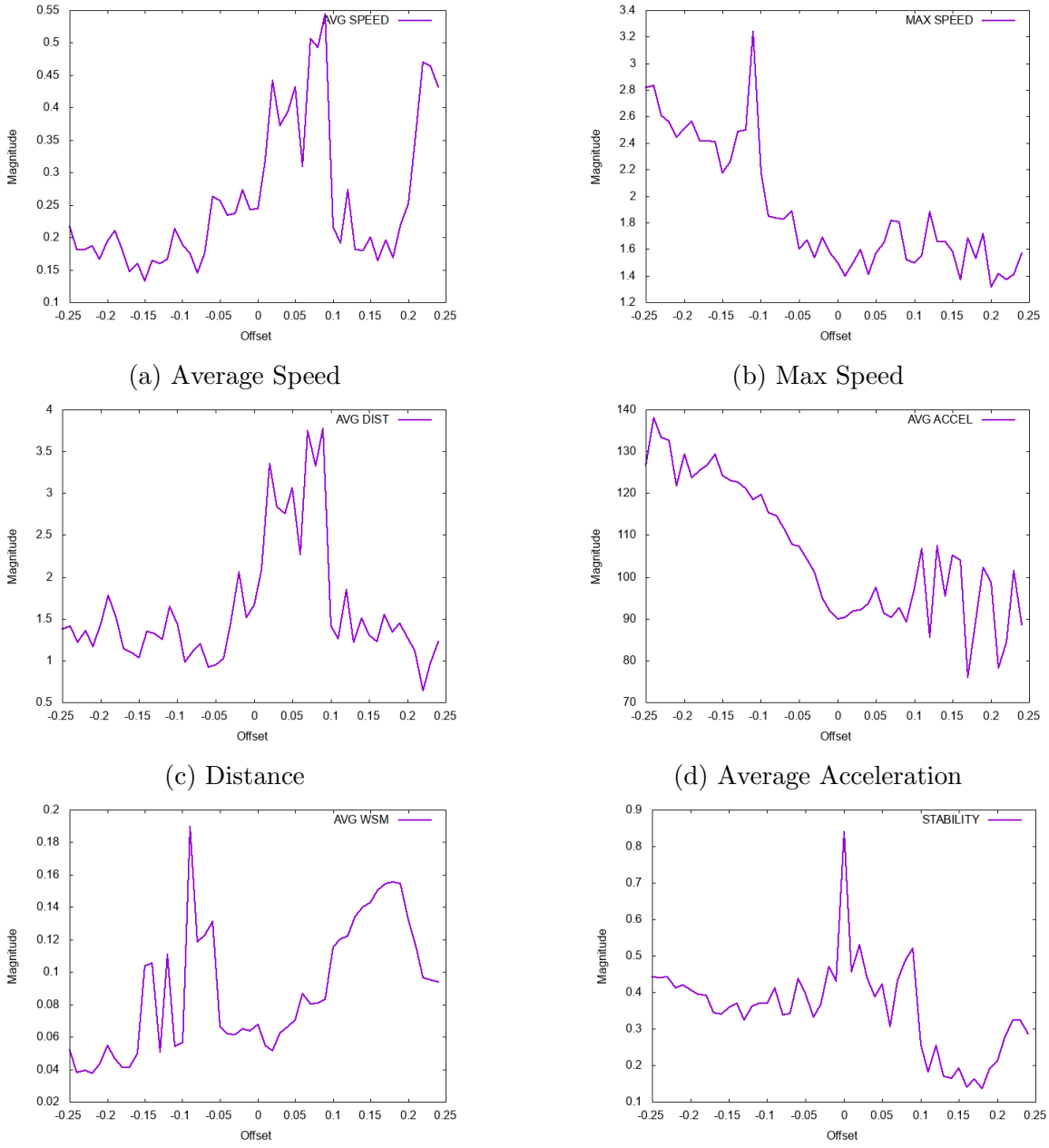
(f) Stability

Figure 38: The effect of an amplitude full sweep on a friction-less walk gait

Fig. 38 shows that there is a strong correlation between increasing the amplitude of every joint and the acceleration of the robot when it is in a walk gait. However this acceleration seems to

be a detriment because at those increased amplitudes, the distance and the stability are much lower, which suggests that the robot fell over.

5.5.3.2 Walk - Offset



(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

(e) Wide Stability Margin

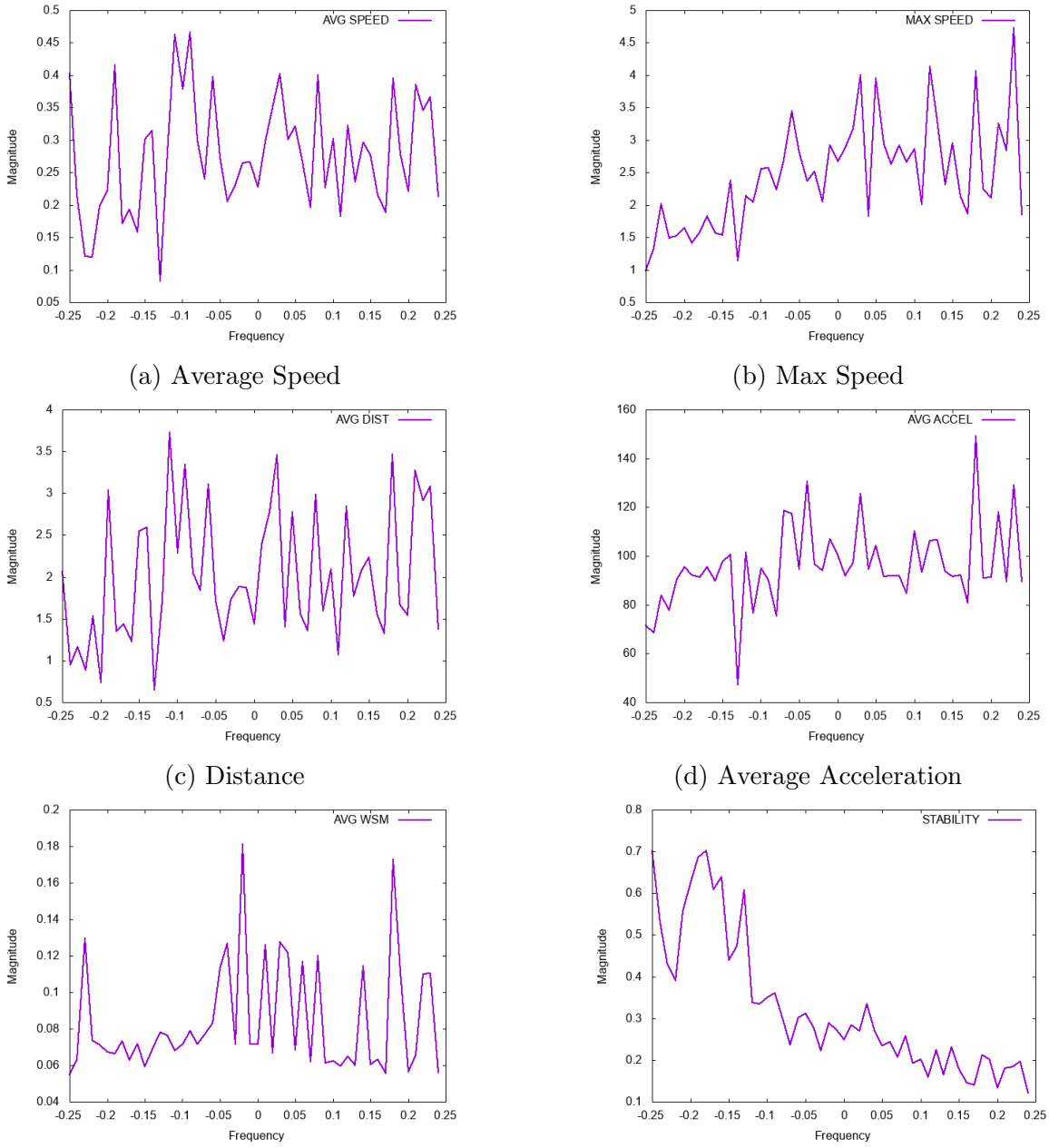
(f) Stability

Figure 39: The effect of an offset full sweep on a friction-less walk gait

Fig. 39 shows that there is room for improvement in the gait, as at around 0.08 higher offset of every oscillator, the speed increases drastically, while also maintaining a good stability and WSM. This shows that there is valid reason to investigate the gaits and ways to improve them, possibly with machine learning. There is also correlation between higher offsets and higher WSM, this is expected because as the shoulder joints have higher offset, the legs will be

more splayed and therefore further away from one another. There is also a negative trend in acceleration when offset is increased.

5.5.3.3 Walk - Frequency



(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

(e) Wide Stability Margin

(f) Stability

Figure 40: The effect of a frequency full sweep on a friction-less walk gait

Fig. 40 shows that there is a correlation between a higher frequency producing a lower stability. There is a slight positive trend in maximum speed with a higher frequency. There is no correlation between any of the other metrics however, as they don't have a trend and just sporadically increase/decrease.

5.5.3.4 Trot - Amplitude

(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

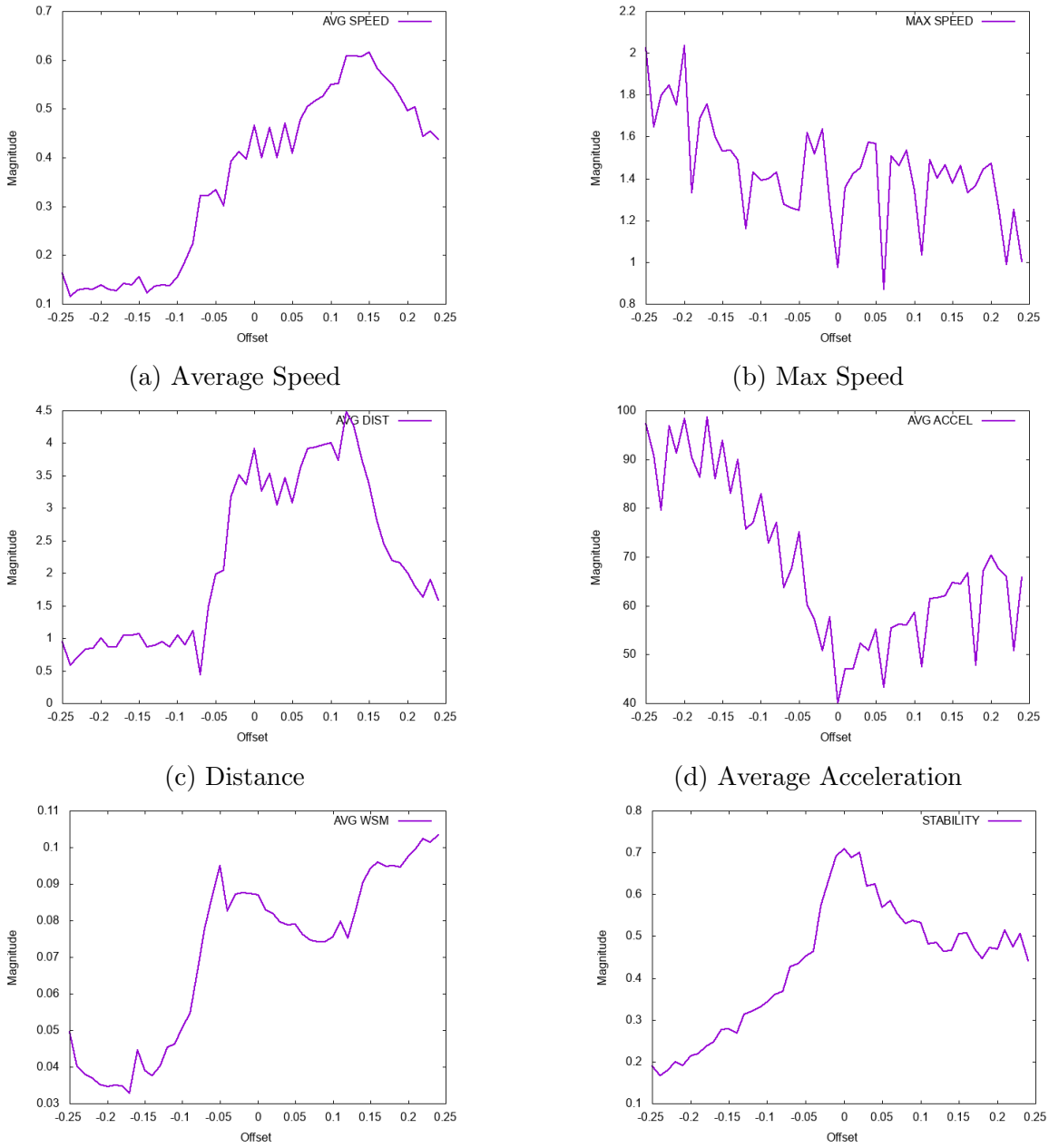
(e) Wide Stability Margin

(f) Stability

Figure 41: The effect of an amplitude full sweep on a friction-less trot gait

Fig. 41 shows an increase in amplitude for the trot gait decreases WSM and stability, while slightly increasing acceleration, similar to the walk gait.

5.5.3.5 Trot - Offset



(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

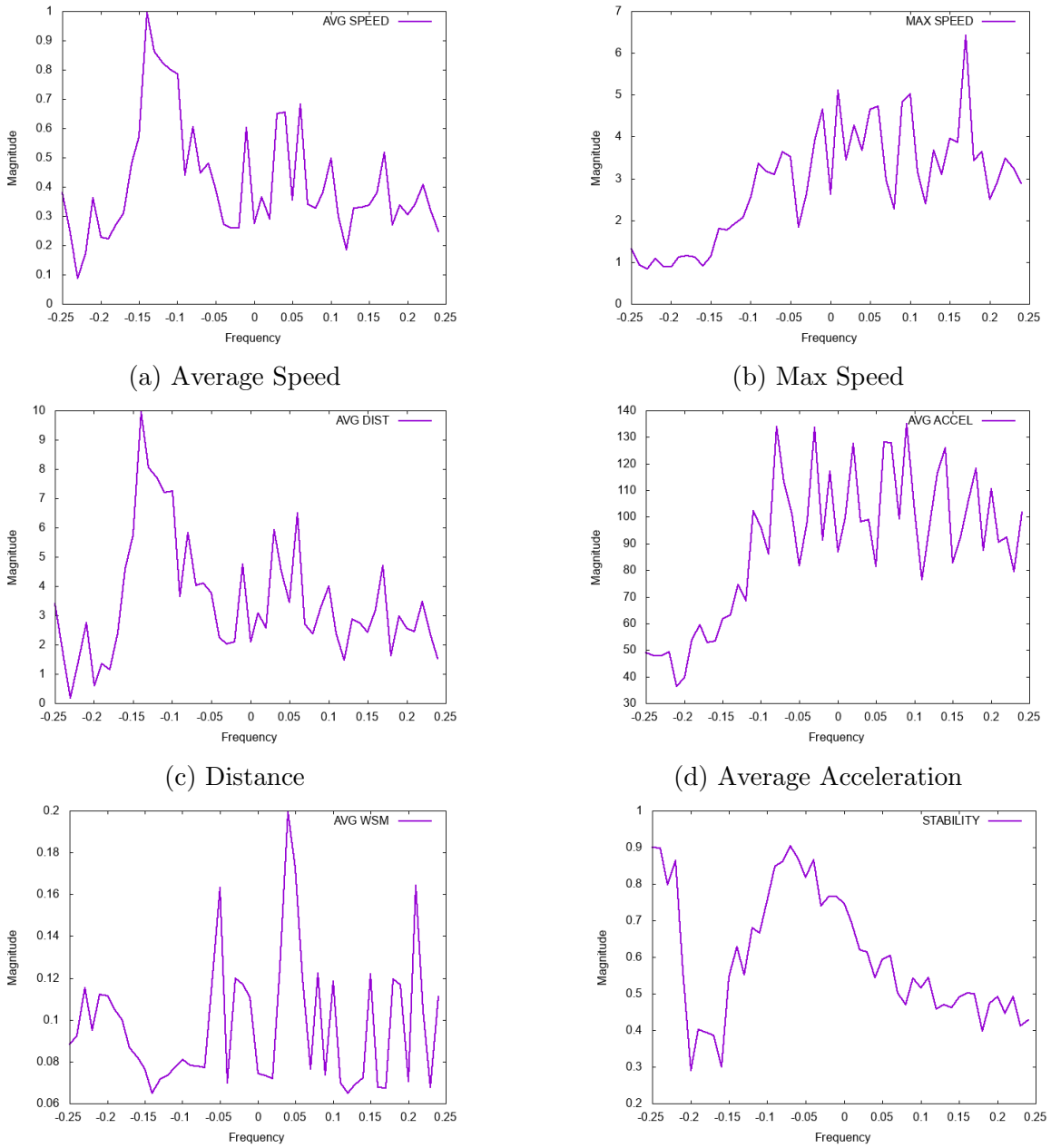
(e) Wide Stability Margin

(f) Stability

Figure 42: The effect of an offset full sweep on a friction-less trot gait

Fig. 42 shows that with an increase in offset there is an increase in WSM as well. It also seems to increase the average speed up to values of around +0.15 of the base value.

5.5.3.6 Trot - Frequency



(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

(e) Wide Stability Margin

(f) Stability

Figure 43: The effect of a frequency full sweep on a friction-less trot gait

Fig. 43 shows that a higher frequency reduces stability for the trot gait. There seems to be an ever so slightly downwards trend for the distance travelled as well.

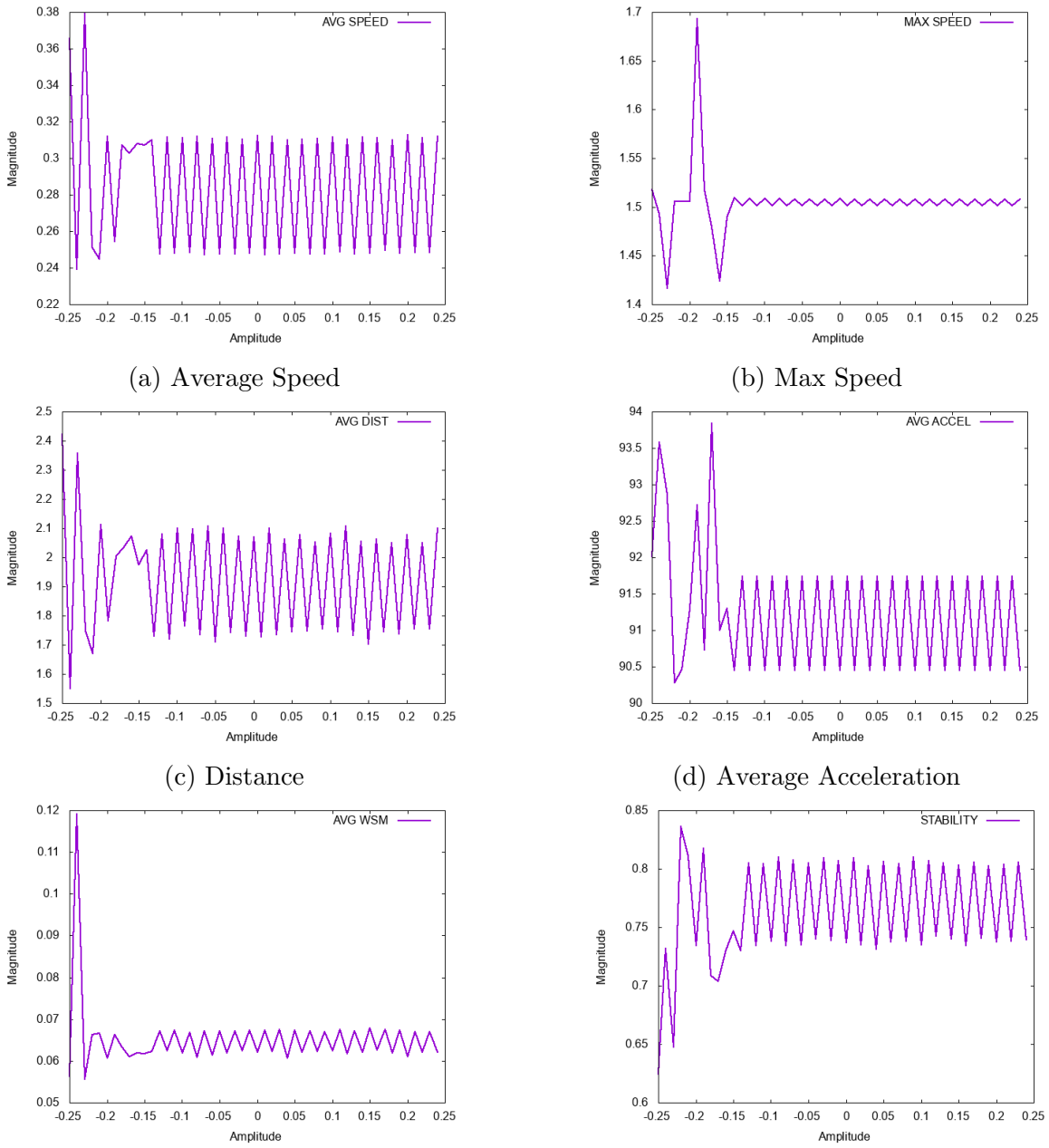
5.5.4 Leg Parameter Sweeps

The full sweep provided a good insight into the robots full parameter sensitivity. The research continued and the sweep was completed again, this time just affecting the legs. This was done because when looking at the stability results from the section "The Effect of an Active Spine on Gait Stability", the spine was quite influential, so the spine's values were left at a base value,

and the effect of the legs was investigated. This way, the research is able to investigate the effects of the legs on the performance of the robot. They do have quite a large effect, and so it is important to know how to utilise them properly.

This experiment also utilised the base sweep code, however it was modified so that the spine oscillators were not affected.

5.5.4.1 Walk - Amplitude



(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

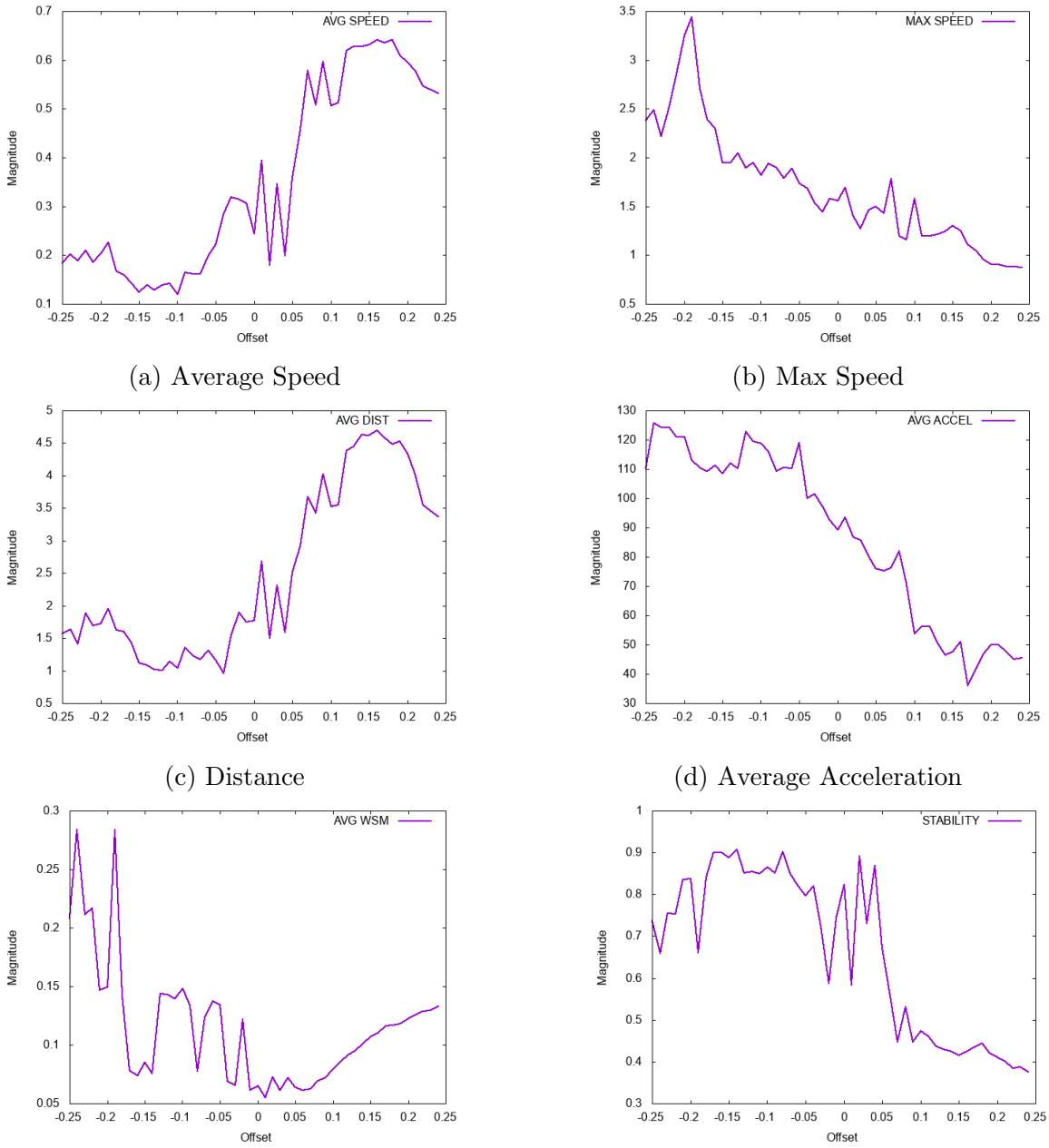
(e) Wide Stability Margin

(f) Stability

Figure 44: The effect of an amplitude leg sweep on a friction-less walk gait

Fig. 44 shows that just changing the amplitude of the legs has little effect on the performance of the walk gait, as each line centres around a median value with no trend.

5.5.4.2 Walk - Offset



(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

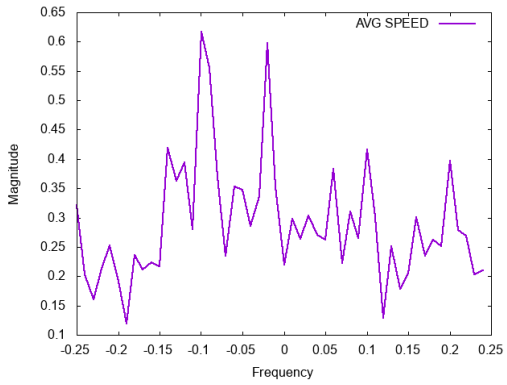
(e) Wide Stability Margin

(f) Stability

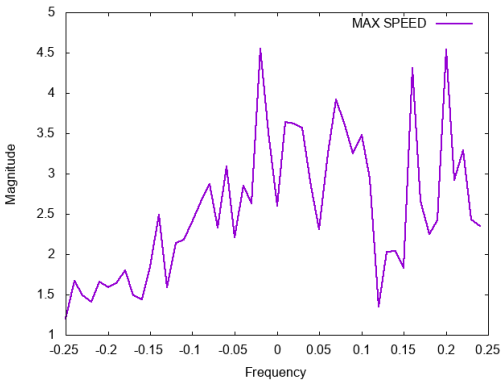
Figure 45: The effect of an offset leg sweep on a friction-less walk gait

Fig. 39 shows a positive trend of an increased offset and an increased average speed and distance for the leg sweep. There was a negative trend for max speed and average acceleration however. There is also a dropoff in stability and a slight rise in WSM.

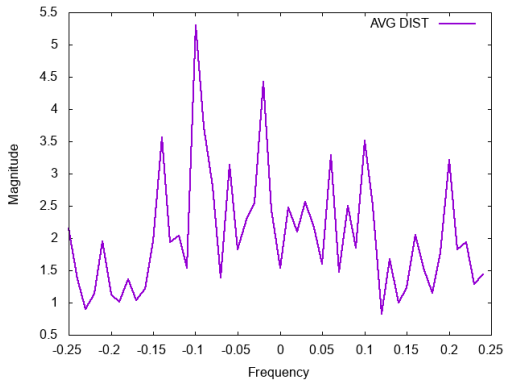
5.5.4.3 Walk - Frequency



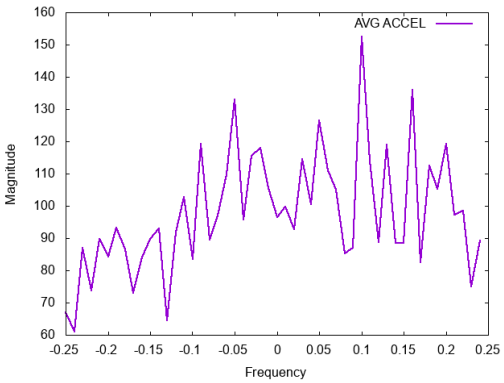
(a) Average Speed



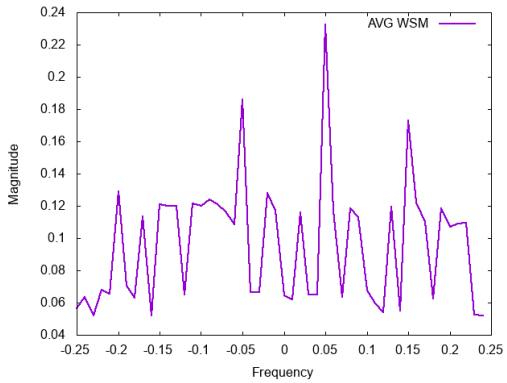
(b) Max Speed



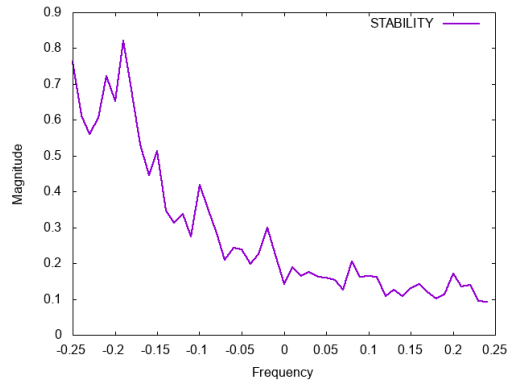
(c) Distance



(d) Average Acceleration



(e) Wide Stability Margin

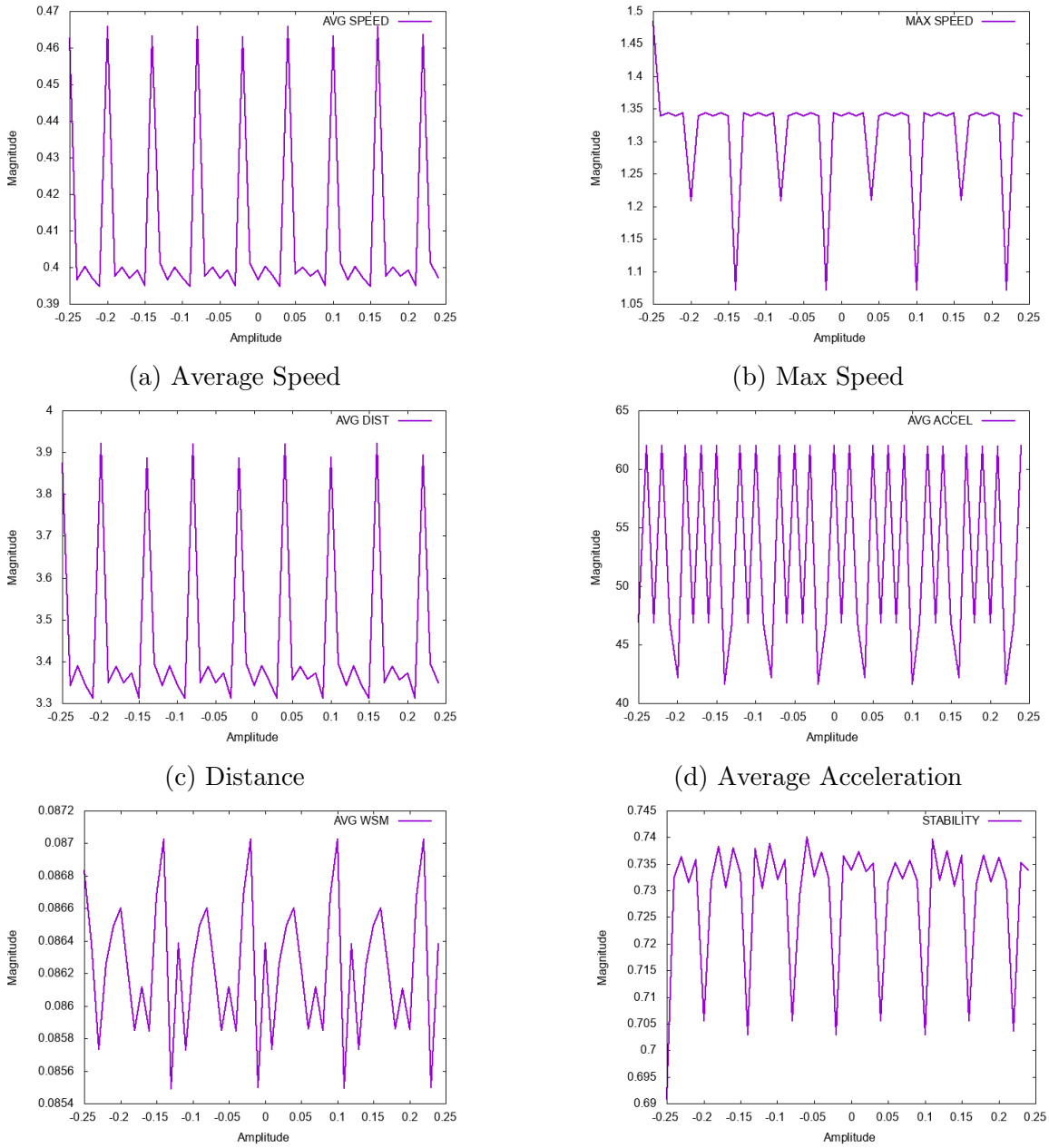


(f) Stability

Figure 46: The effect of a frequency leg sweep on a friction-less walk gait

Fig. 46 shows a downward trend in stability, and no trend elsewhere.

5.5.4.4 Trot - Amplitude



(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

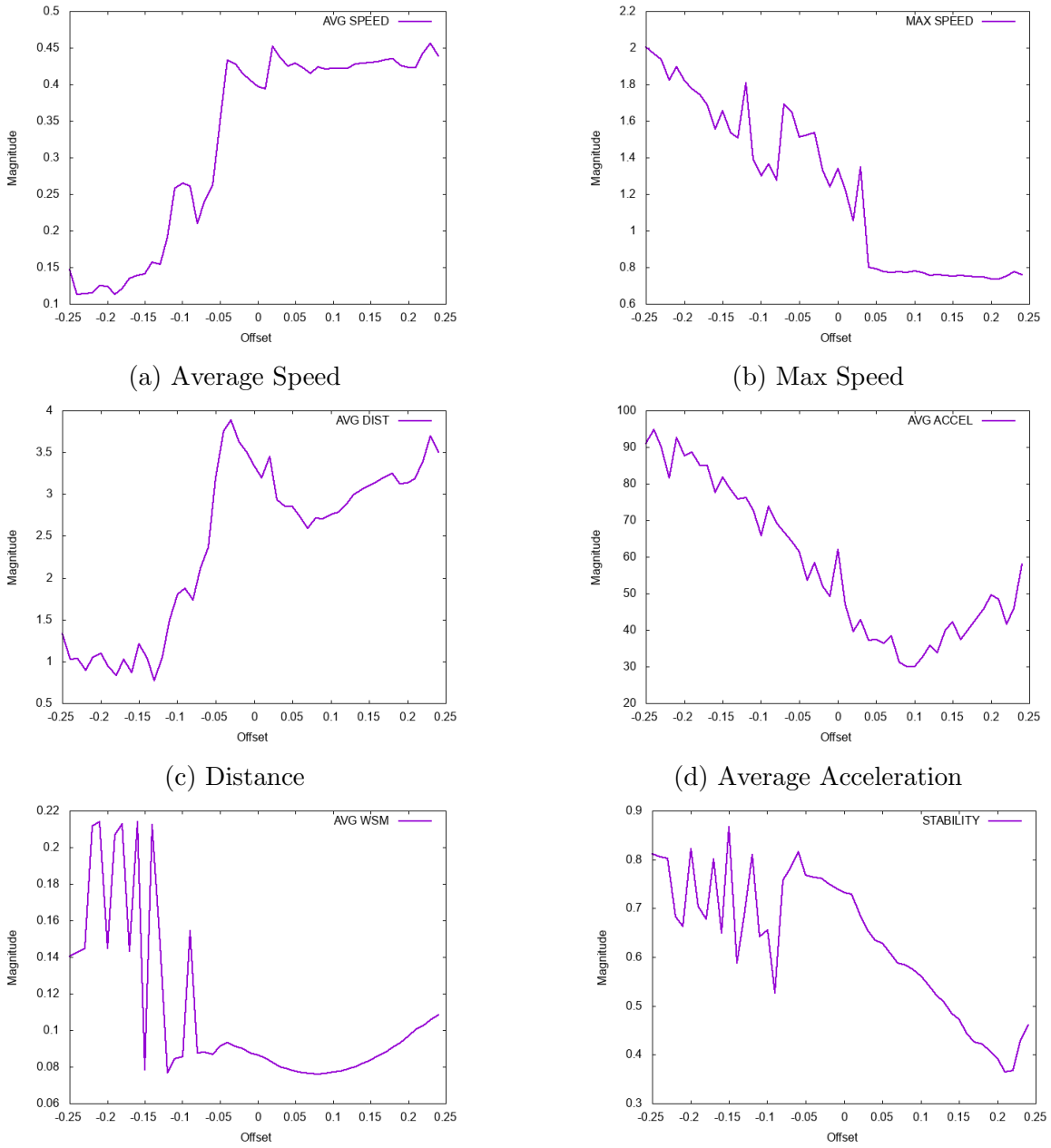
(e) Wide Stability Margin

(f) Stability

Figure 47: The effect of an amplitude leg sweep on a friction-less trot gait

Fig. 47 shows a similar pattern to the walk leg sweep, with peaks and troughs appearing at specific intervals. With more time and space in the project it could be possible to investigate what the interval is, and why the interval exists.

5.5.4.5 Trot - Offset



(a) Average Speed

(b) Max Speed

(c) Distance

(d) Average Acceleration

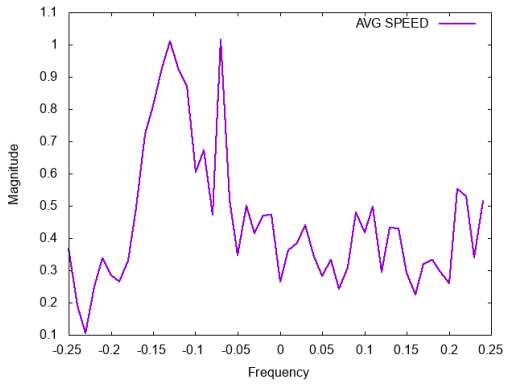
(e) Wide Stability Margin

(f) Stability

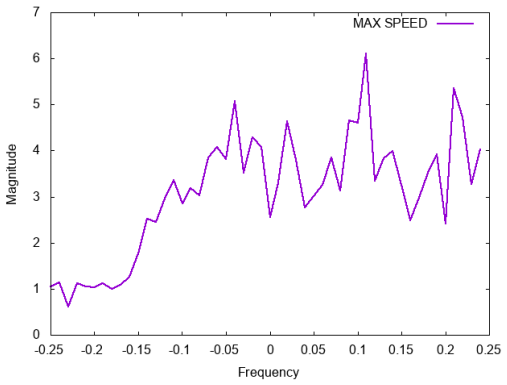
Figure 48: The effect of an offset leg sweep on a friction-less trot gait

Fig. 48 shows a downwards trend in stability when the leg offset is increased, there is also a major drop in maximum speed and acceleration at around 0.04 above the base value.

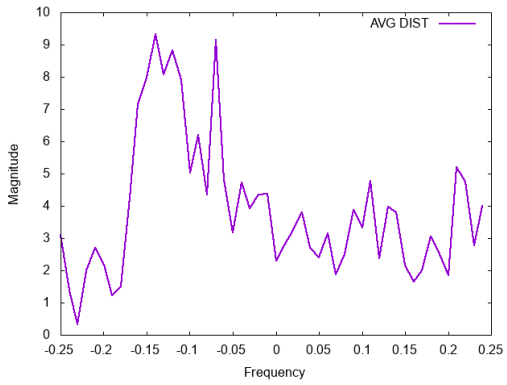
5.5.4.6 Trot - Frequency



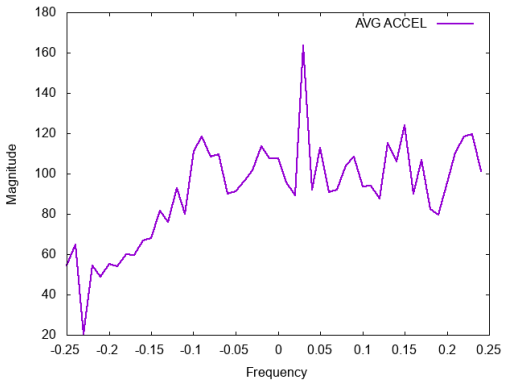
(a) Average Speed



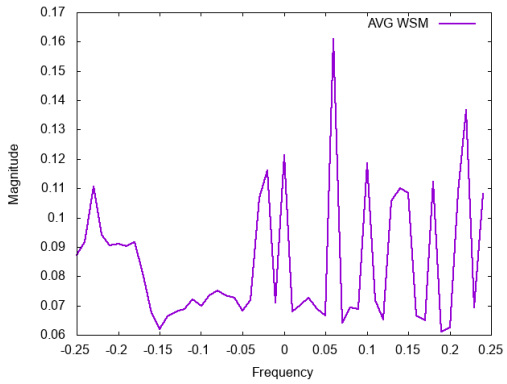
(b) Max Speed



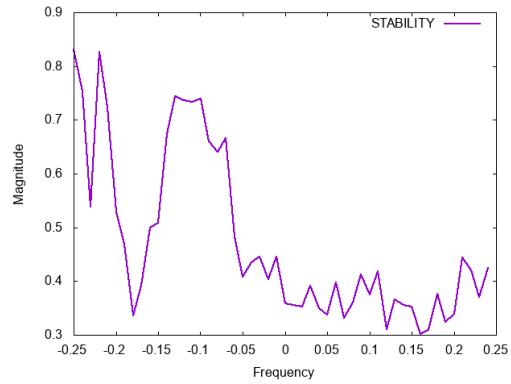
(c) Distance



(d) Average Acceleration



(e) Wide Stability Margin



(f) Stability

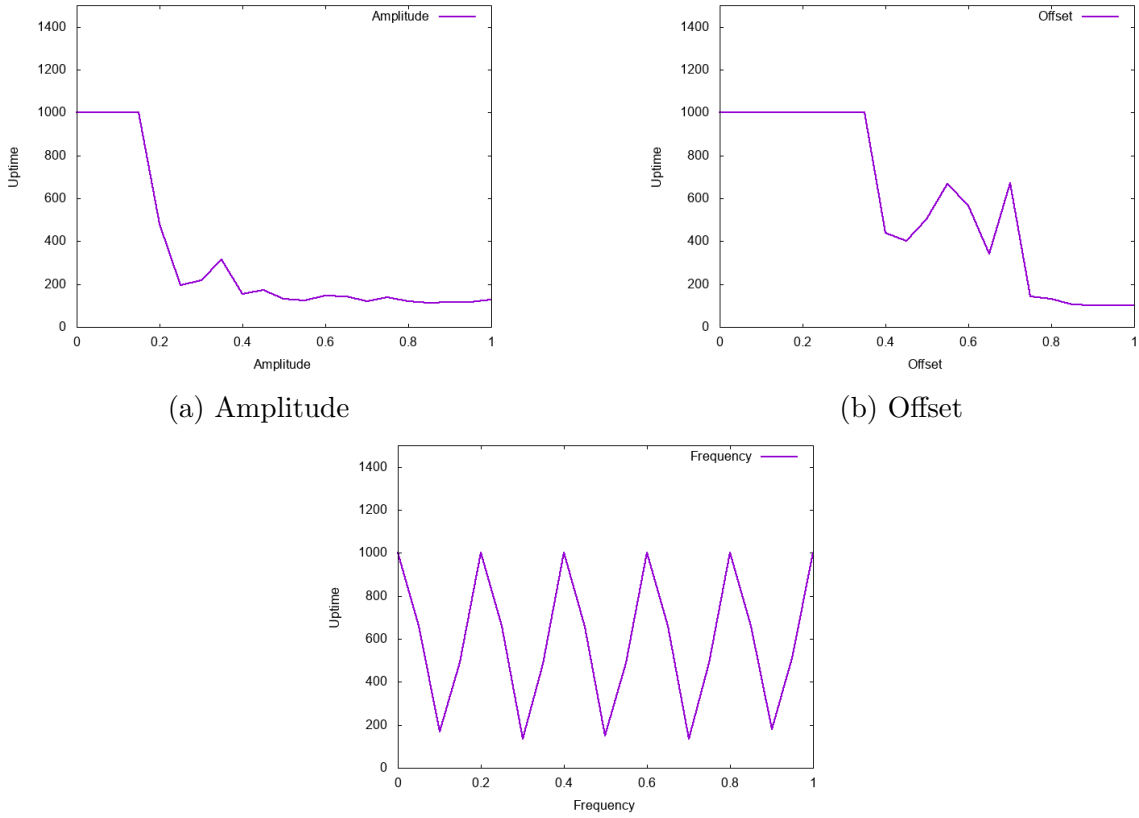
Figure 49: The effect of a frequency leg sweep on a friction-less trot gait

Fig. 49 shows that a higher frequency produces a lower stability. Too low of a frequency also lowers average speed, max speed and average acceleration (which directly affects distance).

5.5.5 Upright Time Sweeps

The metrics don't give the perfect view of whether the robot is truly stable or not, so alongside the previous experiment another experiment was run, which timed how long the robot stayed upright during the parameter sweeps. As such the research shows what range of values above the base value are acceptable for use during simulation which can produce valid locomotion,

this way the range for any machine learning techniques can be refined to allow for greater performance.



(a) Amplitude

(b) Offset

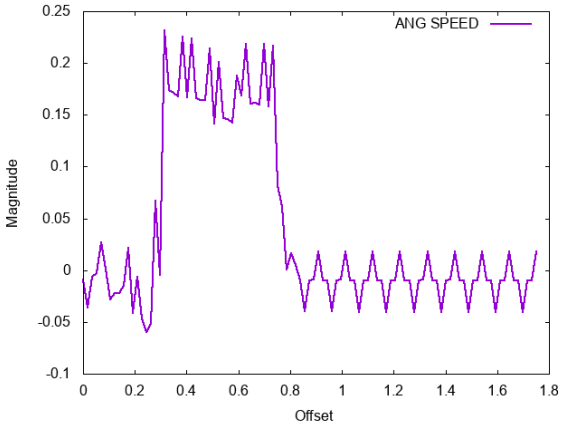
(c) Frequency

Figure 50: Full Sweep Uptime on the friction-less trot gait

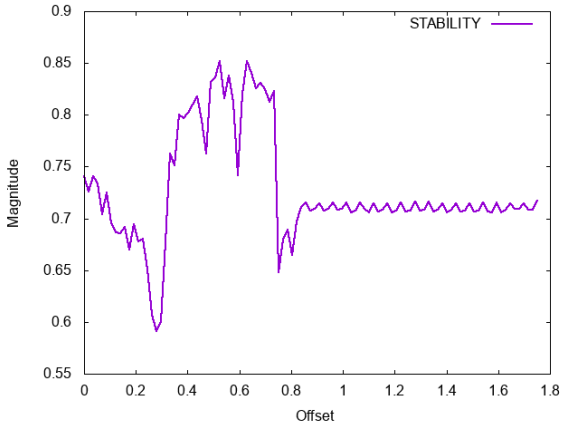
Fig. 50 shows that +0.17 over the base amplitude in every oscillator causes the robot will fall down during runtime. The same happens if every oscillator is +0.35 over the base offset. Whereas frequency produces a pattern, every odd value causes the robot to fall, and every even number remains upright.

5.5.6 Turn Speed

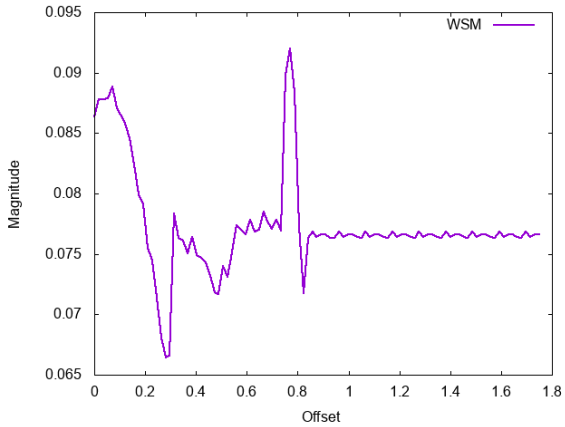
As discussed earlier in the paper, the tensegrity spine allows for more flexible movement. Turning was able to be achieved simply by setting the offset of the yaw of the spine to point in the direction of turning. To ensure that the correct offset was selected for specific situations, an experiment was conducted, similar to previous ones, where a parametric sweep was done over the offset of the yaw of the spine, while maintaining the default values for the rest of the oscillators. It was important to not only look at the angular velocity, but also the stability to ensure that the robot isn't sacrificing stability for speed.



(a) Angular Velocity



(b) Stability



(c) Wide Stability Margin

Figure 51: Turn Speed Experiment friction-less Results

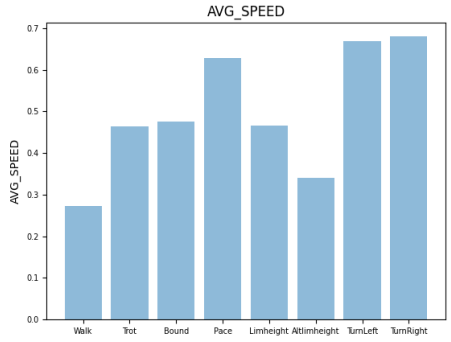
Fig. 51 shows that the best range of investigation for turnspeed is between +0 - +0.7 for yaw offset. Otherwise the stability becomes too low (the pattern here suggests that the robot is falling over). It is worth taking a deeper look at this and investigating how moving certain body parts (such as the shoulders) may help with increasing turning efficiency.

5.5.7 Gait Performance Comparison

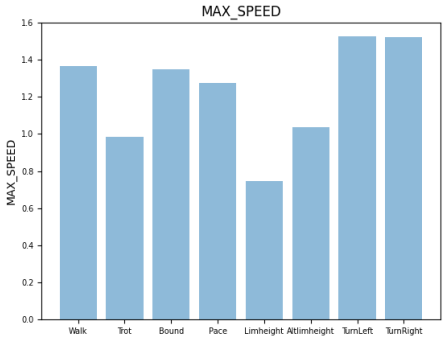
The research compared the performance of gaits with each other, in the hopes that the data could be used to help inform what gaits are useful in what situations. For example, knowing when to use a trot gait and knowing when to use a walk gait could possibly increase the efficiency of the robot.

This experiment was accomplished by using the earlier mentioned parameter editing code, and running the same loop, while swapping the gait CPG each time.

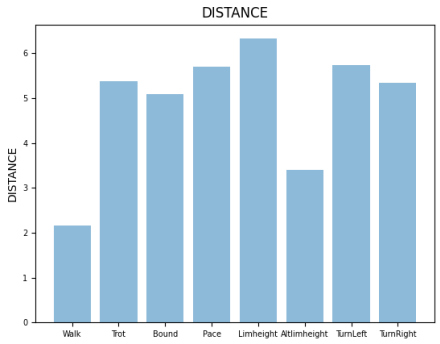
The experiment took an average of each metric (Average Speed, Max Speed, Distance Travelled, Max Acceleration, WSM and Stability) result from each gait so that the results are more accurate and are a better representation of performance.



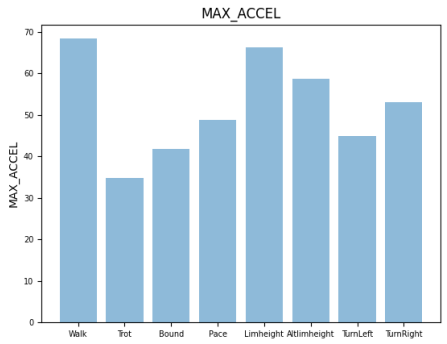
(a) Average Speed



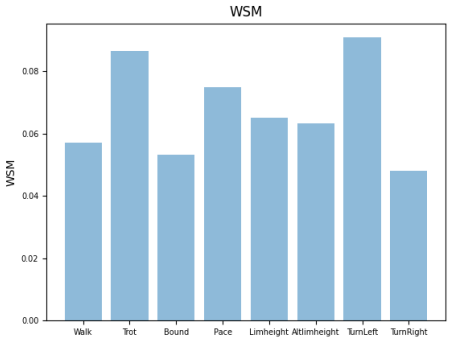
(b) Max Speed



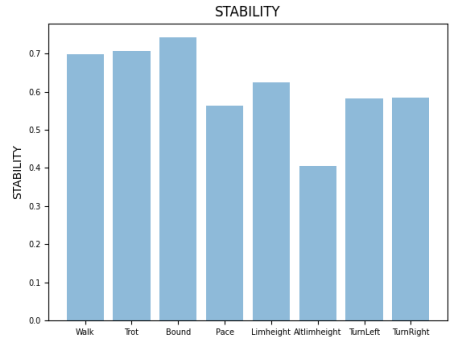
(c) Distance



(d) Max Acceleration



(e) Wide Stability Margin



(f) Stability

Figure 52: Bar charts depicting the different friction-less gait performances

Looking at Fig. 52, it suggests that although the trot isn't the fastest gait, it is the most stable, which arguably would make it the most useful gait overall, especially in uneven and unpredictable areas where stability is key.

It is important to note that this could change depending on the parameters of each gait, and so when they are scrutinised more heavily (with machine-learning etc.) the performance of the gaits will change, and a new best may come out of it. Overall (in line with other literature) the trot gait is the best gait.

As expected, looking at Fig. 52a and Fig. 52b, the turning gaits have a very similar, if not the same speed performance, this is expected because they have exactly the same parameters other than the offset in the spine roll which determines the direction of turn. Any other discrepancies/differences in performance (see Fig. 52e) are probably due to simulation error,

such as the physics engine doing the maths from a certain direction first, which gives bias to that direction.

The turning speeds are the highest because they are measured as angular velocity, and because they are travelling not just in the x-plane they are considered faster. The fastest gait in one plane however is the pace gait.

5.5.8 Manual Control - Extended Experiment

It is useful to have a track that TQBot could navigate (similar to [51]). This would prove that TQBot was capable of traversing non-linear paths, and was capable of turning simply by turning its spine. As opposed to just having a flat plane with which to experiment on, which provided no obstacles whatsoever.

For the purpose of demonstrations, it was also coupled with manual control, so that a user can use a keyboard to determine TQBot's movements. As the CPG is not particularly user friendly, one of the options for control has the keyboard inputs simply registered to specific gaits, allowing for user control but still used the predetermined gaits that were generated earlier. There is an alternative control scheme, which gives the user just the trot gait and has the relevant keys mapped to increasing/decreasing the yaw offset, which affects the turning speed, making it a delicate balance between turn speed and stability. This demonstrates to the user the considerations that are taken when designing a gait for TQBot.

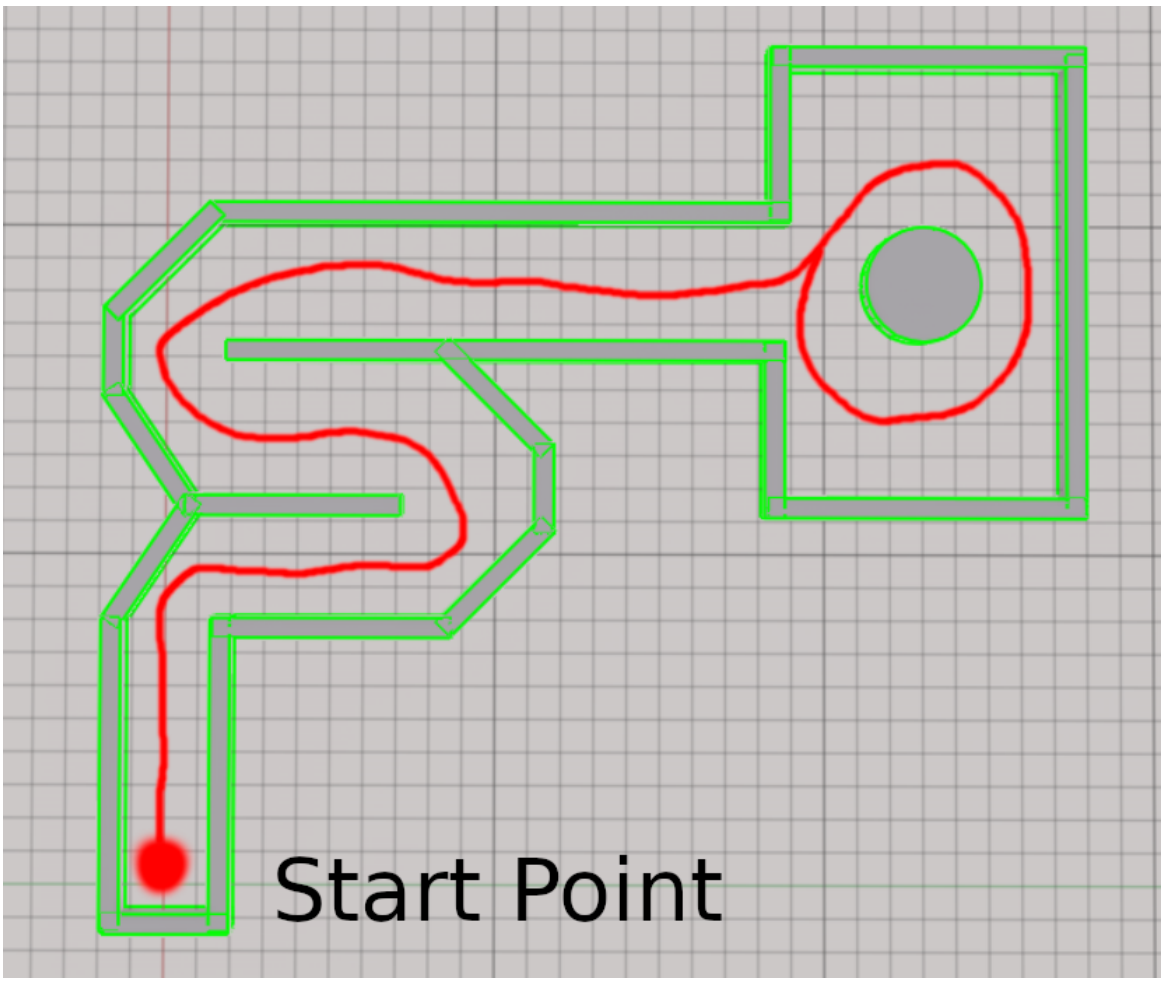


Figure 53: The S-Shaped Track

Fig. 53 displays the track designed for this experiment, the route for TQBot goes from the red starting point roughly following the red line to the ending circle, where the robot is tested for how well it can turn in a full circle.

This track could be expanded to be used for machine learning when used to help teach TQBot how to traverse predetermined routes/courses, or even how to reach an end point with nothing but sensor and location information.

6 Planning and Time Management

6.1 Original Gantt Chart Accuracy

The original Gantt chart was inaccurate almost straight away, as when the research was being completed and the gaits were being generated, it made sense to generate the easier ones first, as a warm-up to generating the more complicated ones. The phase-lags of the original gaits (walk/trot) had already been decided by Yunlong Lian, so to help him finish his paper I did work to decide the other parameters. Then once I had a better understanding of how the parameters affected the gaits, I produced other more complicated gaits.

The work remained ahead of schedule for the entirety of the gait generation section (and

for the rest of the project), with some overlap into the experimentation section, as coding for calculating the WSM and changing the CPG was completed before the gait generation deadline. Being ahead of schedule allowed for more time for fine tuning to produce a good baseline for the gaits before the experimentation could occur.

There was also a change in the plan for the parameter investigations instead of doing it manually and noting down the changes, a parameter sweeper was implemented instead, a sort of "brute force" method, this saved time and allowed for more accuracy in the results, as it allowed for smaller steps in parameter changes without the tedium of doing them all one by one. It also allowed other work to be completed while the parameter sweep ran.

Before any sort of full parameter sweeps the research ended up doing some more work for Yunlong Lian, which looked at the effect of an active spine on gait stability, and although this goes against the original Gantt chart it was still useful to complete early, as it was able to provide good insight into experimentation techniques for later in the project, it also aided Yunlong Lian when he was writing their conference paper.

Due to the automisation performed by me for the experiments, a lot of things occurred earlier than expected, this is a positive though. This allowed to me to get more experimentation done.

The Gantt chart containing the correct timing of everything can be found in the Appendix.

6.2 Reflection on Project Management

The project was managed quite well, as all time-frames/deadlines were adhered to, if not kept ahead of. Even with the setbacks the timeline of the project stayed smooth and well handled.

Overall the experimentation section of the project was an iterative process, with experiments being ran multiple times, changes being made to the experiments, and then the experiments being ran again, this was to ensure that the research was able to gather the most useful data possible. Some experiments were ran upwards of 7-8 times (totalling over 100+ hours). I believe that I managed the project quite well, achieving everything I set out to do and also being able to go much further. I worked hard to stay ahead of my timeline, and I worked even harder to mitigate any time loss caused by model inaccuracies/bugs.

There were a total of 3-4 reruns of experiments due to bugs with the simulation, with one happening just shy of a month before the project was due. I think it is a good testament to the design and flow of my project, which has allowed me to recoup from these time losses efficiently.

If I had to do the project again knowing what I know now, I would work harder to ensure that I plan out the experiments properly, ensuring that I save data correctly and run the experiments correctly, specifically remembering when to reset certain parameters. I would also write better plotting scripts, so that they are done in batches, as opposed to one by one.

I would also plan the project better and not overestimate how long it would take to achieve certain parts of the project, such as the parametric sweeps which I thought would take much longer.

The one goal I did not achieve was a proper investigation into the parameters effect on gait transitioning, I believe that this happened because I got distracted investigating the spine so

heavily, but I don't think my work suffered because of this as I was able to produce an in-depth report on the spine of the robot.

Due to my good planning and hard work, I was also able to assist a bachelors student with their project, providing support, some code, and also constructing a base incline map for them.

I am incredibly proud of the work I have produced, I have worked very hard, and I believe that shows, especially considering Yunlong Lian has added my name to one of his conference papers due to my contribution to his work.

6.3 Risk Assessment

One of the major risks with this project was that the model of TQBot was incorrect, which would produce incorrect data in the simulation. This risk would not stop the project, it would just mean that simulations would have to be repeated. The risk was mitigated, however it did occur twice, meaning that the simulations had to be rerun to produce new and more accurate results. However this was not too costly because the experiments were all set up and ready to go (because they weren't manual), it just meant that there were a few extra hours of running simulations, these were able to be overnight meaning that minimal time was lost and some time was spent reanalysing data. There also needed to be some time dedicated to fixing gaits that were affected by the model change, however this also did not take too long.

Another risk that had a minor chance of occurring (yet did) was the simulation tool being incorrect. This happened after all the work was completed, when a critical bug was discovered by Yunlong Lian, it was quite stressful, however it opened up an interesting chance to compare friction results and friction-less results (as mentioned earlier) and so ended up being a positive thing. The main downside being that I had less time to plan and run extra experiments. It also meant I had less space in this report to discuss any results gained from extra experiments.

7 Conclusion

7.1 Results Conclusion

After the experimentation completed during this project, the tensegrity spine is proven to provide a benefit to the robot if utilised correctly. One prime example are the results from the phase-lag experiment (Fig. 37d), which shows that if you have the yaw of the trot gait move across with an amplitude of 3 degrees and a phase-lag of π, you achieve high stability and a relatively high speed.

Overall the spine allows the robot to be more versatile, allowing lesser used gaits to be more utilised, as the spine can make them perform better. Especially when you combine 2 DOFs at once, particularly the roll and the yaw for the walk gait.

The spine forms an interesting discussion, because it does increase the complexity of the build, and the control scheme (that the implementation of the CPG lessens), but it increases

TQBot's inherent flexibility, and as we have seen in the experimentation, has the capability to increase the stability, or even the speed of TQBot.

However in regards to the spine, pitch has a detrimental effect on the stability of the robot. So the research recommends not using the pitch to try and achieve a high stability.

In general higher amplitude in the joints means higher acceleration, but lower stability, higher offset increases WSM of the robot and a higher frequency reduces stability.

Parameter	Min	Max
Trot Amplitude	0	0.18
Trot Offset	0	0.36
Yaw Offset for turning	-0.7	0.7

Table 1: Recommended range for values when used with base parameters

After looking at the upright experiment, the research can confidently say that when using other techniques to improve the trot on a friction-less surface, when using the base trot provided by this project, do not go 0.18 over the base amplitude, 0.36 over the base offset, and only use even frequencies.

When trying to turn using the flexible spine, don't go over 0.7 with the yaw offset, else the robot risks falling over. These results are also shown in Table 1.

These results are not all-encompassing, as there are an incredibly large amount of different permutations of the parameters that are possible, they do however provide an insight into the behaviour of TQBot.

7.2 Project Conclusion

This project defines the background surrounding TQBot, and describes several areas of interest for investigation in TQBot. It then defines several different gaits with parameters that produce interesting locomotion in simulation. It then thoroughly investigates the effect of the spine, and delivers a verdict to its usefulness. It then rigorously investigates the effect of changing the parameters on the robots gait and states what the research suggests for maximum parameter values.

The project has produced several different gaits/behaviours for TQBot and also lots of different experiments which can be used to gather data and investigate any CPG parameter that is required (even ones not created in this research).

The project has also produced multiple useful tools to help with the development of TQBot and its simulation including; calculating the Wide Stability Margin of TQBot, tools to access and modify the CPG parameters, tools to access and modify individual oscillators within the CPG, tools to save experimental data in CSV files, plotting tools in GNUplot and Pyplot with the matplotlib package.

8 Further Work

The number of experiments you could run in simulation are practically endless, there are thousands of different combinations of joints that you could investigate, with even more different ways to investigate them (i.e. different parameters to change, parameter steps etc.). Some examples of extra experiments include; multi-parameter sweeps (i.e. sweeping two parameters at the same time), a deeper investigation into gait transitioning speed and research into a feedback loop to help stabilise the robot.

This is why I think that the next step would be to introduce machine learning into the equation, and use ML techniques to improve the current gaits proposed. To aid the ML, the research has completed a set of joint parameter sweeps, in which specific sets of joints had there parameters swept and metrics measured. This data has been submitted alongside this report (as well as placed in the appendix), as there was not enough word-count to analyse it all. Once the online learning has been completed, the next step is to construct the real TQBot and implement the CPG controller in real life, and test the proposed gaits on that. Once that has been done and suitable gaits have been verified, it would then be sensible to run similar experiments as tested in the simulation on the real-world counterpart, to try and bridge the reality gap.

There are also plenty of graphs in the appendix that I produced from several different experiments (which mainly consist of reruns post-friction bug), I did not have the word count to analyse them however, so they could be analysed to help with future work with TQBot. Particularly interesting is the joint-sweeps that swept sets of joints together, knowing how to manipulate these to better locomote TQBot is of great interest to the work that continues from this project. There are also parameter sweep graphs for the pace and bound gaits.

9 Statement of Ethics

After looking at the University's code of practice and principles for good ethical practice, no ethical issues were identified in this project.

References

[1] R. B. McGhee, *Robot Locomotion*, pp. 237–264. Boston, MA: Springer US, 1976.

[2] S. Park, Y. Oh, and D. Hong, "Disaster response and recovery from the perspective of robotics," *International Journal of Precision Engineering and Manufacturing*, vol. 18, no. 10, p. 1475–1482, 2017.

[3] T. Ishii, S. Masakado, and K. Ishii, "Locomotion of a quadruped robot using cpg," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, vol. 4, pp. 3179–3184 vol.4, 2004.

[4] D. Owaki and A. Ishiguro, "A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[5] S. Hirose, "A study of design and control of a quadruped walking vehicle," *The International journal of robotics research*, vol. 3, no. 2, pp. 113–133, 1984.

[6] R. W. Xu, K. C. Hsieh, U. H. Chan, H. U. Cheang, W. K. Shi, and C. T. Hon, "Analytical review on developing progress of the quadruped robot industry and gaits research," in *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*, pp. 1–8, IEEE, 2022.

[7] S. K. D, "Continuous tension, discontinuous compression structures," 1965.

[8] C. Paul, J. Roberts, H. Lipson, and F. Valero Cuevas, "Gait production in a tensegrity based robot," in *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pp. 216–222, 2005.

[9] T. Bliss, T. Iwasaki, and H. Bart-Smith, "Central pattern generator control of a tensegrity swimmer," *IEEE/ASME transactions on mechatronics*, vol. 18, no. 2, pp. 586–597, 2013.

[10] B. R. Tietz, R. W. Carnahan, R. J. Bachmann, R. D. Quinn, and V. SunSpiral, "Tetraspine: Robust terrain handling on a tensegrity robot using central pattern generators," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 261–267, IEEE, 2013.

[11] A. P. Sabelhaus, H. Zhao, E. L. Zhu, A. K. Agogino, and A. M. Agogino, "Model-predictive control with inverse statics optimization for tensegrity spine robots," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, pp. 263–277, 2021.

[12] S. M. L., "Control of walking and running by means of electrical stimulation of the midbrain," *Biophysics*, vol. 11, pp. 659–666, 1966.

[13] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, p. 642–653, 2008.

[14] E. Marder and D. Bucher, "Central pattern generators and the control of rhythmic movements," *Current Biology*, vol. 11, no. 23, pp. R986–R996, 2001.

[15] L. Righetti and A. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," pp. 819 – 824, 06 2008.

[16] A. Crespi, D. Lachat, A. Pasquier, and A. J. Ijspeert, "Controlling swimming and crawling in a fish robot using a central pattern generator," *Autonomous Robots*, vol. 25, no. 1-2, p. 3–13, 2007.

[17] A. Crespi and A. Ijspeert, "Amphibot ii: An amphibious snake robot that crawls and swims using a central pattern generator," *Proceedings of the 9th International Conference on Climbing and Walking Robots (CLAWAR 2006)*, 01 2006.

[18] H. Yu, H. Gao, L. Ding, M. Li, Z. Deng, and G. Liu, "Gait generation with smooth transition using cpg-based locomotion control for hexapod walking robot," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 9, pp. 5488–5500, 2016.

[19] D. Grzelczyk, O. Szymanowska, and J. Awrejcewicz, "Kinematic and dynamic simulation of an octopod robot controlled by different central pattern generators," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 233, no. 4, pp. 400–417, 2019.

[20] F. Dzeladini, N. Ait-Bouziad, and A. Ijspeert, "Cpg-based control of humanoid robot locomotion," *Humanoid Robotics: A Reference*, pp. 1–35, 2018.

[21] H. Kimura, S. Akiyama, and K. Sakurama *Autonomous Robots*, vol. 7, no. 3, p. 247–258, 1999.

[22] F. Xie, Y. Zhong, R. Du, and Z. Li, "Central pattern generator (cpg) control of a biomimetic robot fish for multimodal swimming," *Journal of bionics engineering*, vol. 16, no. 2, pp. 222–234, 2019.

[23] J. Yu, S. Chen, Z. Wu, X. Chen, and M. Wang, "Energy analysis of a cpg-controlled miniature robotic fish," *Journal of bionics engineering*, vol. 15, no. 2, pp. 260–269, 2018.

[24] C. Liu, Y. Chen, J. Zhang, and Q. Chen, "Cpg driven locomotion control of quadruped robot," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 2368–2373, 2009.

[25] A. Sproewitz, R. Moeckel, J. Maye, and A. J. Ijspeert, "Learning to move in modular robots using central pattern generators and online optimization," *The International Journal of Robotics Research*, vol. 27, no. 3-4, p. 423–443, 2008.

[26] D. Korkmaz, G. Ozmen Koca, G. Li, C. Bal, M. Ay, and Z. H. Akpolat, "Locomotion control of a biomimetic robotic fish based on closed loop sensory feedback cpg model," *Journal of Marine Engineering & Technology*, vol. 20, no. 2, pp. 125–137, 2021.

[27] M. Wang, J. Yu, and M. Tan, "Cpg-based sensory feedback control for bio-inspired multimodal swimming," *International Journal of Advanced Robotic Systems*, vol. 11, no. 10, p. 170, 2014.

[28] R. Héliot and B. Espiau, "Multisensor input for cpg-based sensory—motor coordination," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 191–195, 2008.

[29] J. Yu, Z. Wu, M. Wang, and M. Tan, "Cpg network optimization for a biomimetic robotic fish via pso," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 9, pp. 1962–1968, 2016.

[30] Q. Lu, Z. Zhang, and W. Li, "Parameter optimization of cpg and its application in robot," *MATEC Web of Conferences*, vol. 232, p. 3018, 2018.

[31] M. Oliveira, V. Matos, C. P. Santos, and L. Costa, "Multi-objective parameter cpg optimization for gait generation of a biped robot," in *2013 IEEE International Conference on Robotics and Automation*, pp. 3130–3135, IEEE, 2013.

[32] P. Wen, X. Linsen, F. Baolin, and W. Zhong, "Cpg control model of snake-like robot parameters of optimization based on ga," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1944–1949, 2015.

[33] D. Hustig-Schultz, V. SunSpiral, and M. Teodorescu, "Morphological design for controlled tensegrity quadruped locomotion," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4714–4719, 2016.

[34] M. Oliveira, C. P. Santos, L. Costa, V. Matos, and M. Ferreira, "Multi-objective parameter cpg optimization for gait generation of a quadruped robot considering behavioral diversity," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2286–2291, IEEE, 2011.

[35] Y. Yusoff, M. S. Ngadiman, and A. M. Zain, "Overview of nsga-ii for optimizing machining process parameters," *Procedia Engineering*, vol. 15, pp. 3978–3983, 2011. CEIS 2011.

[36] G. Bellegarda and A. Ijspeert, "Cpg-rl: Learning central pattern generators for quadruped locomotion," *Ieee Robotics And Automation Letters*, vol. 7, no. 4, pp. 12547–12554, 2022.

[37] G. Bellegarda and A. Ijspeert, "Visual cpg-rl: Learning central pattern generators for visually-guided quadruped navigation," *arXiv preprint arXiv:2212.14400*, 2022.

[38] F. Ruppert and A. Badri-Spröwitz, "Learning plastic matching of robot dynamics in closed-loop central pattern generators," *Nature Machine Intelligence*, vol. 4, no. 7, pp. 652–660, 2022.

[39] N. Sasagawa, S. Oyake, and Y. Maeda, "Analysis of quadrupedal locomotion and energy using hardware cpg model," *Transaction of Japanese Society for Medical and Biological Engineering*, vol. 54Annual, no. 27PM-Abstract, pp. S200–S200, 2016.

[40] M. Oliveira, C. P. Santos, and L. Costa, "Sensitivity analysis of multi-objective optimization of cpg parameters for quadruped robot locomotion," in *AIP Conference Proceedings*, vol. 1479, pp. 495–498, 2012.

[41] S. Pouya, M. Khodabakhsh, A. Spröwitz, and A. Ijspeert, "Spinal joint compliance and actuation in a simulated bounding quadruped robot," *Autonomous Robots*, vol. 41, pp. 437–452, 2017.

[42] Nvidia, "Isaacsim tutorial [online], available: docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim," Feb 2023.

[43] Pixar, "Usd home [online], available: https://graphics.pixar.com/usd/release/index.html," Feb 2023.

[44] Y. Lian, "Tqbot pdf - private communication." Jan 2023.

[45] Y. Lian, T. Wang, J. Ingham, M. A Post, and A. Tyrell, "Cpg-based locomtion control of a quadruped robot with an active spine." 2023.

[46] "Boston dynamics spot available: https://www.bostondynamics.com/products/spot."

[47] "Unitree go1 available: https://shop.unitree.com/en-gb/products/unitreeyushutechnologydog-artificial-intelligence-companion-bionic-companion-intelligent-robot-go1-quadruped-robot-dog."

[48] D. J. Cho, J. H. Kim, and D. G. Gweon, "Optimal turning gait of a quadruped walking robot," *Robotica*, vol. 13, no. 6, p. 559–564, 1995.

[49] D. Sunday, *Practical Geometry Algorithms: With C++ Code*. Amazon Digital Services LLC - KDP Print US, 2021.

[50] Y. Lian, "Stability value document - private communication." Feb 2023.

[51] T. Horvat, K. Melo, and A. J. Ijspeert, "Spine controller for a sprawling posture robot," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1195–1202, 2017.

10 Bibliography

CPG-based locomotion control of a quadruped robot with an active spine - Yunlong Lian, Tianyuan Wang, Joe Ingham, Mark A Post, and Andy Tyrell - A conference paper submitted to TAROS 2023 for review.

The gaits accompanying videos that were produced for this project by me can be found here https://shorturl.at/dlmU0

A Pseudocode

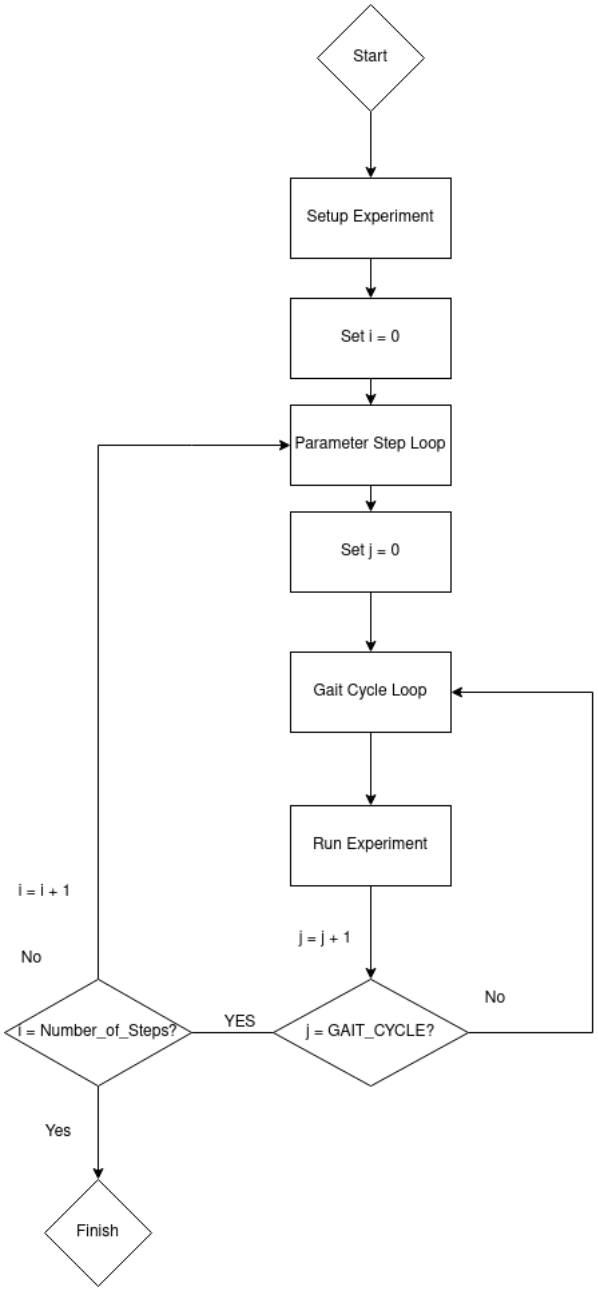


Figure 54: General Experiment Function Flowchart

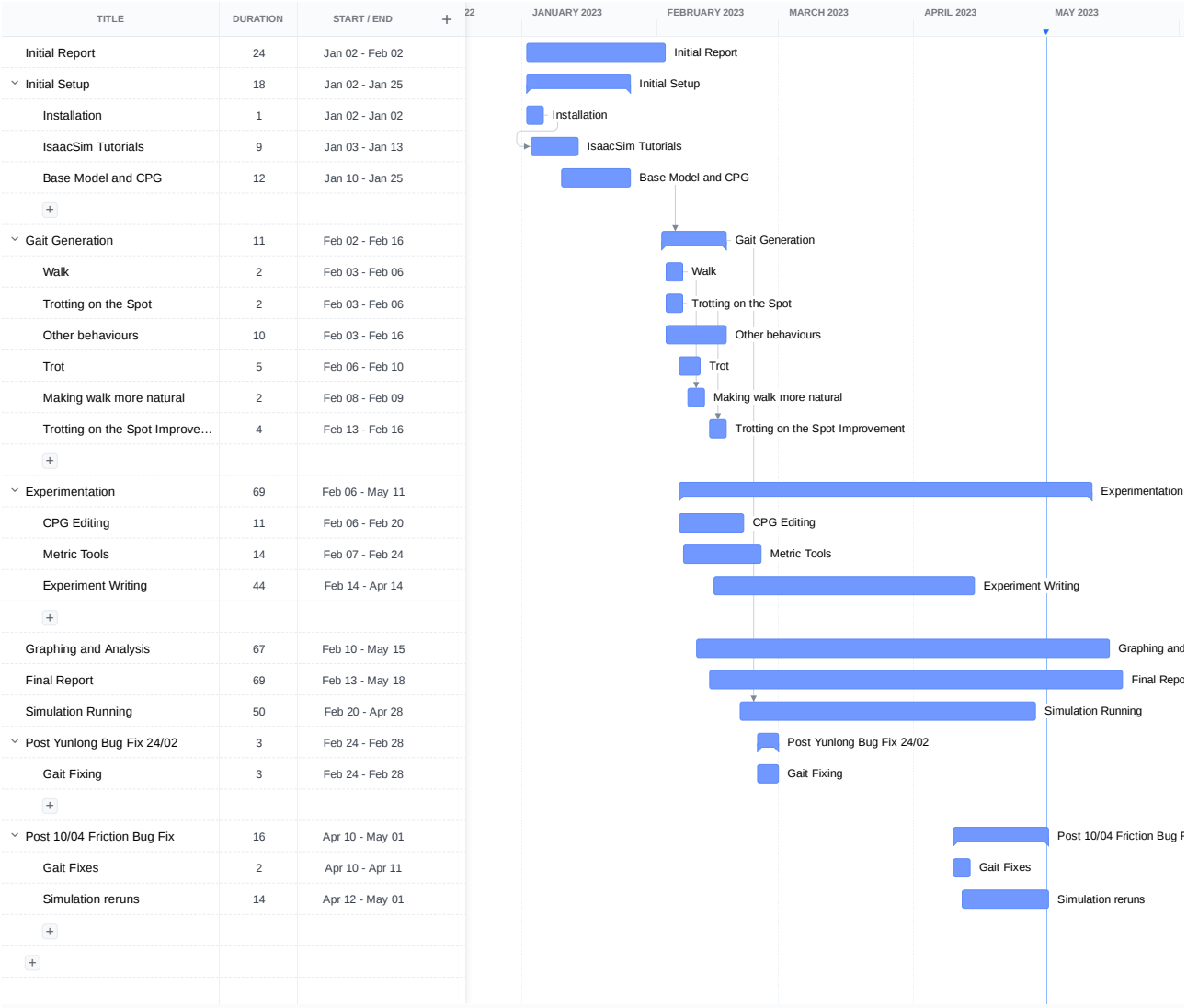
```
Calculate_WSM():

        get RightFrontFoot Coords
        get RightRearFoot Coords
        get LeftFrontFoot Coords
        get LeftRearFoot Coords
        get CentreofGravity Coords

        create polygon from foot coords

        for each vertex in polygon:
                calculate distance from vertex to CentreofGravity

                if distance = shortest_distance
                        WSM = distance


        return WSM
```

Figure 55: WSM Pseudocode

```
run_generic_sweep(gait, total_steps, parameter_to_change, parameter_range):

        generate model with gait

        set model parameter_to_change to minmum(parameter_range)

        for each step in total_steps:

                for each cycle in gait_cycles:

                        run simulation

                        record data


                average simulation data

                increase relevant joint parameters


        save data to csv file
```

Figure 56: Sweep Pseudocode

B Gantt Chart



C Gait Parameters

C.1 Walk

```
JOE_WALK_VECTOR = [0.1, 0, 0.1, -np.pi/2, 0, np.pi/2, np.pi, -np.pi, -np.pi/2, 0, np.pi/2, -np.pi/2, 0, np.pi/2, np.pi]
JOE_WALK_SPEED_STANCE = [0.4 for i in range(OSCILLATOR_NUMBERS)]
JOE_WALK_SPEED_SWING = [1.2 for i in range(OSCILLATOR_NUMBERS)]
JOE_WALK_DES_AMPLITUDE = [0.05, 0, 0.05, 0.025, 0.025, 0.025, 0.025, 0.3925, 0.3925, 0.3925, 0.3925, 0.2, 0.4, 0.2, 0.4]
JOE_WALK_DES_OFFSET = [0, 0, 0, 0.1, 0.1, 0.1, 0.1, 0, 0, 0, 0, 0, -0.3, 0, -0.3]
```

Figure 57: Friction-less Walk Parameters

C.2 Trot

```
JOE_TROT_VECTOR = [0, 0, 0, -np.pi/2, np.pi/2, np.pi/2, -np.pi/2, -np.pi, 0, 0, -np.pi, -np.pi/2, np.pi/2, np.pi/2, -np.pi/2]
JOE_TROT_SPEED_STANCE = [0.7 for i in range(OSCILLATOR_NUMBERS)]
JOE_TROT_SPEED_SWING = [0.7 for i in range(OSCILLATOR_NUMBERS)]
JOE_TROT_DES_AMPLITUDE = [0.05, 0, 0.05, 0.025, 0.025, 0.025, 0.025, 0.3925, 0.3925, 0.3925, 0.3925, 0.2, 0.2, 0.2, 0.2]
JOE_TROT_DES_OFFSET = [0, 0, 0, 0.1, 0.1, 0.1, 0.1, 0, -0.3, 0, -0.3, 0, 0, 0, 0]
```

Figure 58: Friction-less Trot Parameters

C.3 Friction-less Trotting on the Spot Parameters

```
JOE_SPOTTROT_VECTOR = [0, 0, 0, 0, 0, 0, 0, np.pi, 0, 0, np.pi, 0, np.pi*0.99, np.pi*0.99, 0]
JOE_SPOTTROT_SPEED_STANCE = [1 for i in range(OSCILLATOR_NUMBERS)]
JOE_SPOTTROT_SPEED_SWING = [1 for i in range(OSCILLATOR_NUMBERS)]
JOE_SPOTTROT_DES_AMPLITUDE = [0, 0, 0, 0, 0, 0, 0, 0.15, 0.15, 0.15, 0.15, 0.35, 0.35, 0.35, 0.35]
JOE_SPOTTROT_DES_OFFSET = [0, 0, 0, 0.05, 0.05, 0.05, 0.05, 0, 0, 0, 0, 0, 0, 0, 0]
```

Figure 59: Friction-less Trotting on the Spot Parameters

C.4 Turning on the Spot

```
JOE_SPOTTURN_VECTOR = [0, 0, 0, 0, np.pi*0.99, np.pi*0.99, 0, np.pi, 0, 0, np.pi, 0, np.pi*0.99, np.pi*0.99, 0]
JOE_SPOTTURN_SPEED_STANCE = [1 for i in range(OSCILLATOR_NUMBERS)]
JOE_SPOTTURN_SPEED_SWING = [1 for i in range(OSCILLATOR_NUMBERS)]
JOE_SPOTTURN_DES_SWING_AMP = [0, 0, 0, 0, 0, 0, 0, 0.15, 0.15, 0.15, 0.15, 0.35, 0.35, 0.35, 0.35]
JOE_SPOTTURN_DES_STANCE_AMP = [0, 0, 0, 0.25, -0.25, -0.25, 0.25, 0, 0, 0, 0, 0, 0, 0, 0.01]
JOE_SPOTTURN_DES_OFFSET = [0, 0, 0, 0.05, 0.05, 0.05, 0.05, 0, 0, 0, 0, 0, 0, 0, 0]
```

Figure 60: Friction-less Turning on the Spot Parameters

C.5 Pace

```
JOE_PACE_VECTOR = [0, 0, 0, -np.pi/2, -np.pi/2, np.pi/2, np.pi/2, -np.pi, -np.pi, 0, 0, -np.pi/2, -np.pi/2, np.pi/2, np.pi/2]
JOE_PACE_DES_AMPLITUDE = [0.125, 0, 0.05, 0.05, 0.05, 0.05, 0.05, 0.4, 0.4, 0.4, 0.4, 0.25, 0.25, 0.25, 0.25]
JOE_PACE_DES_OFFSET = [0, -0.1, 0, 0.1, 0.1, 0.1, 0.1, 0, 0, 0, 0, 0, 0, 0, 0]
JOE_PACE_SPEED_STANCE = [0.8 for i in range(OSCILLATOR_NUMBERS)]
JOE_PACE_SPEED_SWING =  [0.8 for i in range(OSCILLATOR_NUMBERS)]
```

Figure 61: Friction-less Pace Parameters

C.6 Bound

```
JOE_BOUND_VECTOR = [0, 0, 0, np.pi/2, -np.pi/2, np.pi/2, -np.pi/2, 0, np.pi, 0, np.pi, np.pi/2, -np.pi/2, np.pi/2, -np.pi/2]
JOE_BOUND_DES_AMPLITUDE = [0.05, 0.05, 0.05, 0.025, 0.025, 0.025, 0.025, 0.3925, 0.3925, 0.3925, 0.3925, 0.2, 0.1, 0.2, 0.1]
JOE_BOUND_SPEED_STANCE = [0.8 for i in range(OSCILLATOR_NUMBERS)]
JOE_BOUND_SPEED_SWING = [0.8 for i in range(OSCILLATOR_NUMBERS)]
DESIRED_OFFSET = [0, 0, 0, 0.1, 0.1, 0.1, 0.1, 0, 0, 0, 0, 0, 0, 0, 0]
```

Figure 62: Friction-less Bound Parameters

```
JOE_LIMHEIGHT_VECTOR = [0, 0, 0, -np.pi/2, 0, np.pi/2, np.pi, -np.pi, -np.pi/2, 0, np.pi/2, -np.pi/2, 0, np.pi/2, np.pi]
JOE_LIMHEIGHT_DES_AMPLITUDE = [0.05, 0, 0.05, 0.025, 0.025, 0.025, 0.025, 0.3925, 0.3925, 0.3925, 0.3925, 0.2, 0.2, 0.2, 0.2]
#Essentially increase the shoulder offset so that the legs are lower, almost like a crawl
JOE_LIMHEIGHT_DES_OFFSET = [0, 0, 0, 0.5, 0.5, 0.5, 0.5, 0, 0, 0, 0, 0, 0, 0, 0]
JOE_LIMHEIGHT_SPEED_STANCE = [0.4 for i in range(OSCILLATOR_NUMBERS)]
JOE_LIMHEIGHT_SPEED_SWING = [1.2 for i in range(OSCILLATOR_NUMBERS)]
```

Figure 64: Friction-less Limited Height Movement Parameters

C.7 Bending the Spine

```
#Bending the spine - no other movement
#all 0s because only the spine is going to move - no forward movement
JOE_SPINEBEND_VECTOR = [0, -(2*np.pi/3), -(4*np.pi/3), 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
#Only let the spine move
JOE_SPINEBEND_DES_AMPLITUDE = [0.15, 0.15, 0.15, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
JOE_SPINEBEND_SPEED_STANCE = [0.4 for i in range(OSCILLATOR_NUMBERS)]
JOE_SPINEBEND_SPEED_SWING = [0.4 for i in range(OSCILLATOR_NUMBERS)]
DESIRED_OFFSET = [0, 0, 0, 0.1, 0.1, 0.1, 0.1, 0, 0, 0, 0, 0, 0, 0, 0]
```

Figure 63: Friction-less Spine Bending Parameters

C.8 Limited Height Movement

```
JOE_ALTLIMHEIGHT_VECTOR = [0, 0, 0, -np.pi/2, 0, np.pi/2, np.pi, -np.pi, -np.pi/2, 0, np.pi/2, -np.pi/2, 0, np.pi/2, np.pi]
JOE_ALTLIMHEIGHT_DES_AMPLITUDE = [0.05, 0, 0.05, 0.025, 0.025, 0.025, 0.3925, 0.3925, 0.3925, 0.3925, 0.2, 0.4, 0.2, 0.4]
#Have the hips on front and back offset in the opposite way
JOE_ALTLIMHEIGHT_DES_OFFSET = [0, 0, 0, 0.1, 0.1, 0.1, 0.1, 0.75, -0.75, 0.75, -0.75, -0.5, 0.5, -0.5, 0.5]
JOE_ALTLIMHEIGHT_SPEED_STANCE = [0.7 for i in range(OSCILLATOR_NUMBERS)]
JOE_ALTLIMHEIGHT_SPEED_SWING = [0.7 for i in range(OSCILLATOR_NUMBERS)]
```

Figure 65: Friction-less Alternate Limited Height Movement Parameters

C.9 Turning

```
JOE_TURN_VECTOR = [0, 0, 0, -np.pi/2, np.pi/2, np.pi/2, -np.pi/2, -np.pi, 0, 0, -np.pi, -np.pi/2, np.pi/2, np.pi/2, -np.pi/2]
JOE_TURN_SPEED_STANCE = [0.7 for i in range(OSCILLATOR_NUMBERS)]
JOE_TURN_SPEED_SWING = [0.7 for i in range(OSCILLATOR_NUMBERS)]
JOE_TURN_DES_AMPLITUDE = [0.05, 0, 0.05, 0.025, 0.025, 0.025, 0.025, 0.3925, 0.3925, 0.3925, 0.3925, 0.2, 0.2, 0.2, 0.2]
JOE_TURN_DES_OFFSET = [0, 0, -0.1, 0.1, 0.1, 0.1, 0.1, 0, 0, 0, 0, 0, 0, 0, 0]
```

Figure 66: Turning Left Parameters

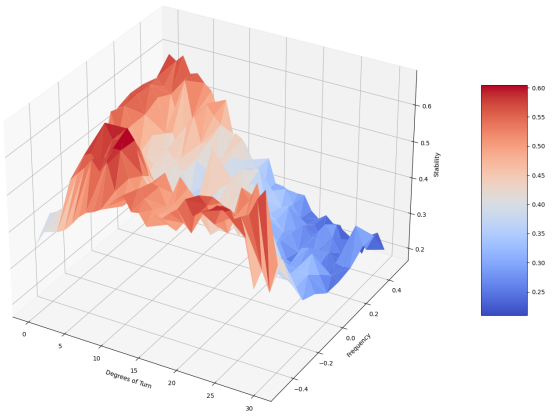
```
JOE_TURN_VECTOR = [0, 0, 0, -np.pi/2, np.pi/2, np.pi/2, -np.pi/2, -np.pi, 0, 0, -np.pi, -np.pi/2, np.pi/2, np.pi/2, -np.pi/2]
JOE_TURN_SPEED_STANCE = [0.7 for i in range(OSCILLATOR_NUMBERS)]
JOE_TURN_SPEED_SWING = [0.7 for i in range(OSCILLATOR_NUMBERS)]
JOE_TURN_DES_AMPLITUDE = [0.05, 0, 0.05, 0.025, 0.025, 0.025, 0.025, 0.3925, 0.3925, 0.3925, 0.3925, 0.2, 0.2, 0.2, 0.2]
JOE_TURN_DES_OFFSET = [0, 0, 0.1, 0.1, 0.1, 0.1, 0.1, 0, 0, 0, 0, 0, 0, 0, 0]
```

Figure 67: Turning Right Parameters

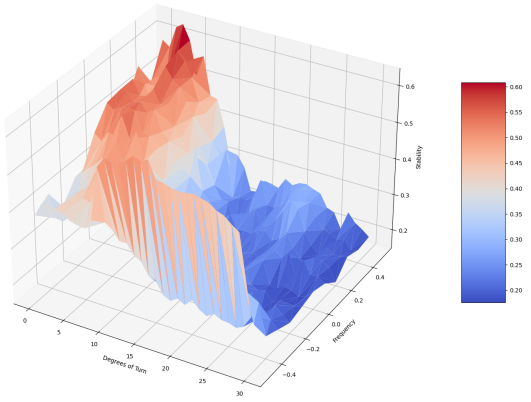
D Extra Results

D.1 Spine Frequency Experiments

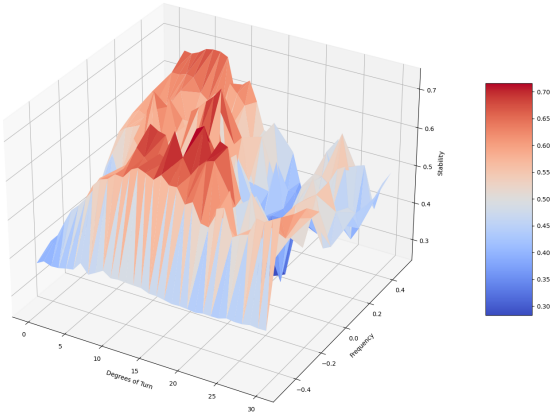
D.1.1 Part 1



(a) Pitch Angle Vs Frequency Vs Stability

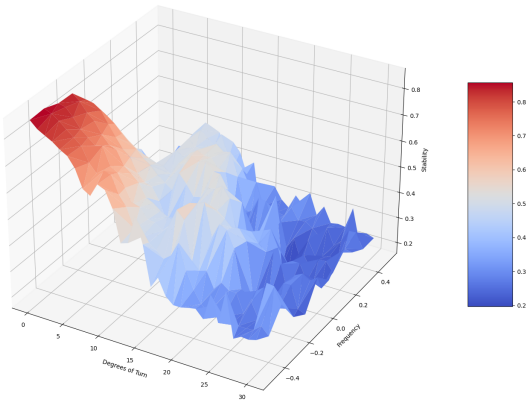


(b) Roll Angle Vs Frequency Vs Stability

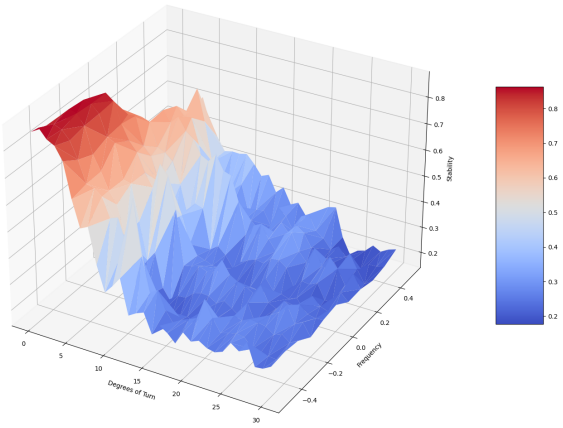


(c) Yaw Angle Vs Frequency Vs Stability

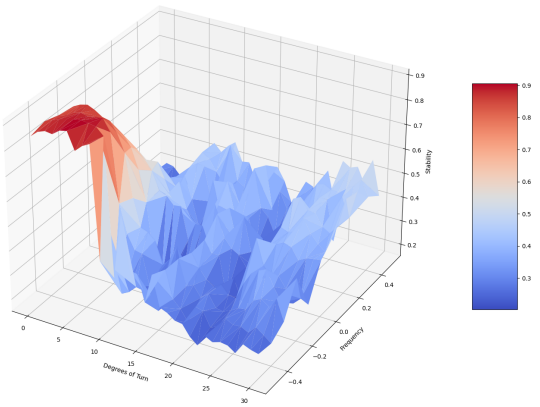
Figure 68: Walk Spine Frequency Experiment Results with friction



(a) Pitch Angle Vs Frequency Vs Stability



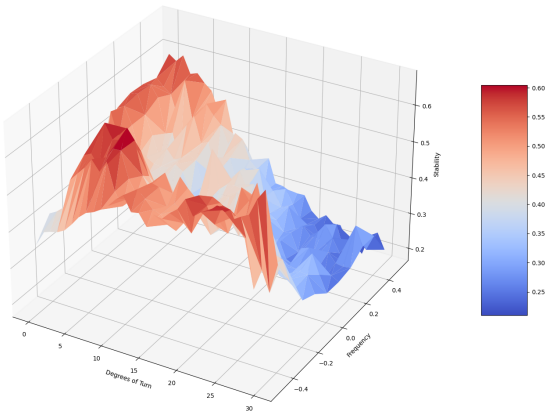
(b) Roll Angle Vs Frequency Vs Stability



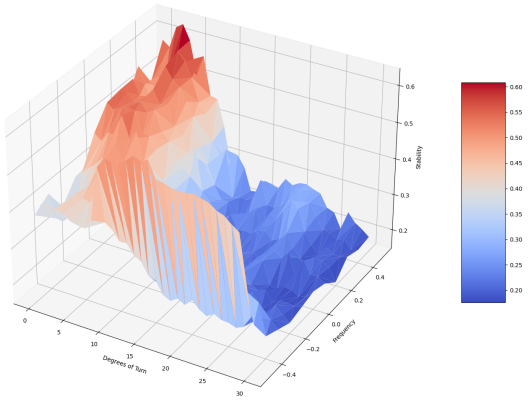
(c) Yaw Angle Vs Frequency Vs Stability

Figure 69: Trot Spine Frequency Experiment Results with friction

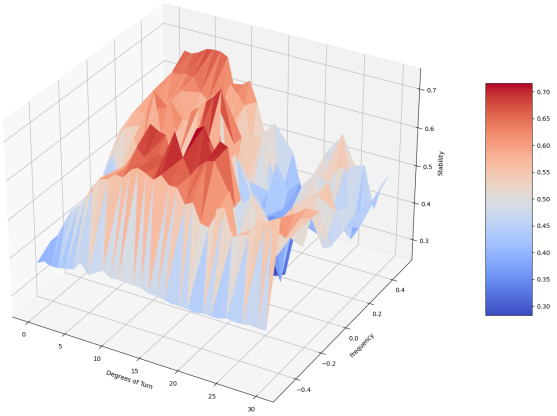
D.1.2 Part 2



(a) Pitch & Roll Angle Vs Frequency Vs Stability

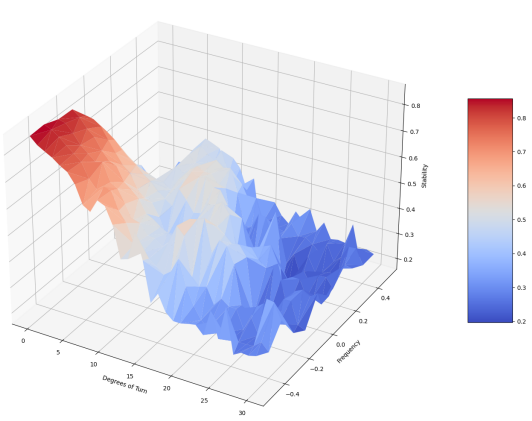


(b) Roll & Yaw Angle Vs Frequency Vs Stability

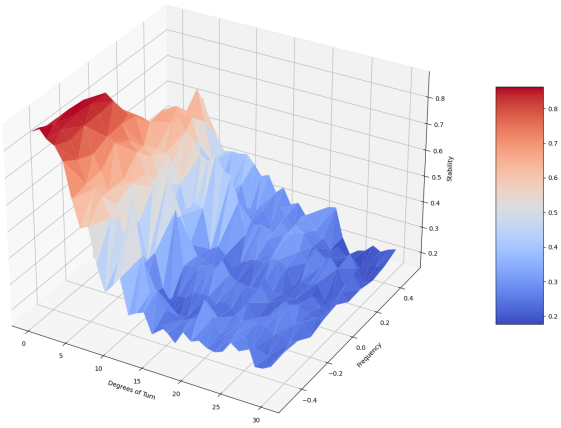


(c) Yaw & Pitch Angle Vs Frequency Vs Stability

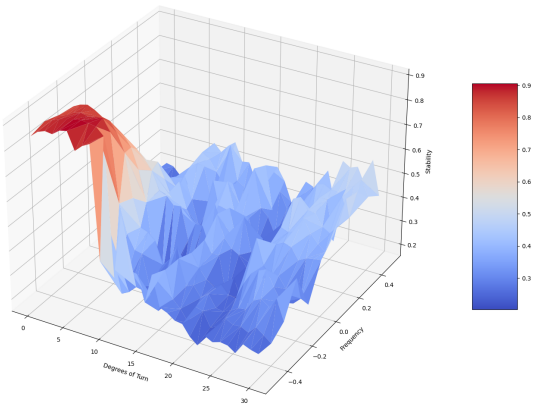
Figure 70: Walk Spine Frequency Experiment Results with friction



(a) Pitch & Roll Angle Vs Frequency Vs Stability



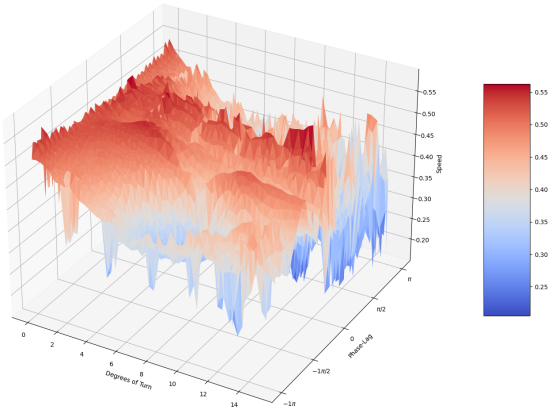
(b) Roll & Yaw Angle Vs Frequency Vs Stability



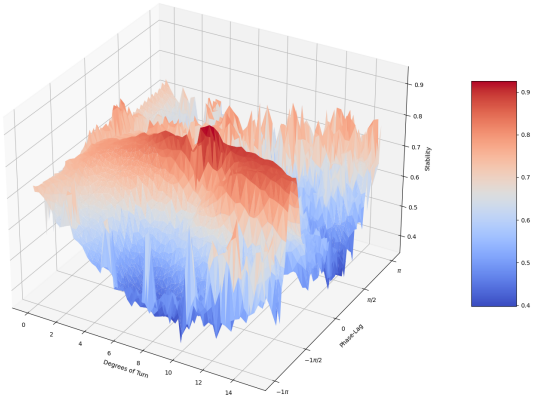
(c) Yaw & Pitch Angle Vs Frequency Vs Stability

Figure 71: Trot Spine Frequency Experiment Results with friction

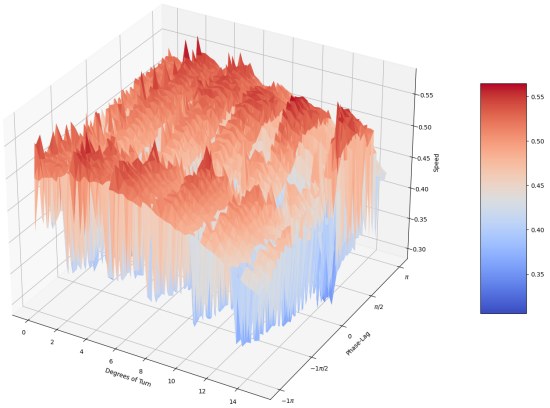
D.2 Spine Phase Experiments



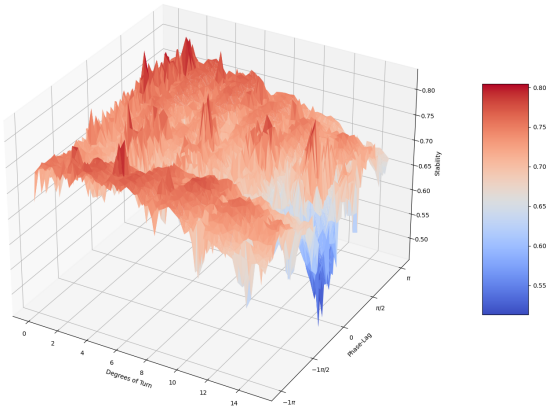
(a) Trot Roll: Phase vs Angle vs Speed



(b) Trot Roll: Phase vs Angle vs Stability



(c) Trot Yaw: Phase vs Angle vs Speed

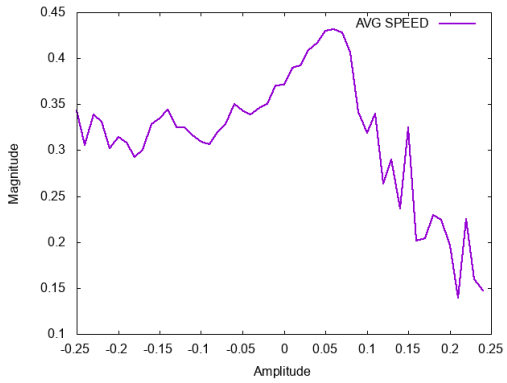


(d) Trot Yaw: Phase vs Angle vs Stability

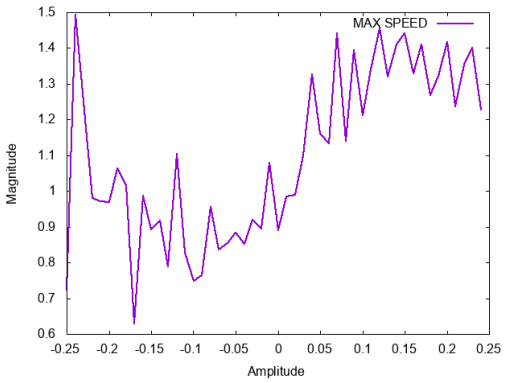
Figure 72: Phase-Lag vs Spine Angle Results with Friction

D.3 Parametric Sweeps

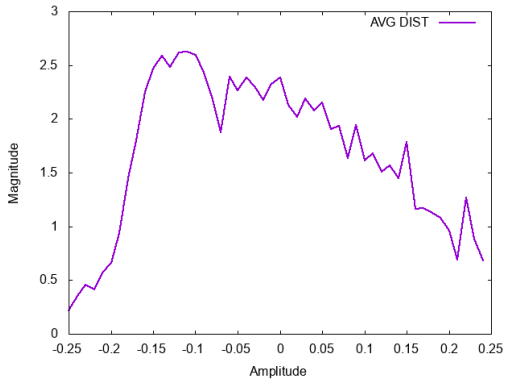
D.3.1 Full Body Sweeps



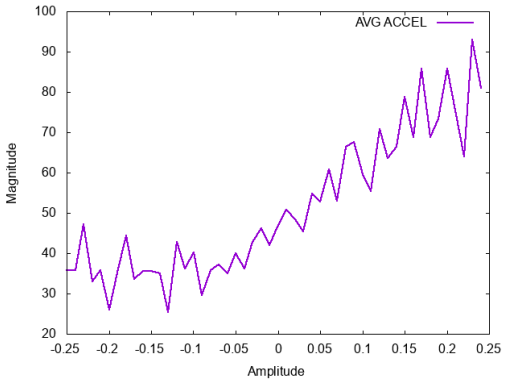
(a) Average Speed



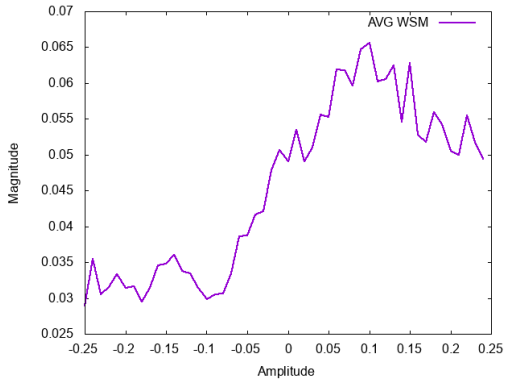
(b) Max Speed



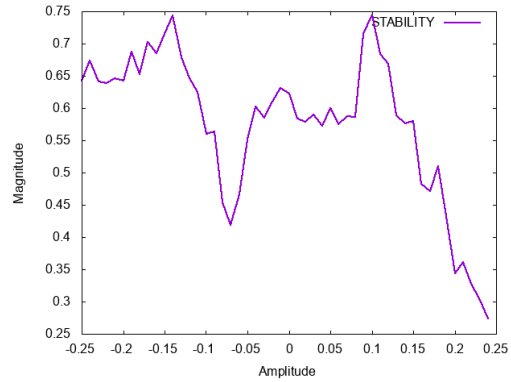
(c) Distance



(d) Average Acceleration

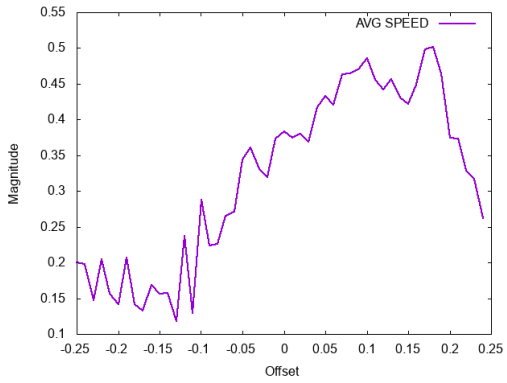


(e) Wide Stability Margin

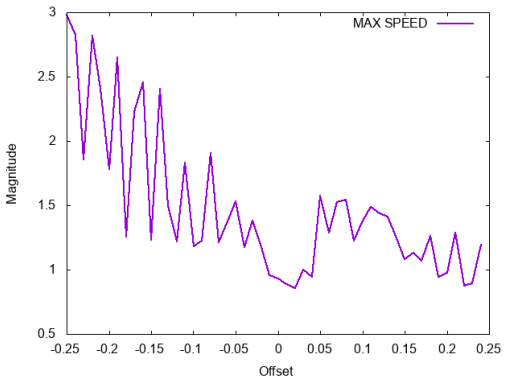


(f) Stability

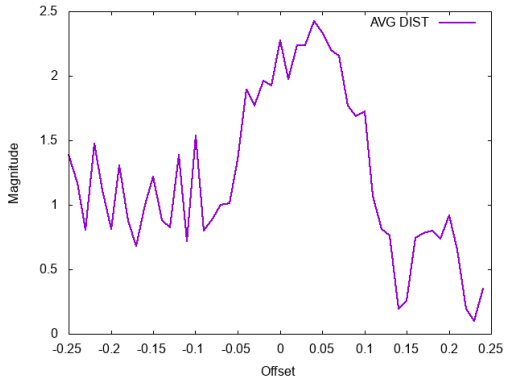
Figure 73: The effect of an amplitude full sweep on a walk gait with friction



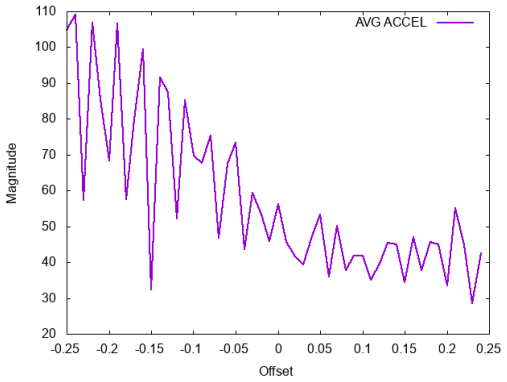
(a) Average Speed



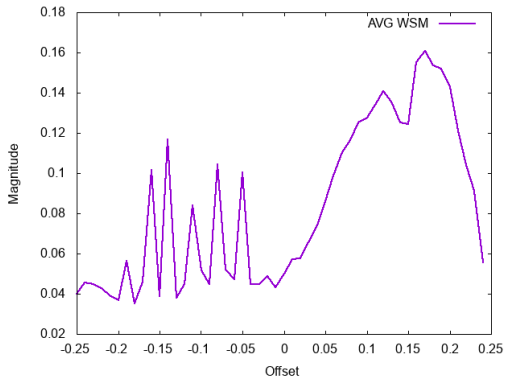
(b) Max Speed



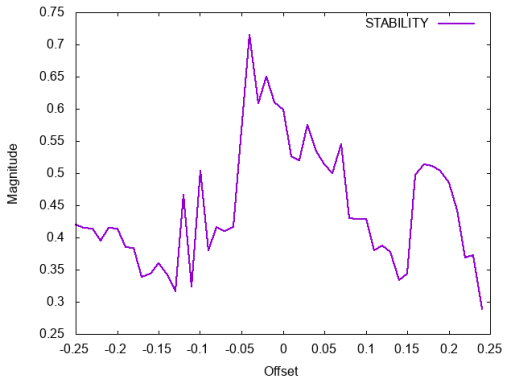
(c) Distance



(d) Average Acceleration

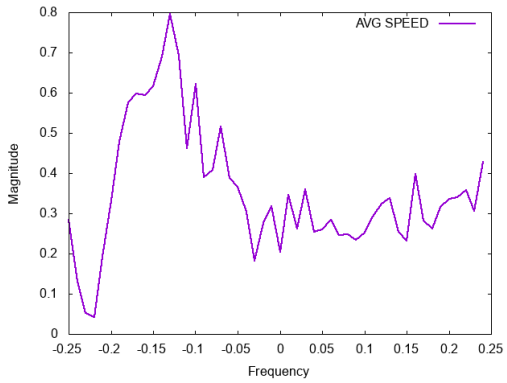


(e) Wide Stability Margin

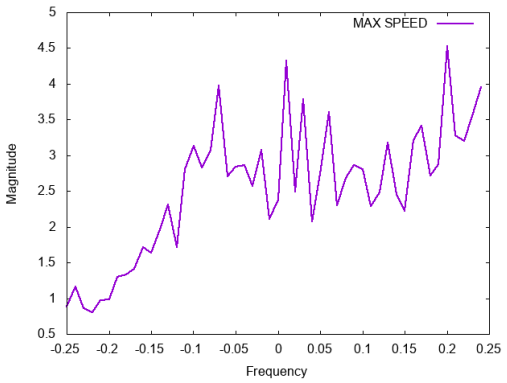


(f) Stability

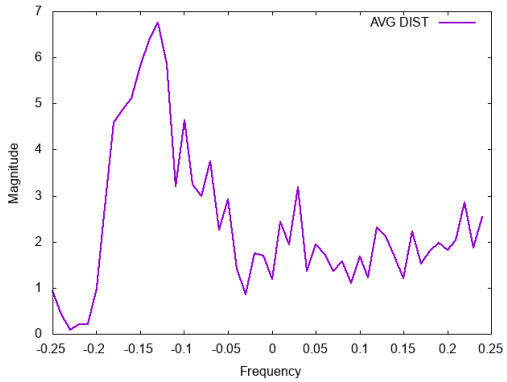
Figure 74: The effect of an offset full sweep on a walk gait with friction



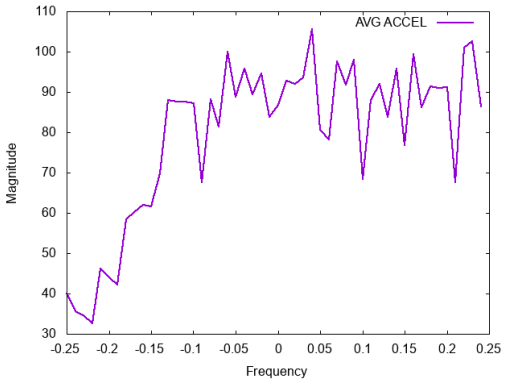
(a) Average Speed



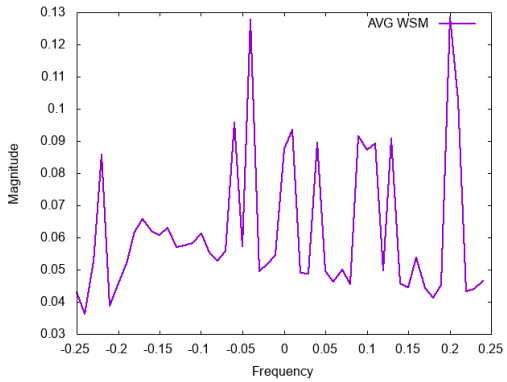
(b) Max Speed



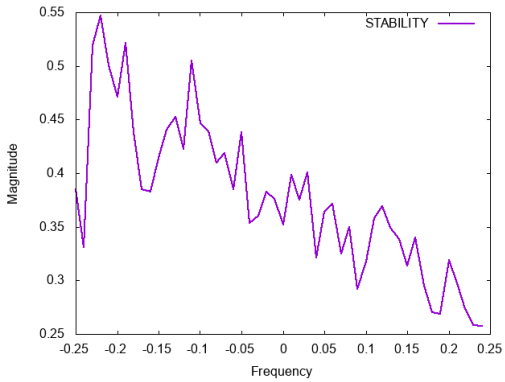
(c) Distance



(d) Average Acceleration

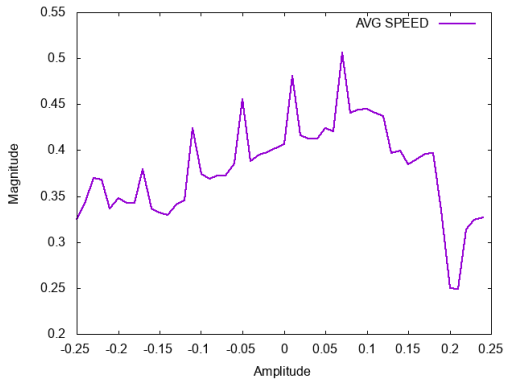


(e) Wide Stability Margin

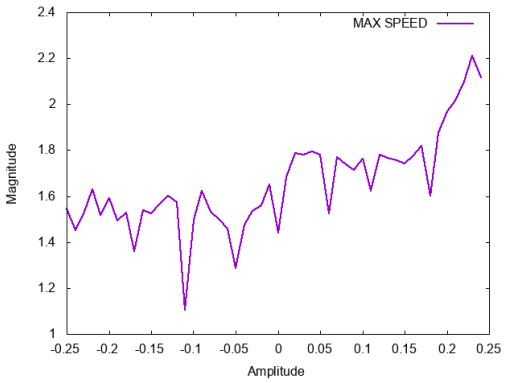


(f) Stability

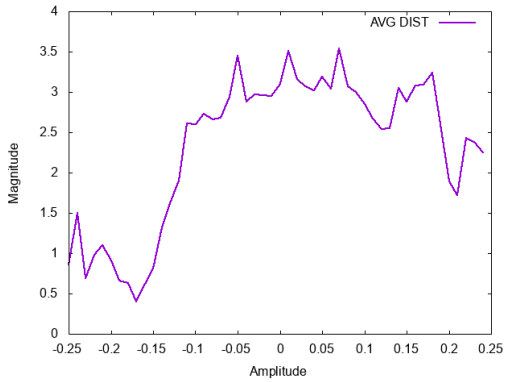
Figure 75: The effect of a frequency full sweep on a walk gait with friction



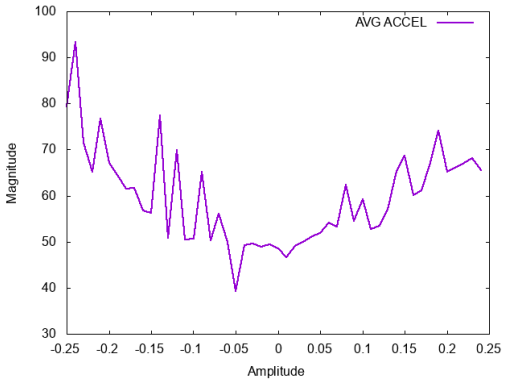
(a) Average Speed



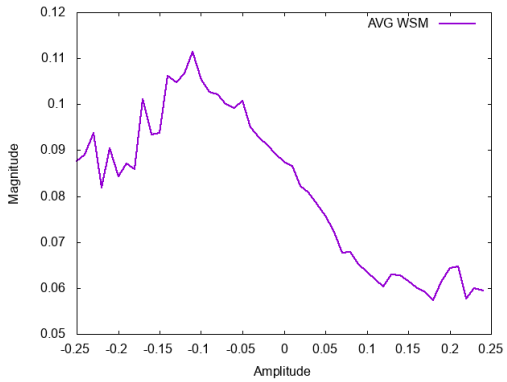
(b) Max Speed



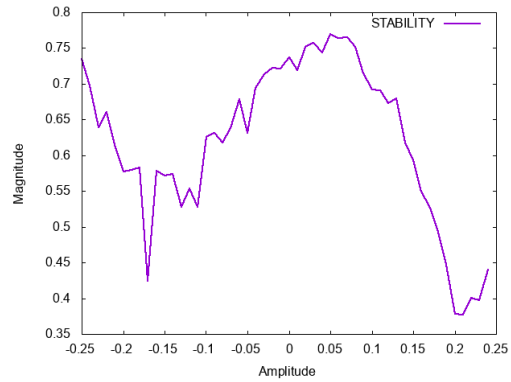
(c) Distance



(d) Average Acceleration

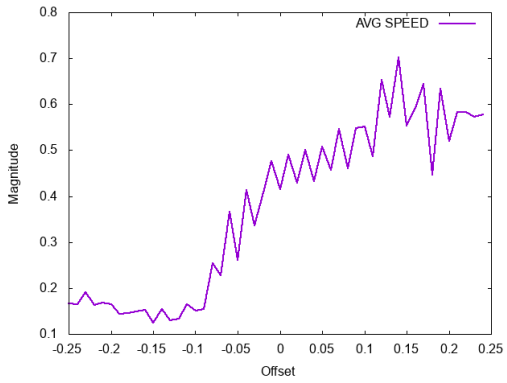


(e) Wide Stability Margin

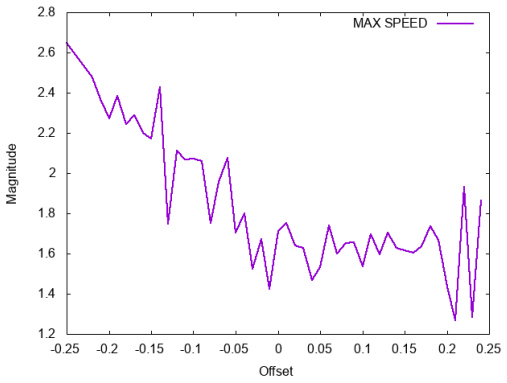


(f) Stability

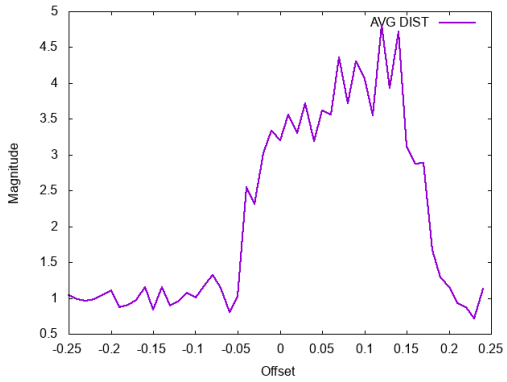
Figure 76: The effect of an amplitude full sweep on a trot gait with friction



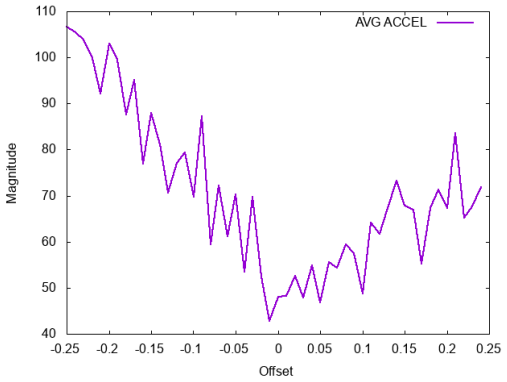
(a) Average Speed



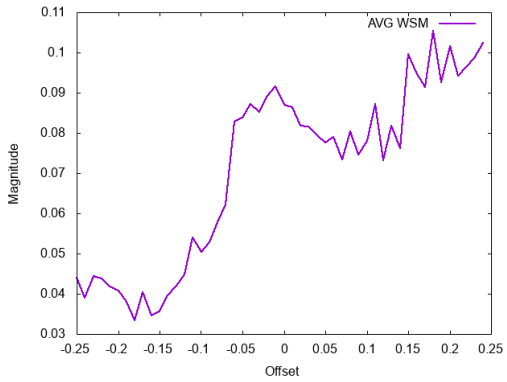
(b) Max Speed



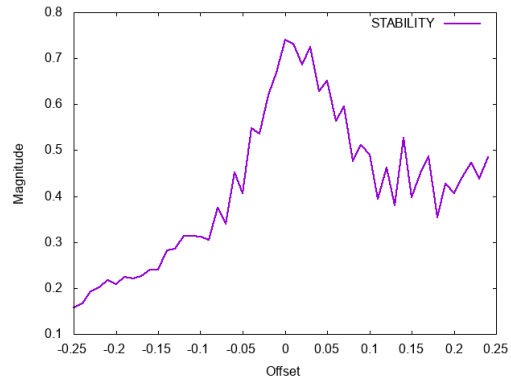
(c) Distance



(d) Average Acceleration

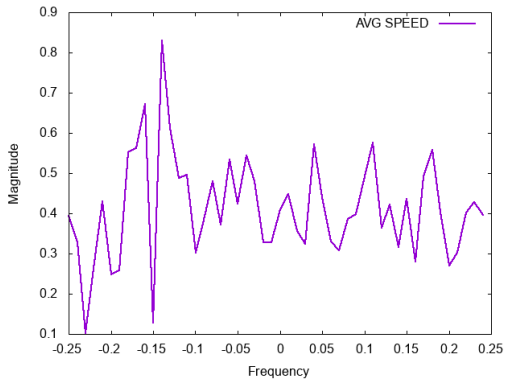


(e) Wide Stability Margin

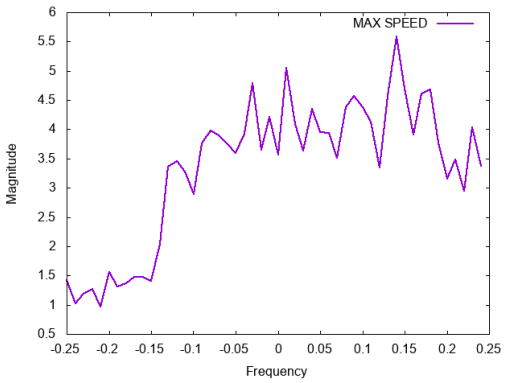


(f) Stability

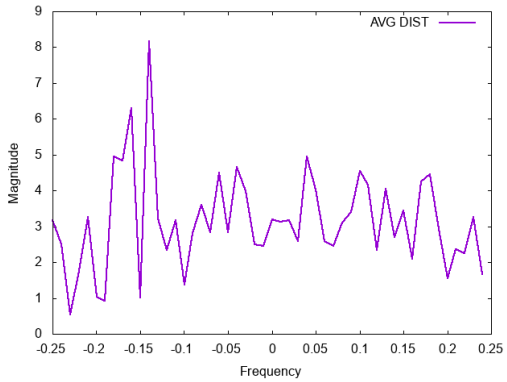
Figure 77: The effect of an offset full sweep on a trot gait with friction



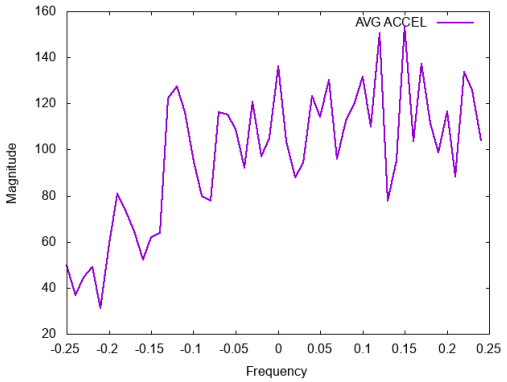
(a) Average Speed



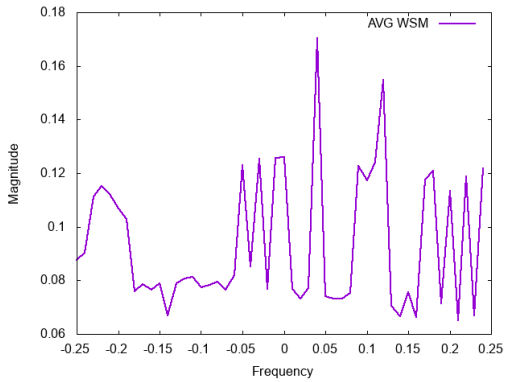
(b) Max Speed



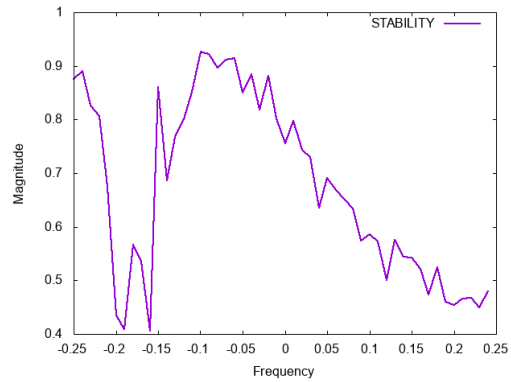
(c) Distance



(d) Average Acceleration



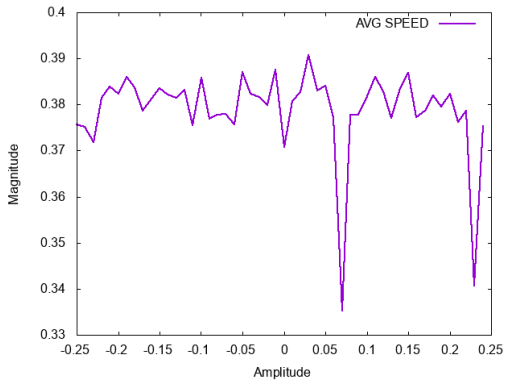
(e) Wide Stability Margin



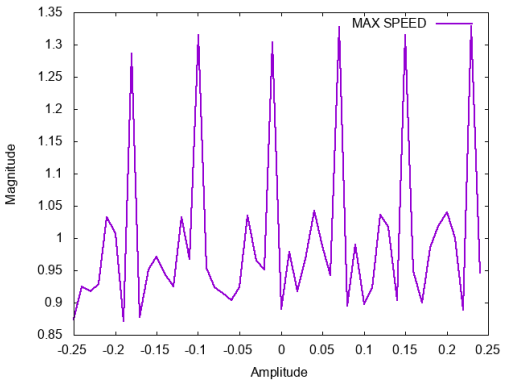
(f) Stability

Figure 78: The effect of a frequency full sweep on a trot gait with friction

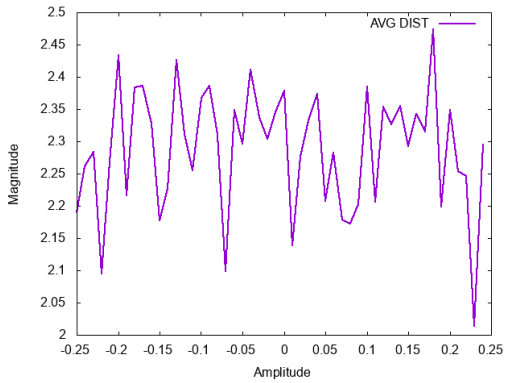
D.3.2 Leg Joint Sweeps



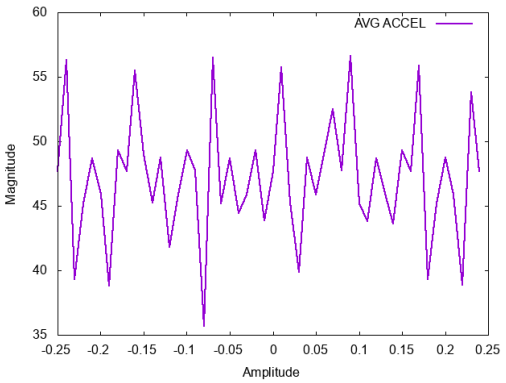
(a) Average Speed



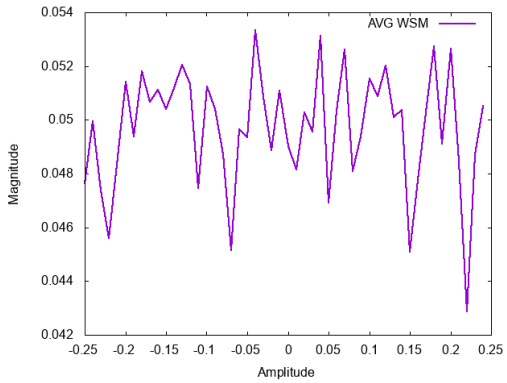
(b) Max Speed



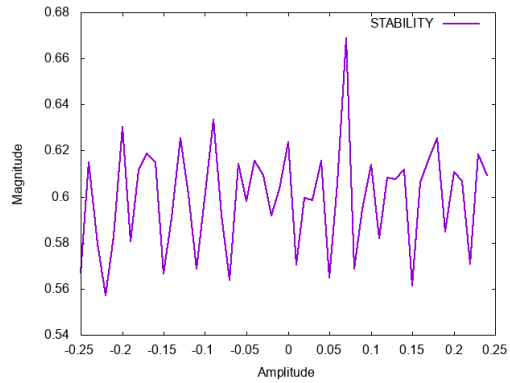
(c) Distance



(d) Average Acceleration

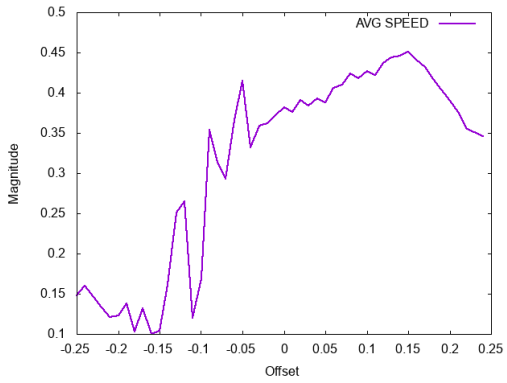


(e) Wide Stability Margin

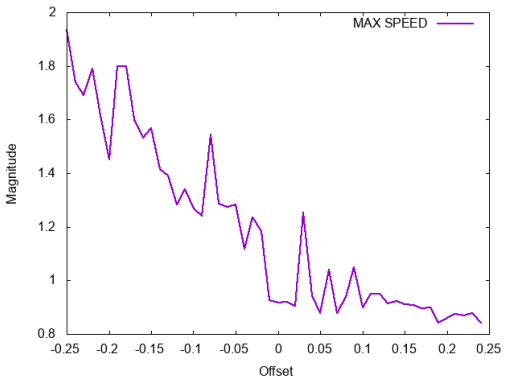


(f) Stability

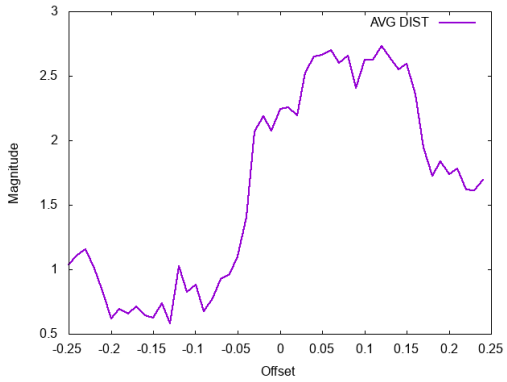
Figure 79: The effect of an amplitude leg sweep on a walk gait with friction



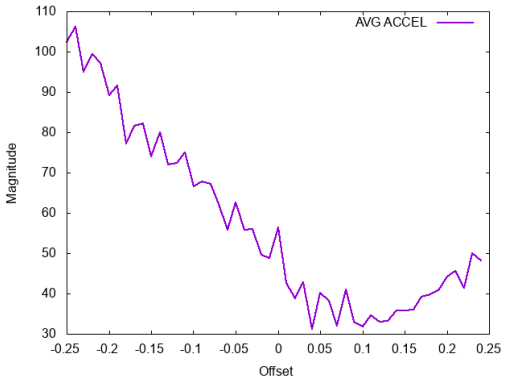
(a) Average Speed



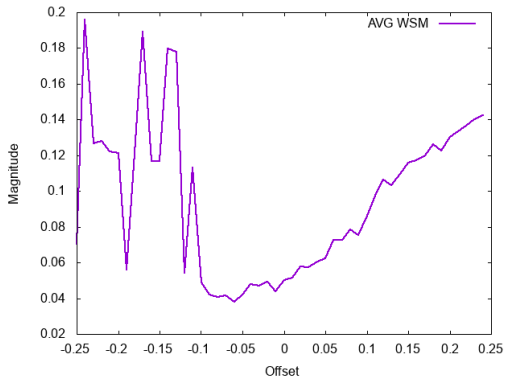
(b) Max Speed



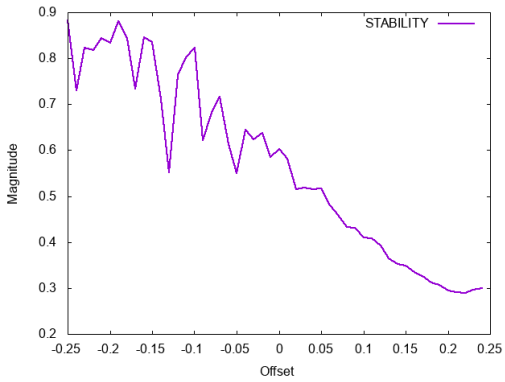
(c) Distance



(d) Average Acceleration

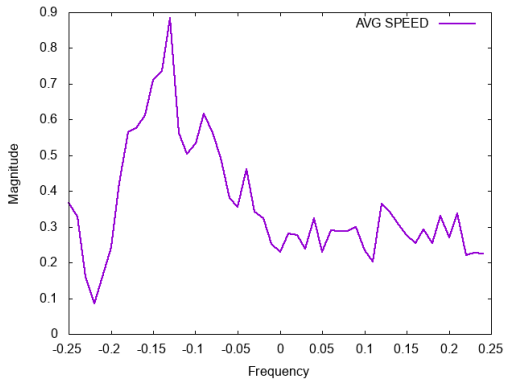


(e) Wide Stability Margin

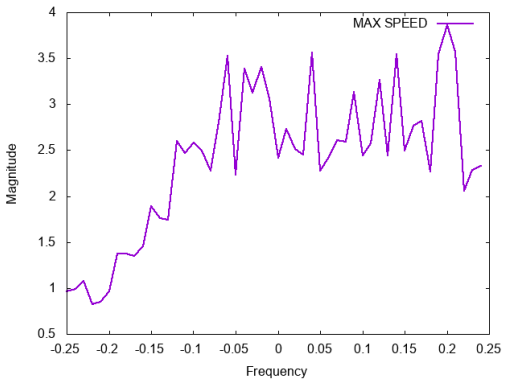


(f) Stability

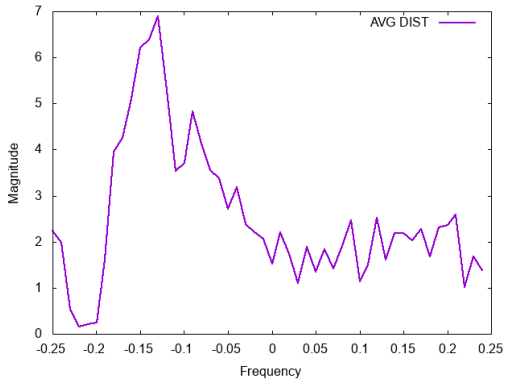
Figure 80: The effect of an offset leg sweep on a walk gait with friction



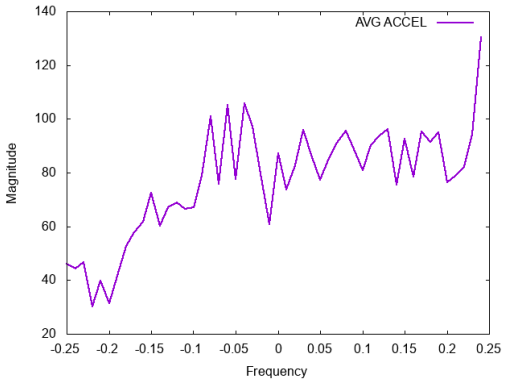
(a) Average Speed



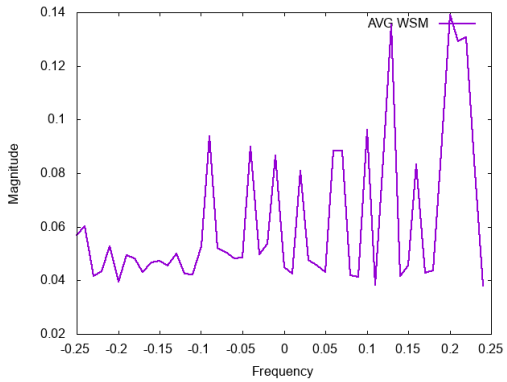
(b) Max Speed



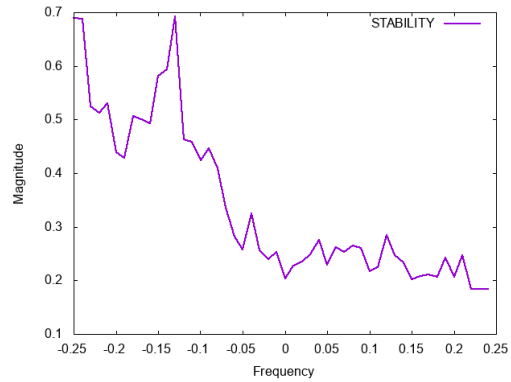
(c) Distance



(d) Average Acceleration

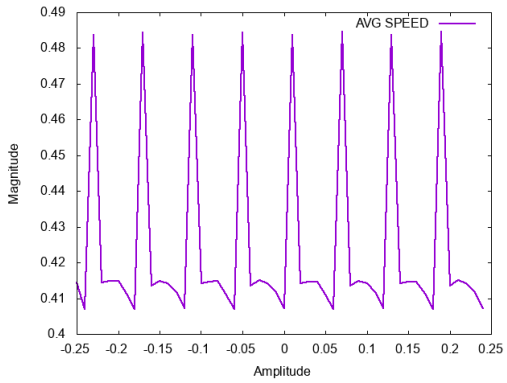


(e) Wide Stability Margin

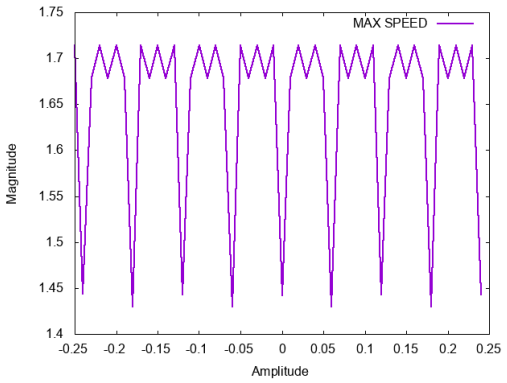


(f) Stability

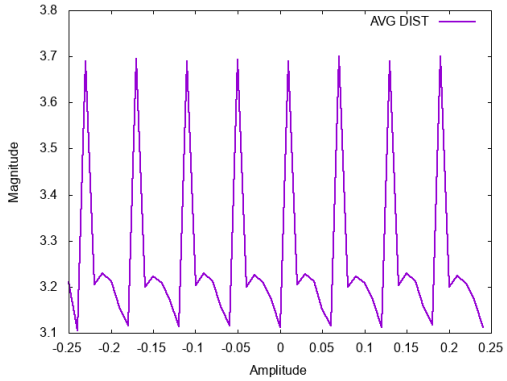
Figure 81: The effect of a frequency leg sweep on a walk gait with friction



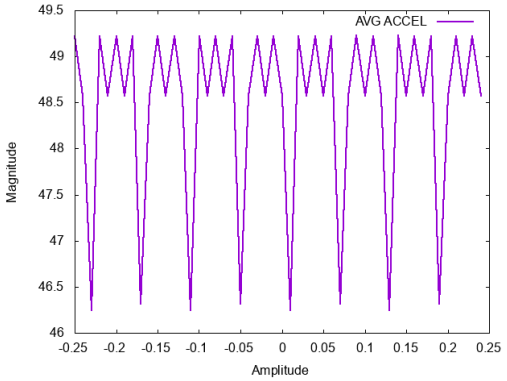
(a) Average Speed



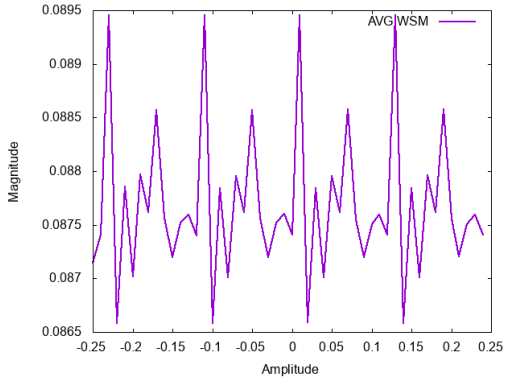
(b) Max Speed



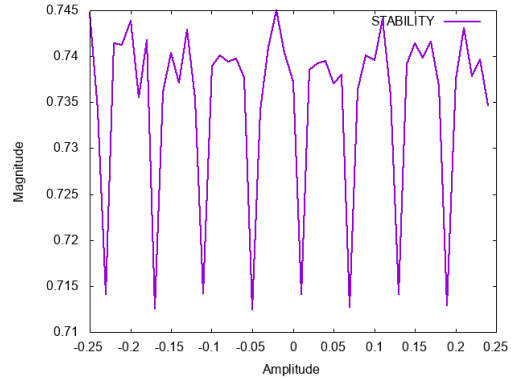
(c) Distance



(d) Average Acceleration

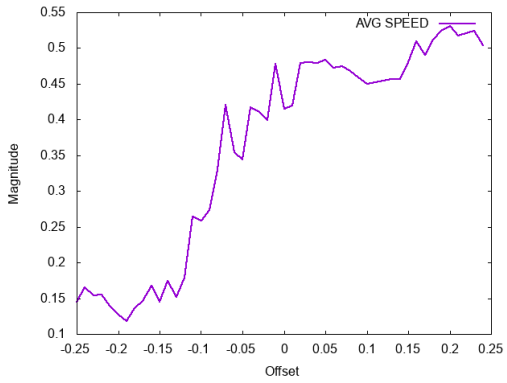


(e) Wide Stability Margin

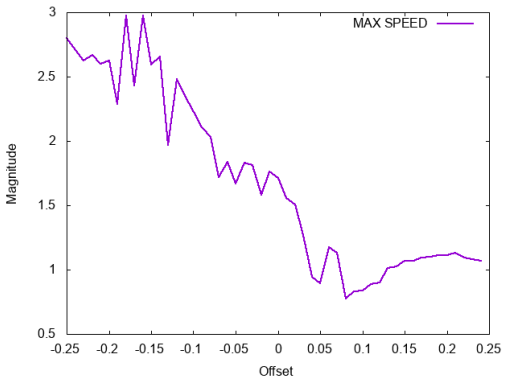


(f) Stability

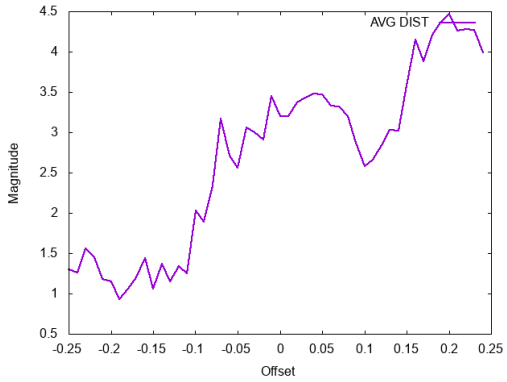
Figure 82: The effect of an amplitude leg sweep on a trot gait with friction



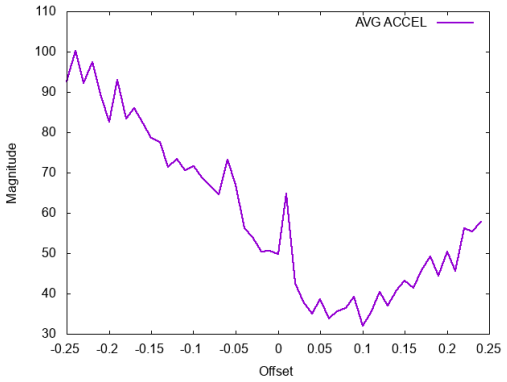
(a) Average Speed



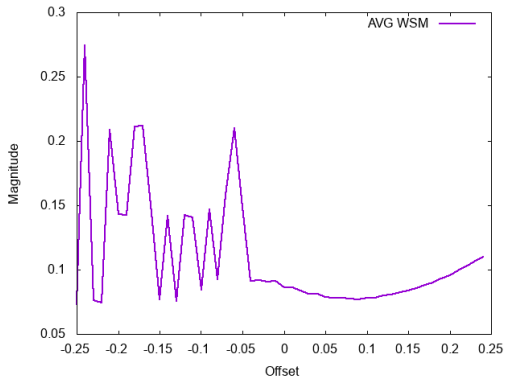
(b) Max Speed



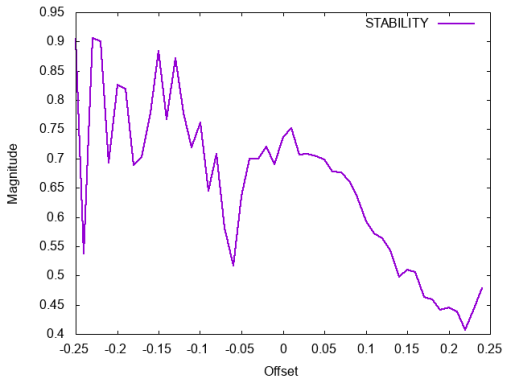
(c) Distance



(d) Average Acceleration

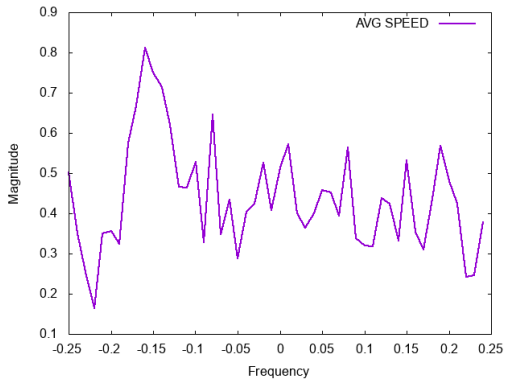


(e) Wide Stability Margin

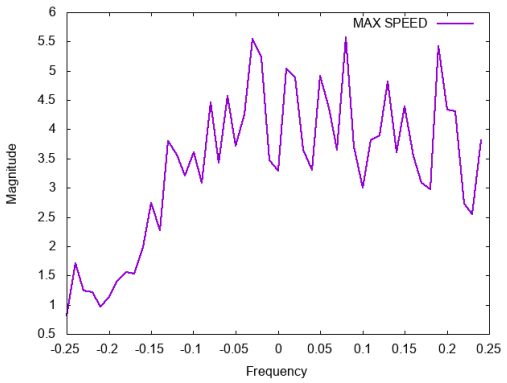


(f) Stability

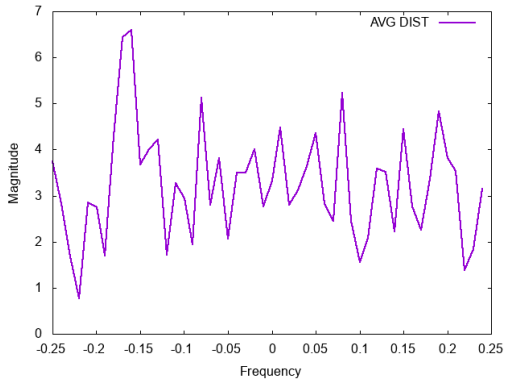
Figure 83: The effect of an offset leg sweep on a trot gait with friction



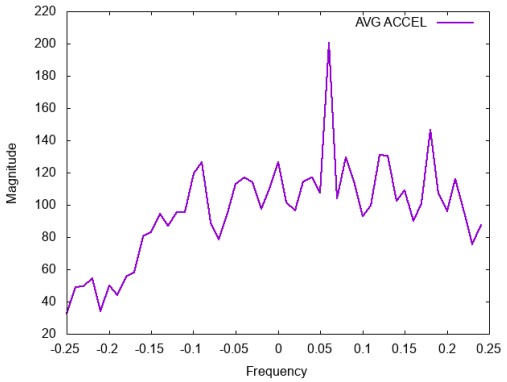
(a) Average Speed



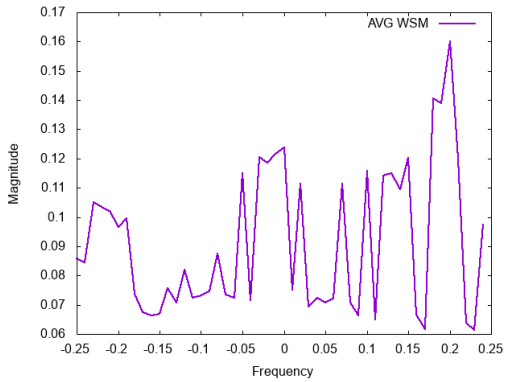
(b) Max Speed



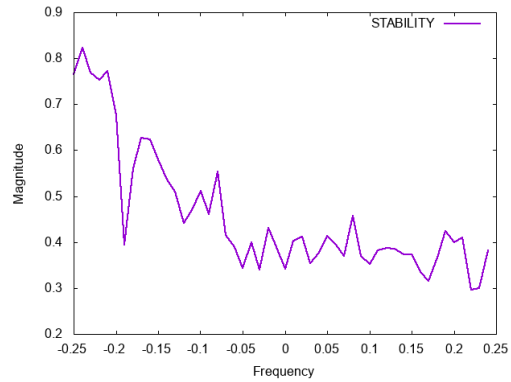
(c) Distance



(d) Average Acceleration



(e) Wide Stability Margin

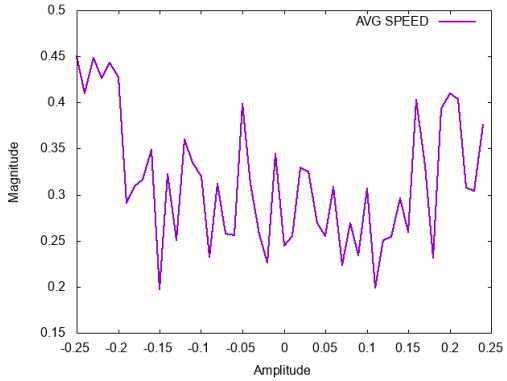


(f) Stability

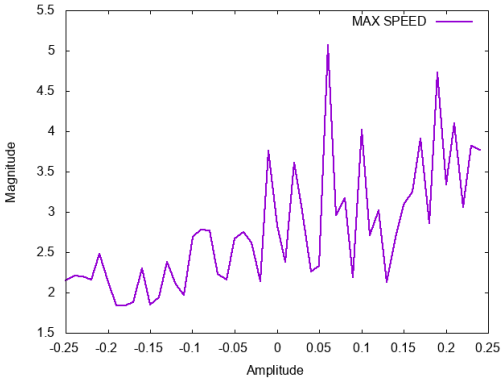
Figure 84: The effect of a frequency leg sweep on a trot gait with friction

D.3.3 Independent Joint Sweeps

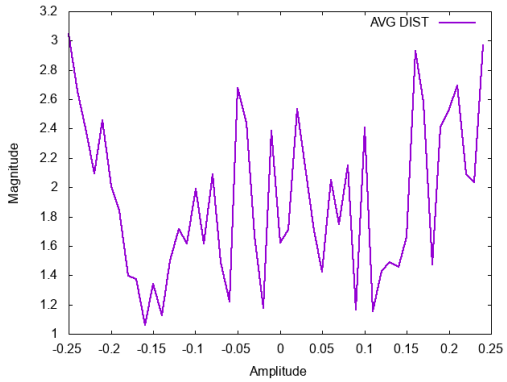
D.3.3.1 Hip



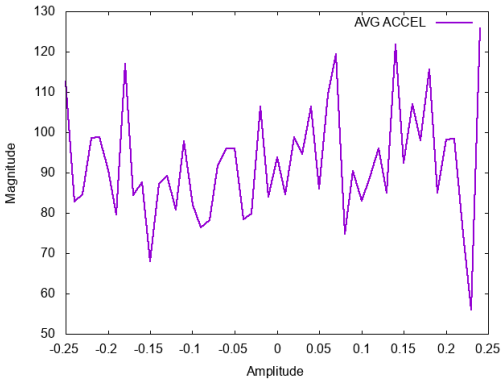
(a) Average Speed



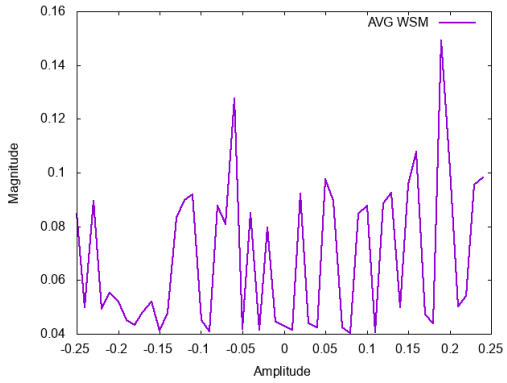
(b) Max Speed



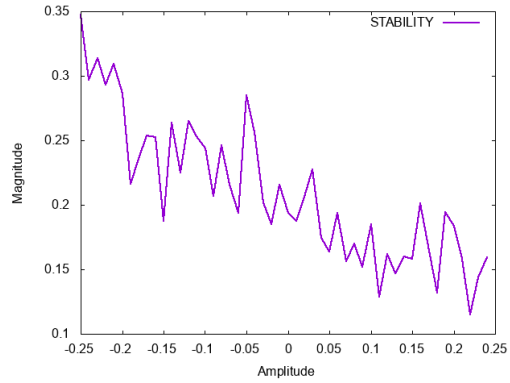
(c) Distance



(d) Average Acceleration

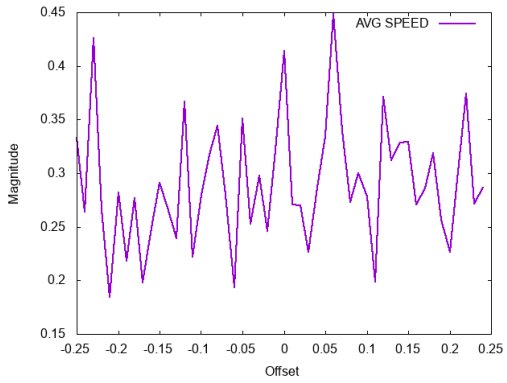


(e) Wide Stability Margin

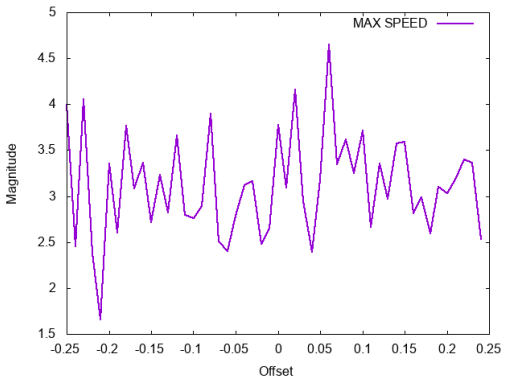


(f) Stability

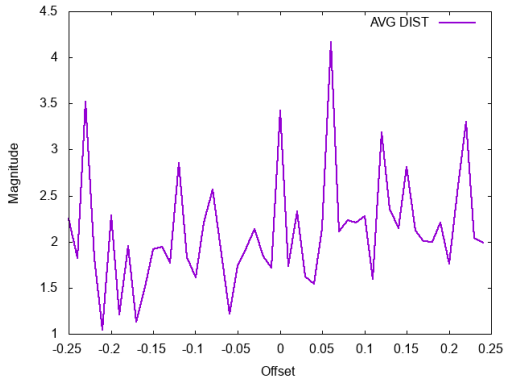
Figure 85: The effect of an amplitude sweep on a walk gaits hip joints with friction



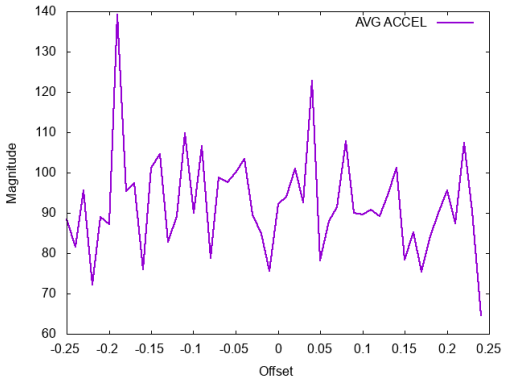
(a) Average Speed



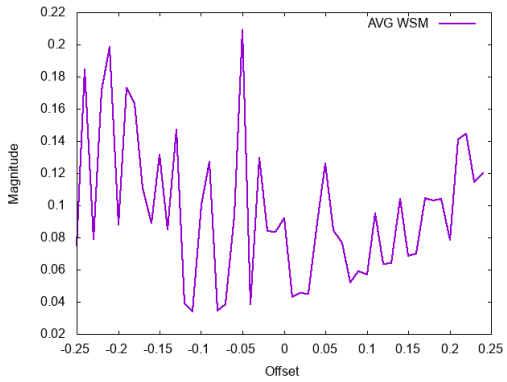
(b) Max Speed



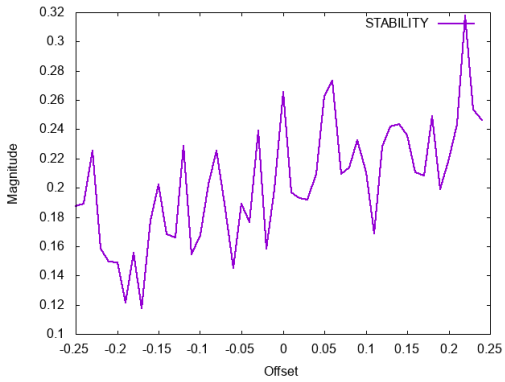
(c) Distance



(d) Average Acceleration

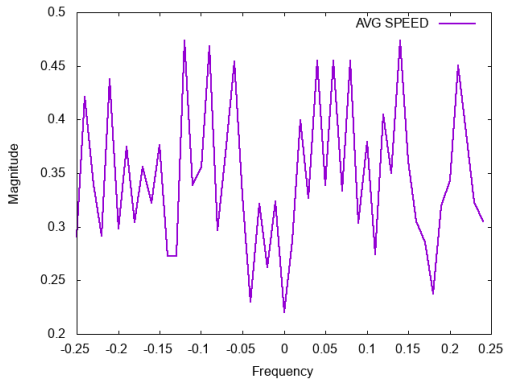


(e) Wide Stability Margin

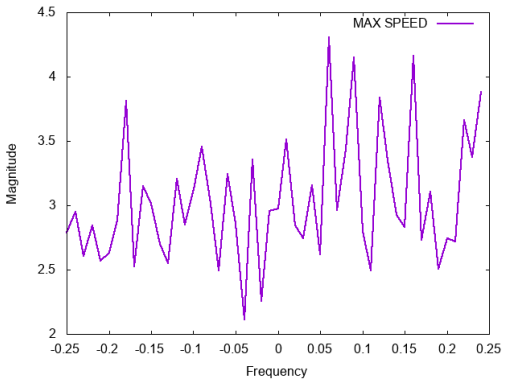


(f) Stability

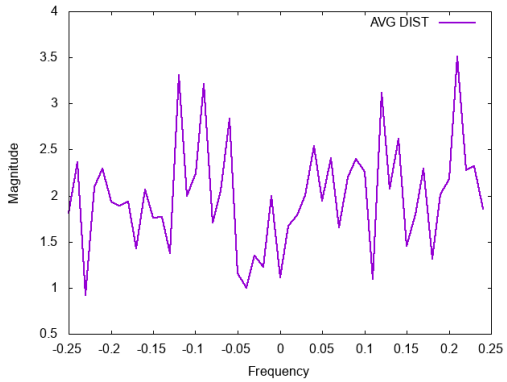
Figure 86: The effect of an offset sweep on a walk gaits hip joints with friction



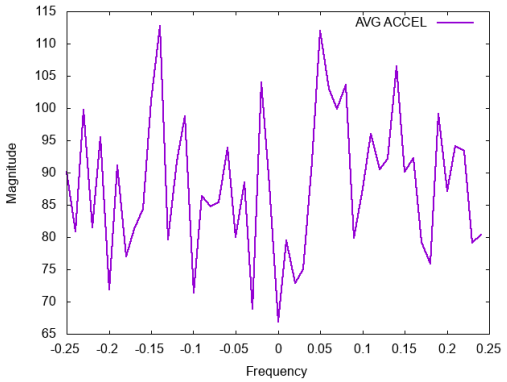
(a) Average Speed



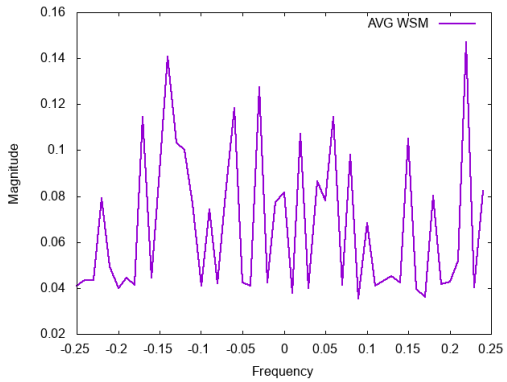
(b) Max Speed



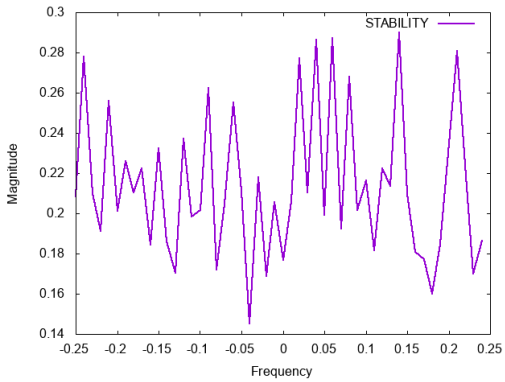
(c) Distance



(d) Average Acceleration

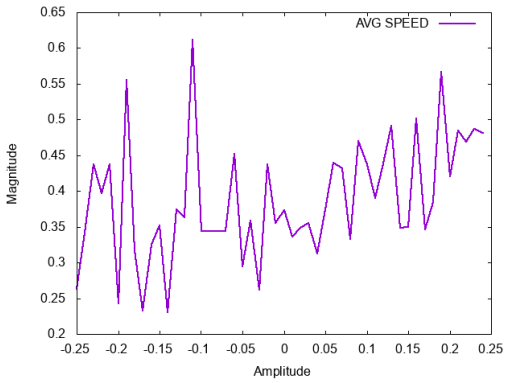


(e) Wide Stability Margin

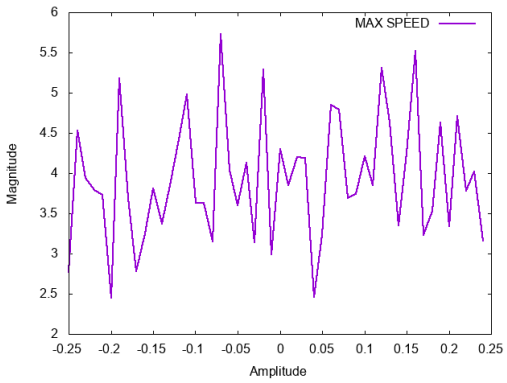


(f) Stability

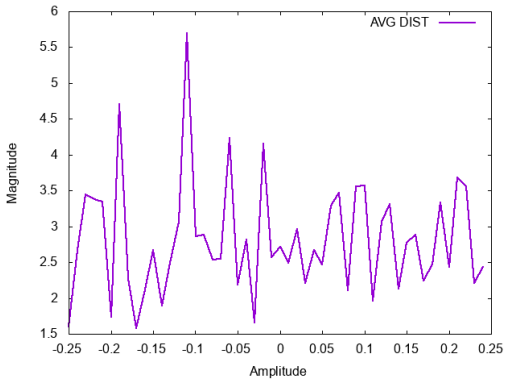
Figure 87: The effect of a frequency sweep on a walk gaits hip joints with friction



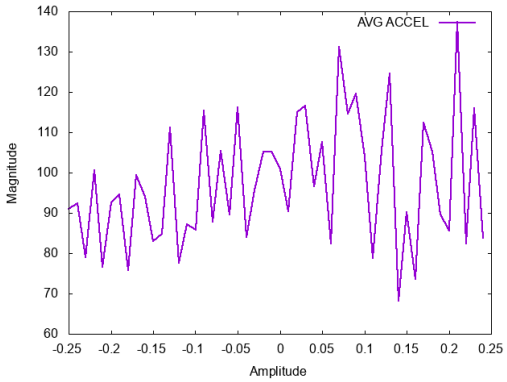
(a) Average Speed



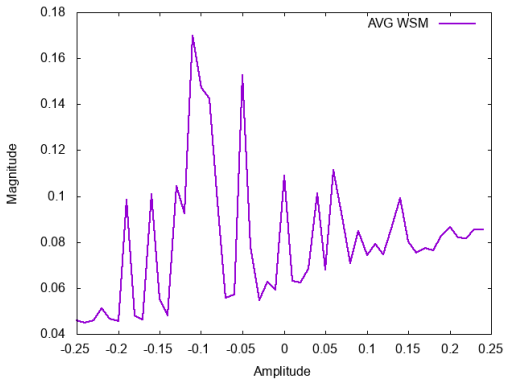
(b) Max Speed



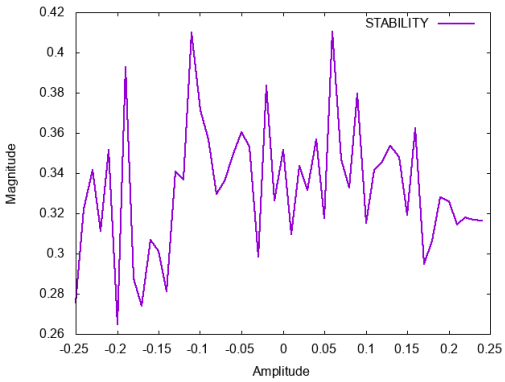
(c) Distance



(d) Average Acceleration

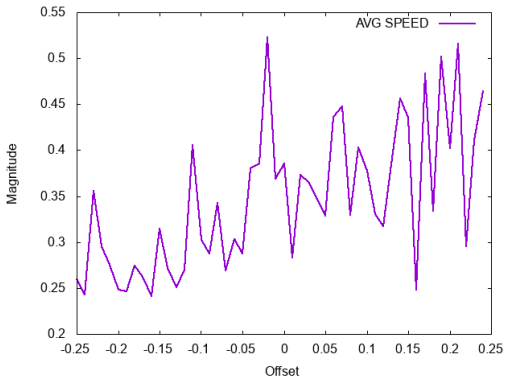


(e) Wide Stability Margin

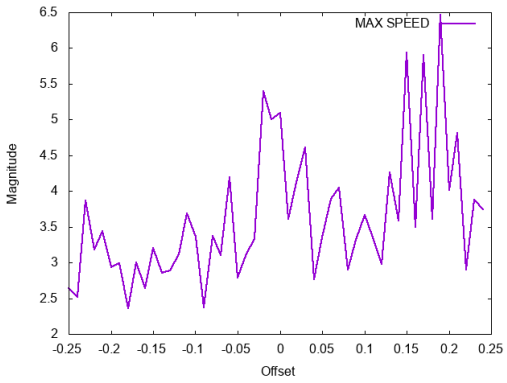


(f) Stability

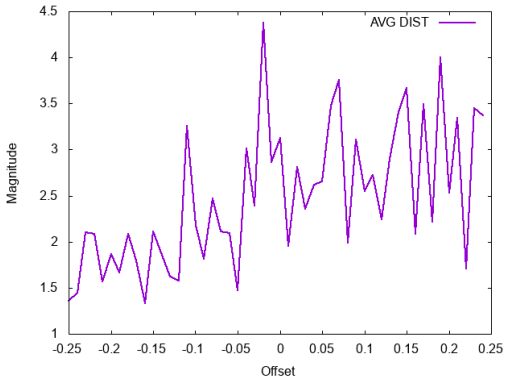
Figure 88: The effect of an amplitude sweep on a trot gaits hip joints with friction



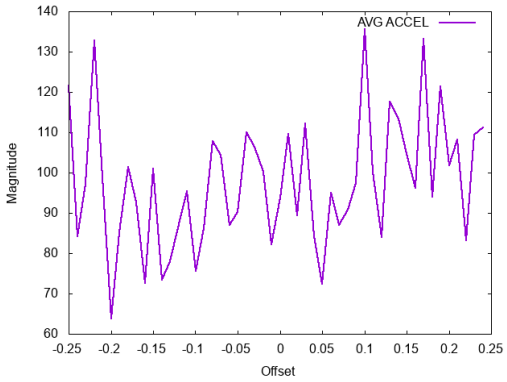
(a) Average Speed



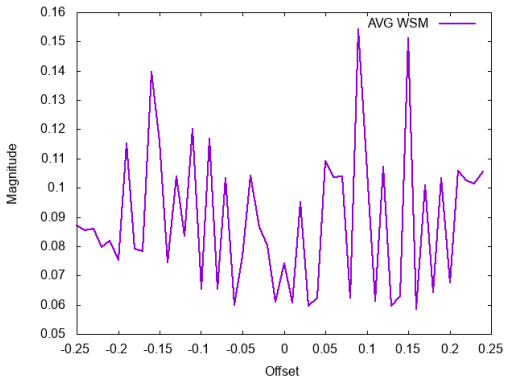
(b) Max Speed



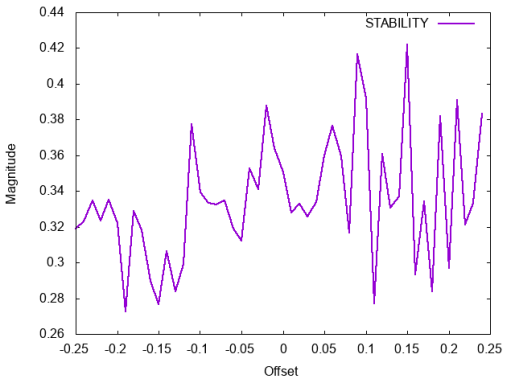
(c) Distance



(d) Average Acceleration

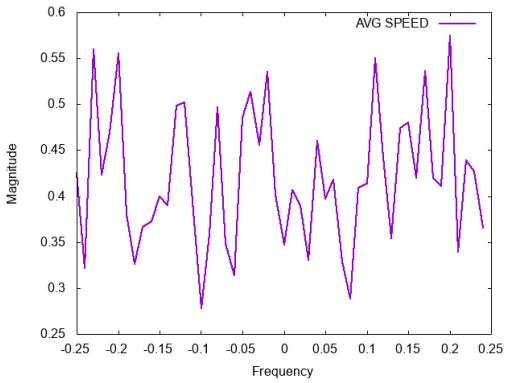


(e) Wide Stability Margin

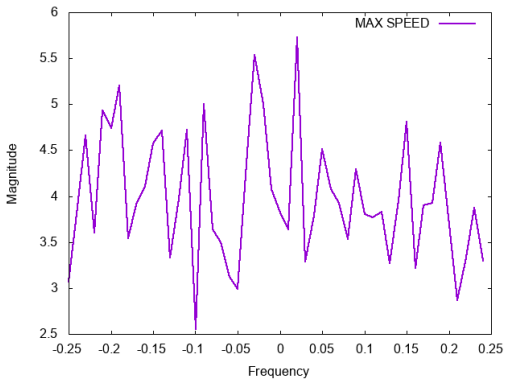


(f) Stability

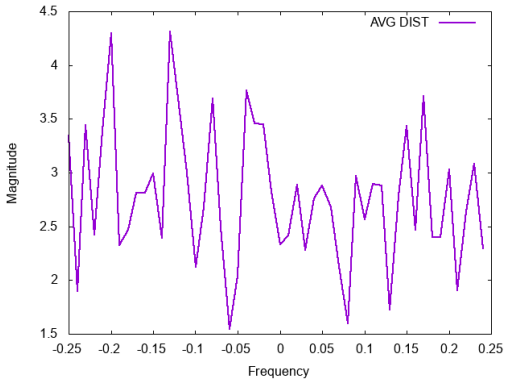
Figure 89: The effect of an offset sweep on a trot gaits hip joints with friction



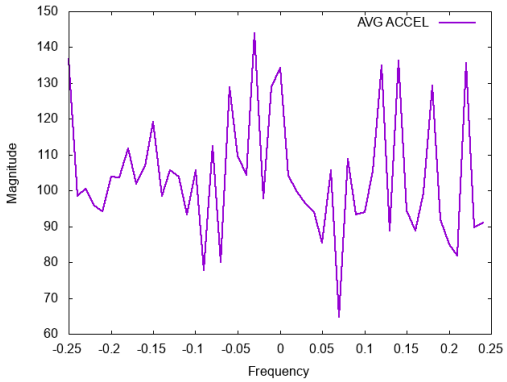
(a) Average Speed



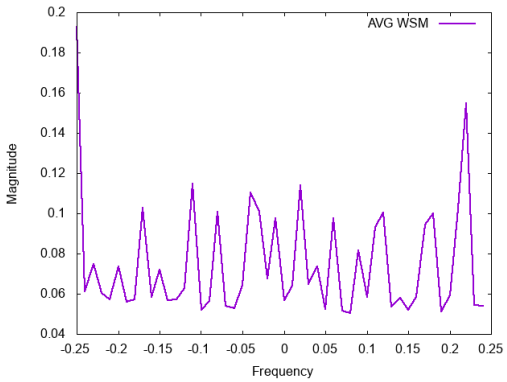
(b) Max Speed



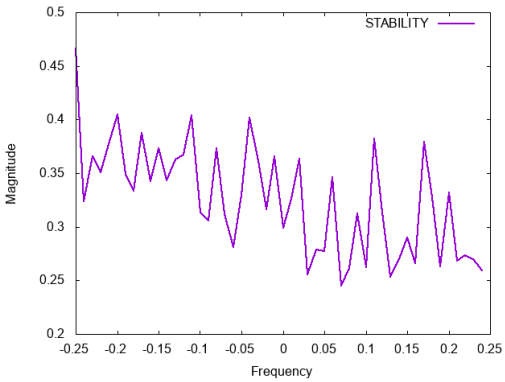
(c) Distance



(d) Average Acceleration



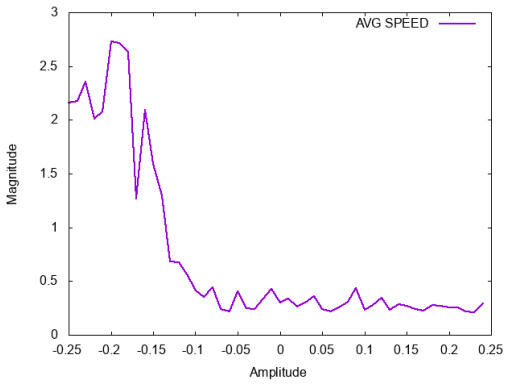
(e) Wide Stability Margin



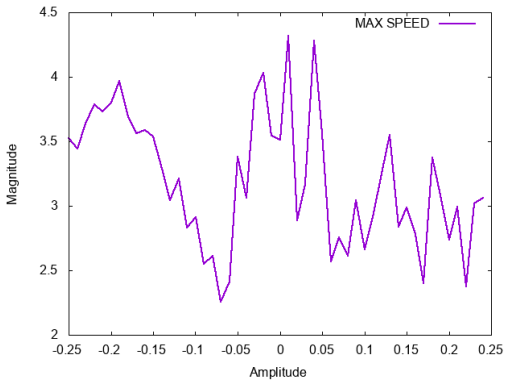
(f) Stability

Figure 90: The effect of a frequency sweep on a trot gaits hip joints with friction

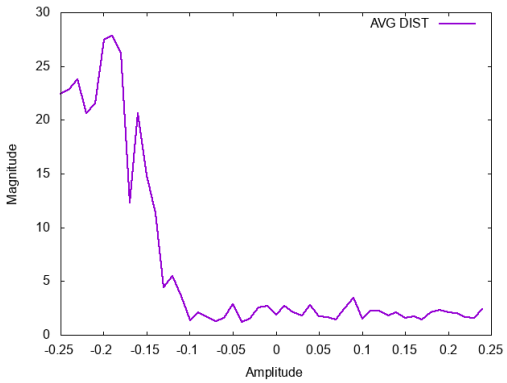
D.3.3.2 Knee



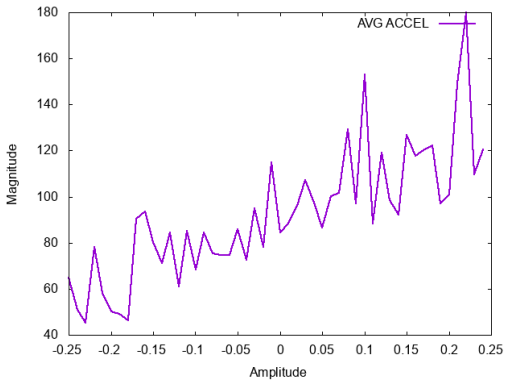
(a) Average Speed



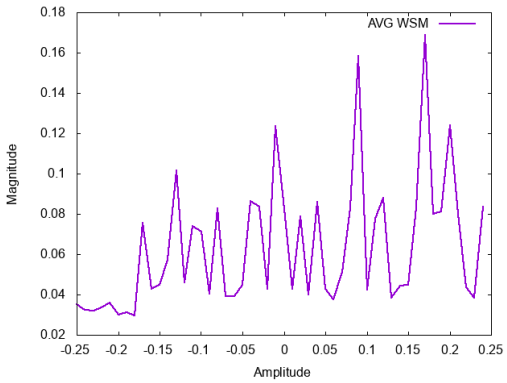
(b) Max Speed



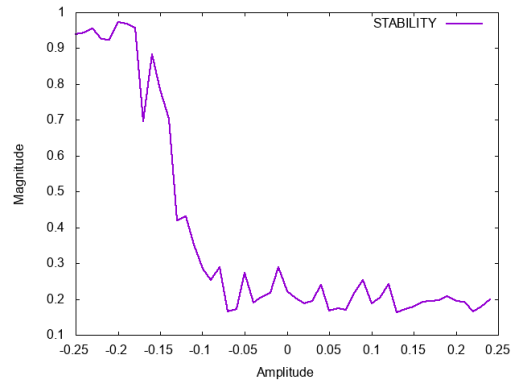
(c) Distance



(d) Average Acceleration

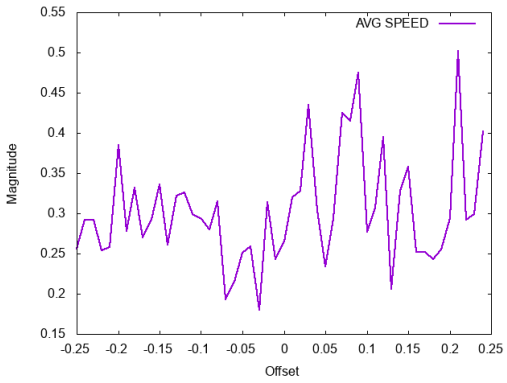


(e) Wide Stability Margin

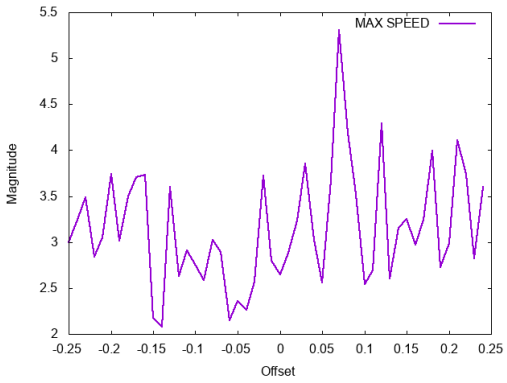


(f) Stability

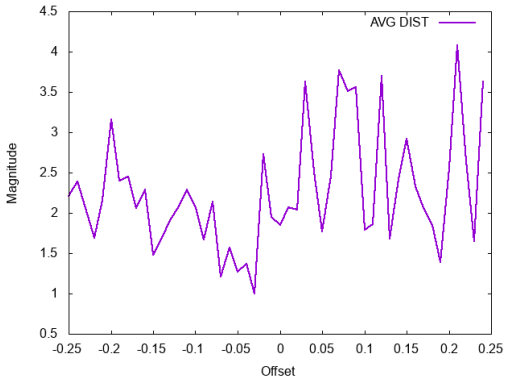
Figure 91: The effect of an amplitude sweep on a walk gaits knee joints with friction



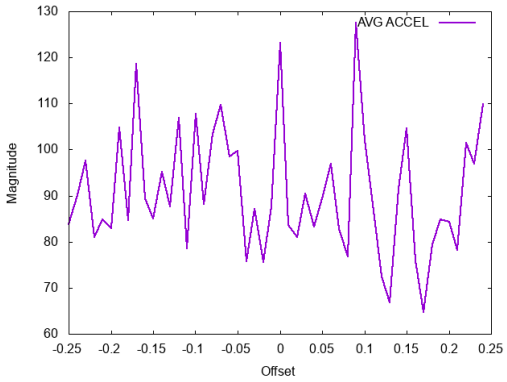
(a) Average Speed



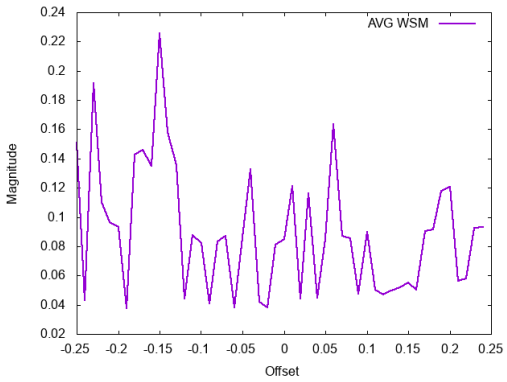
(b) Max Speed



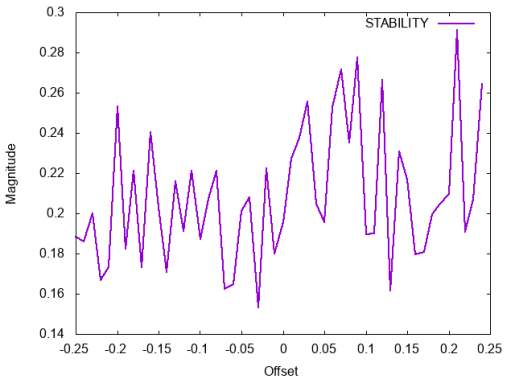
(c) Distance



(d) Average Acceleration

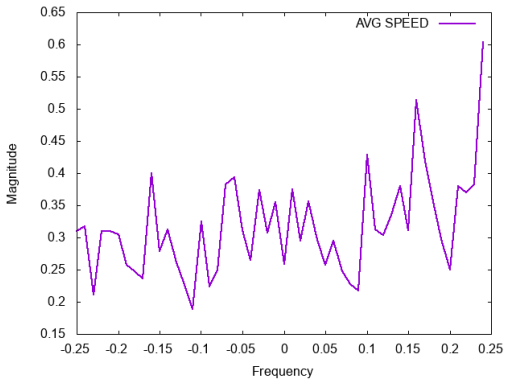


(e) Wide Stability Margin

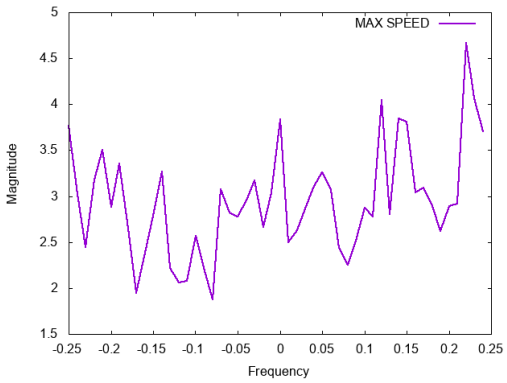


(f) Stability

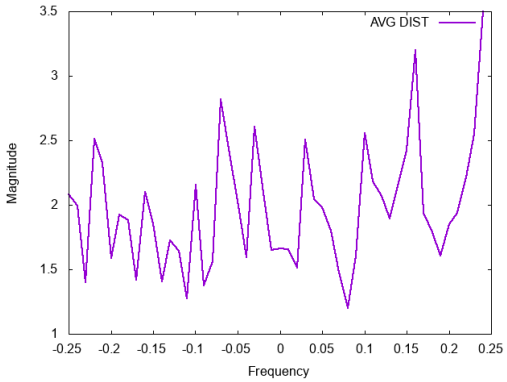
Figure 92: The effect of an offset sweep on a walk gaits knee joints with friction



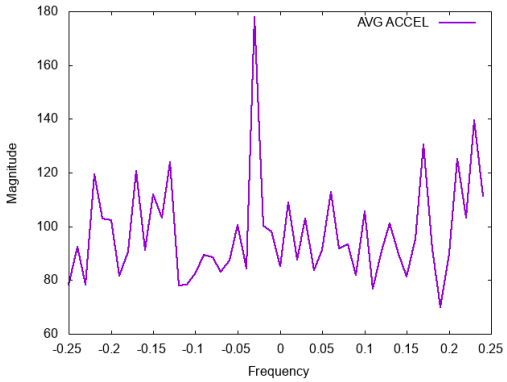
(a) Average Speed



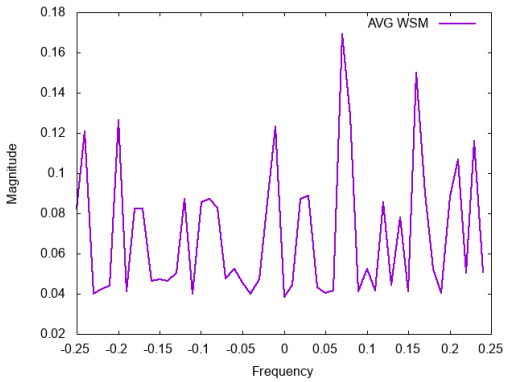
(b) Max Speed



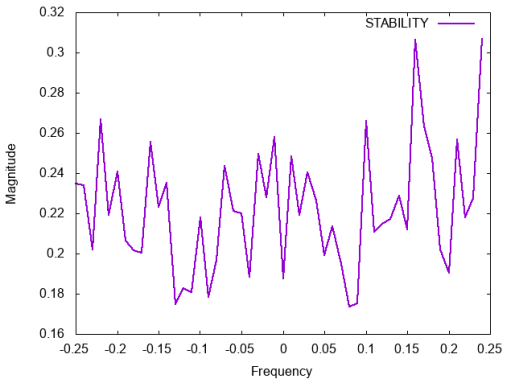
(c) Distance



(d) Average Acceleration

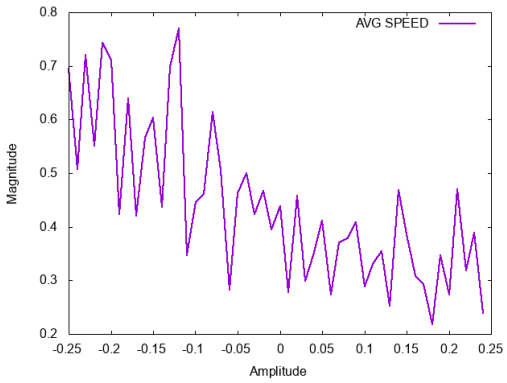


(e) Wide Stability Margin

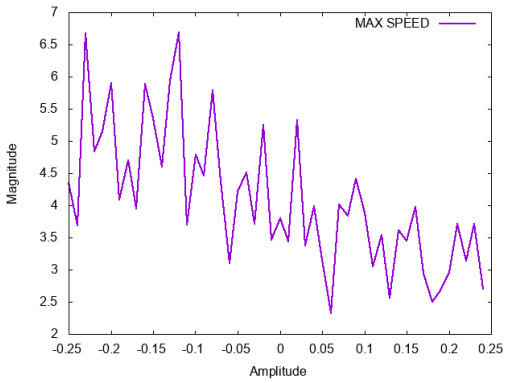


(f) Stability

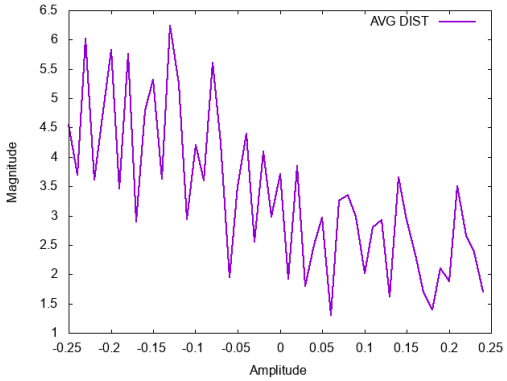
Figure 93: The effect of a frequency sweep on a walk gaits knee joints with friction



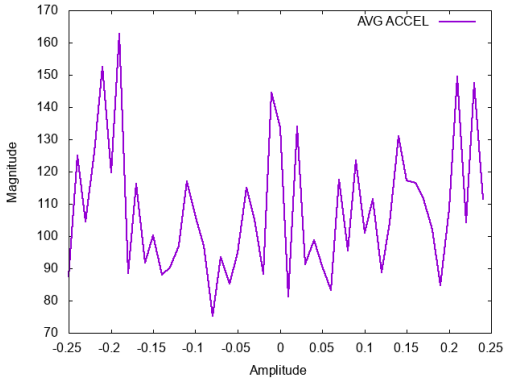
(a) Average Speed



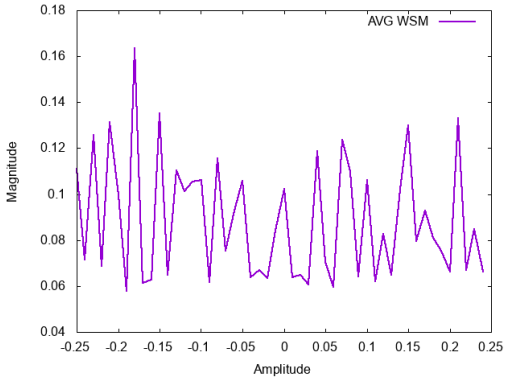
(b) Max Speed



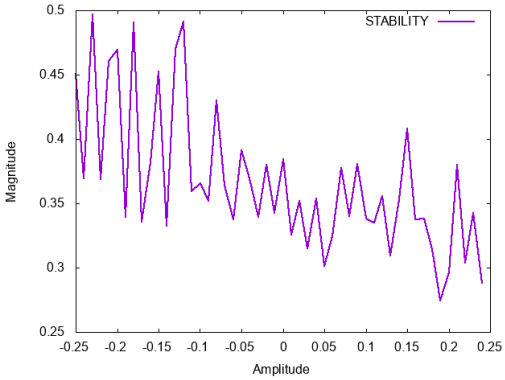
(c) Distance



(d) Average Acceleration

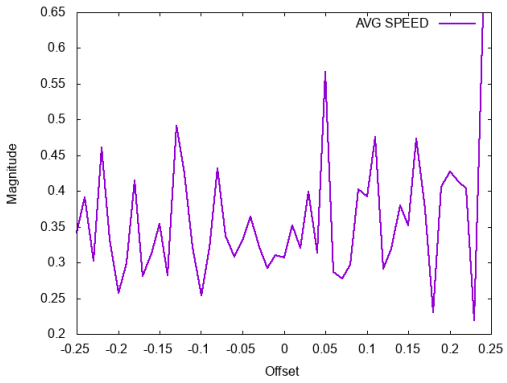


(e) Wide Stability Margin

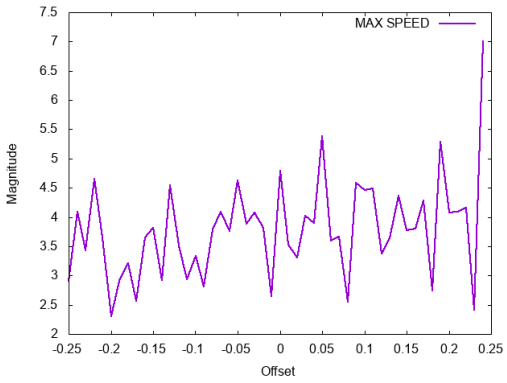


(f) Stability

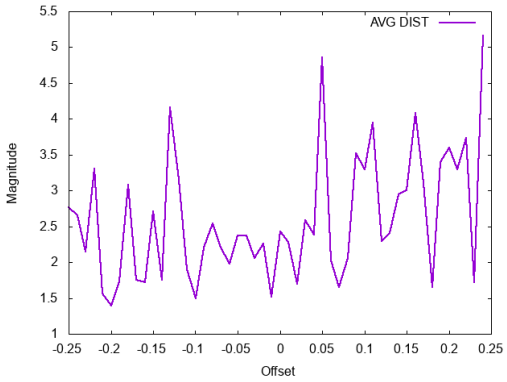
Figure 94: The effect of an amplitude sweep on a trot gaits knee joints with friction



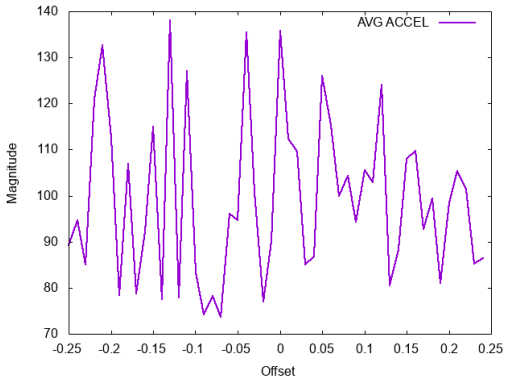
(a) Average Speed



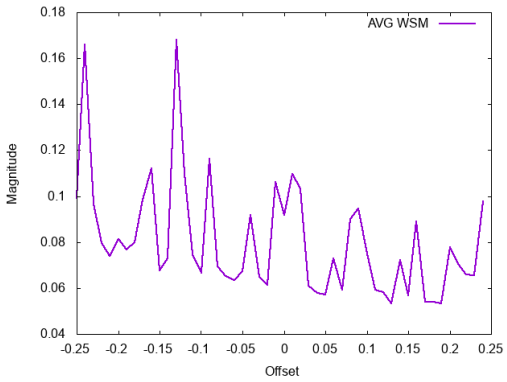
(b) Max Speed



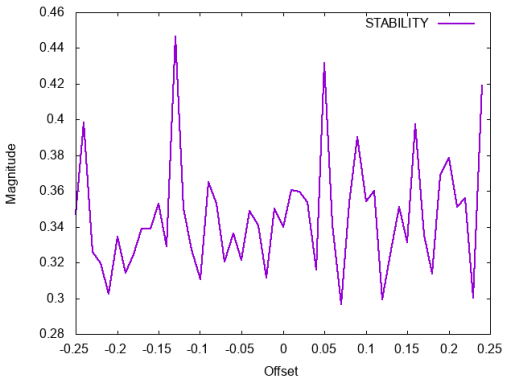
(c) Distance



(d) Average Acceleration

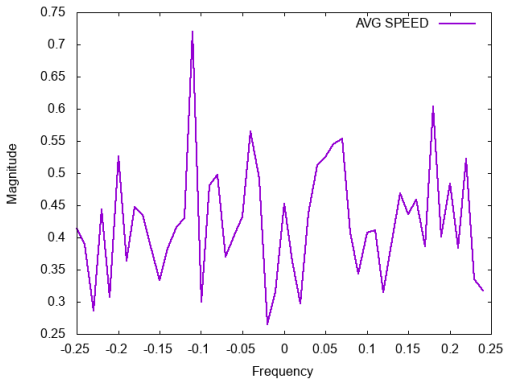


(e) Wide Stability Margin

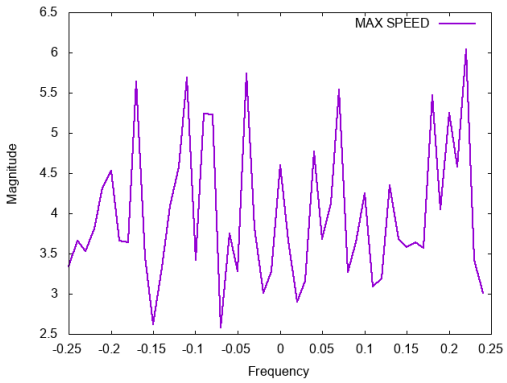


(f) Stability

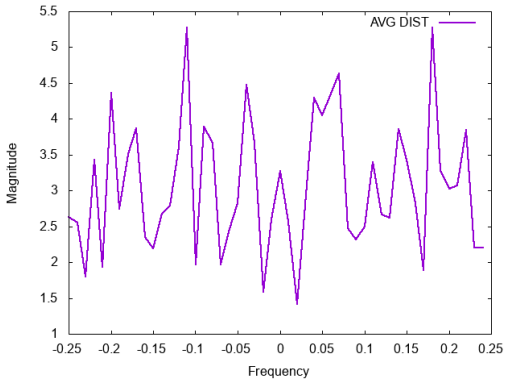
Figure 95: The effect of an offset sweep on a trot gaits knee joints with friction



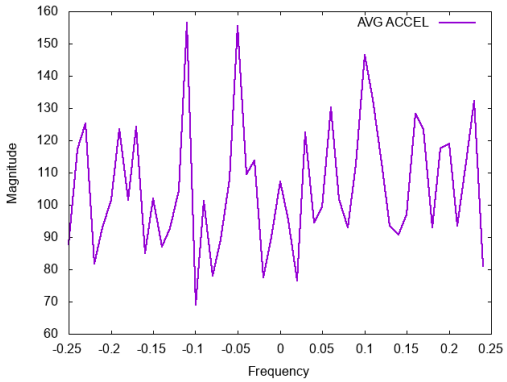
(a) Average Speed



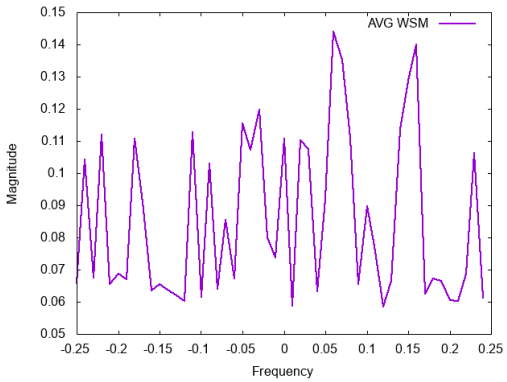
(b) Max Speed



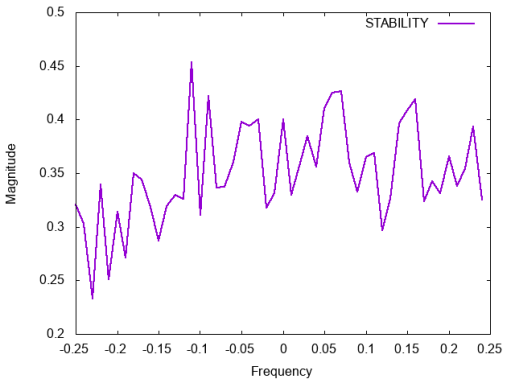
(c) Distance



(d) Average Acceleration



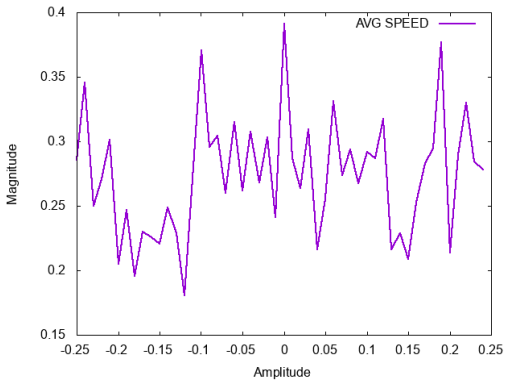
(e) Wide Stability Margin



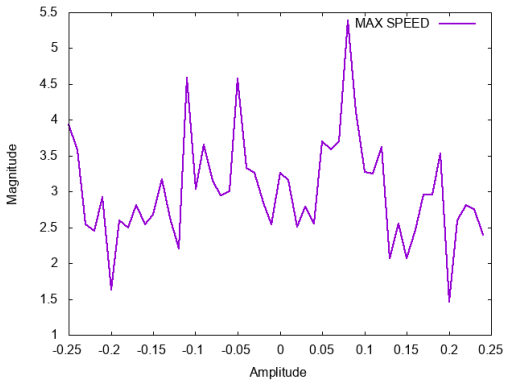
(f) Stability

Figure 96: The effect of a frequency sweep on a trot gaits knee joints with friction

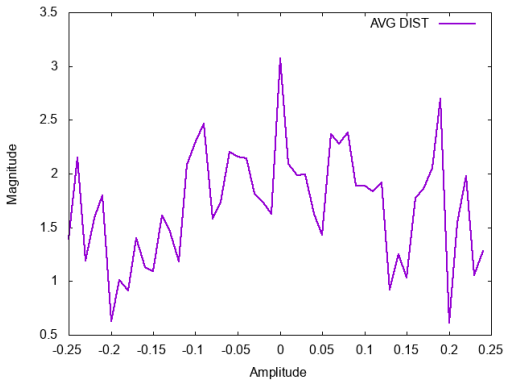
D.3.3.3 Shoulder



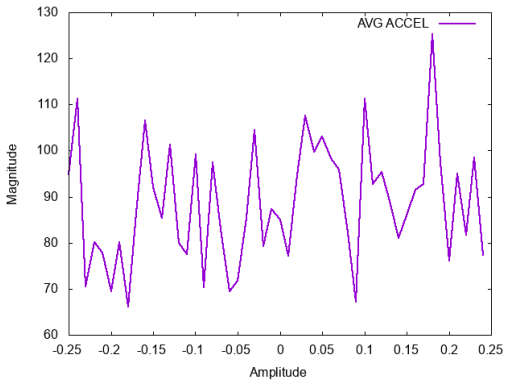
(a) Average Speed



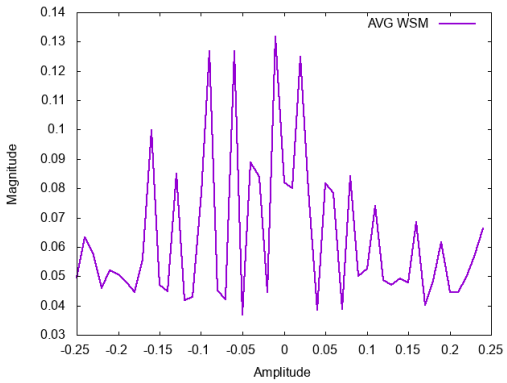
(b) Max Speed



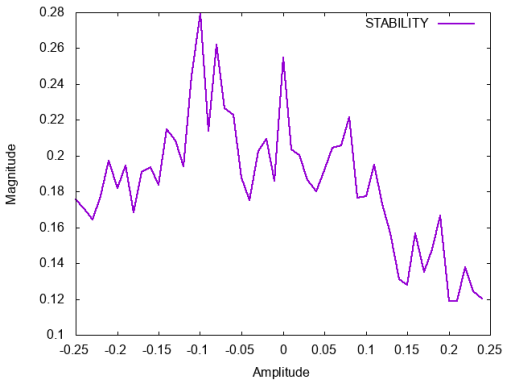
(c) Distance



(d) Average Acceleration

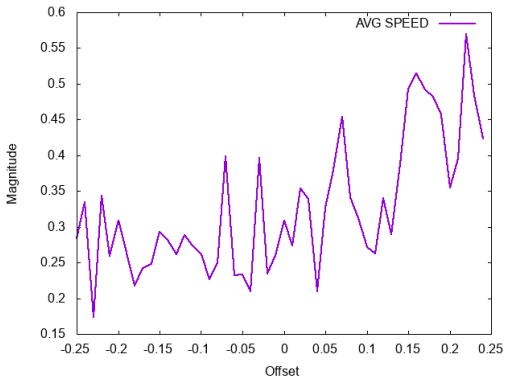


(e) Wide Stability Margin

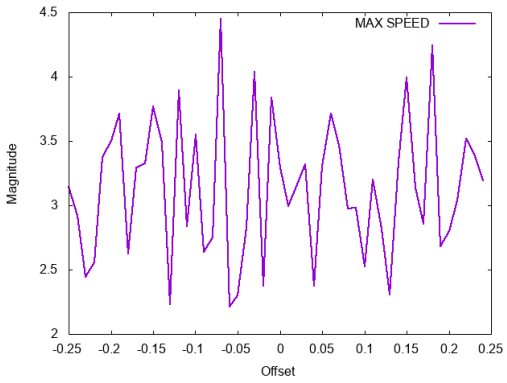


(f) Stability

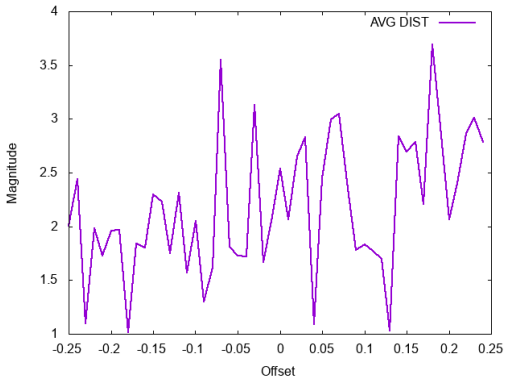
Figure 97: The effect of an amplitude sweep on a walk gaits shoulder joints with friction



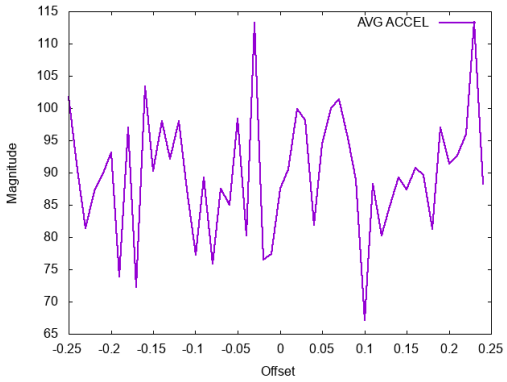
(a) Average Speed



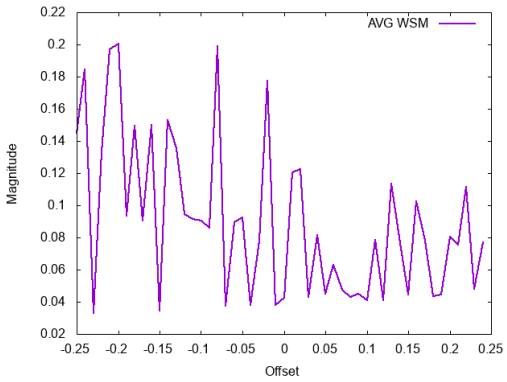
(b) Max Speed



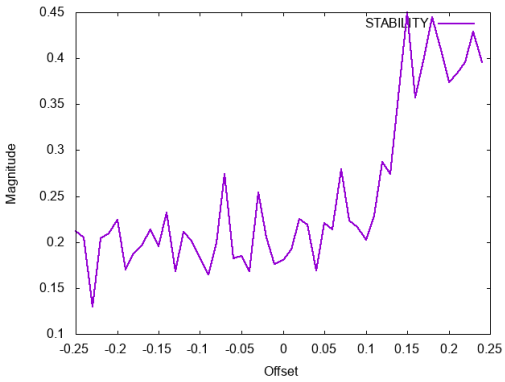
(c) Distance



(d) Average Acceleration

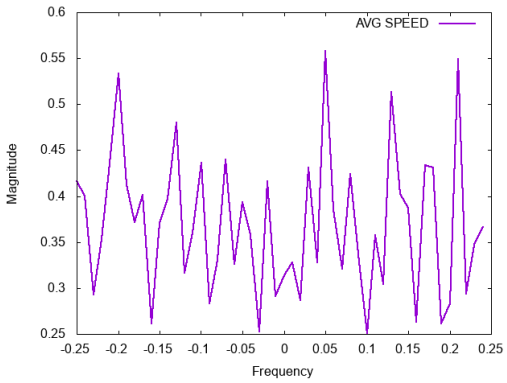


(e) Wide Stability Margin

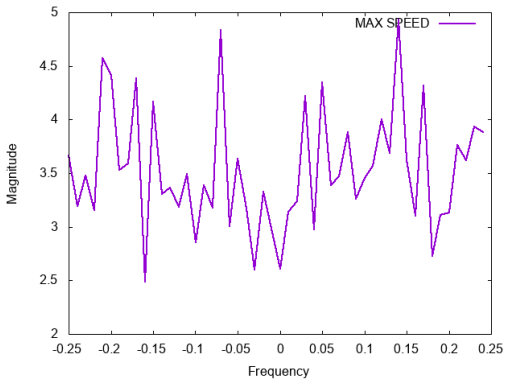


(f) Stability

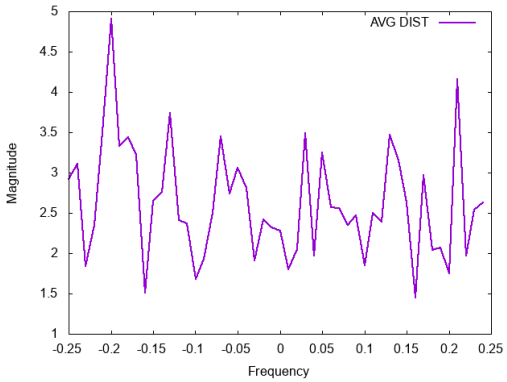
Figure 98: The effect of an offset sweep on a walk gaits shoulder joints with friction



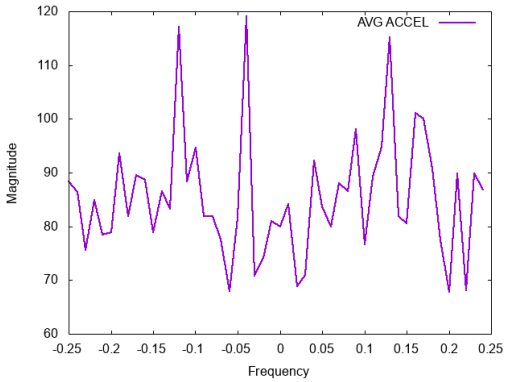
(a) Average Speed



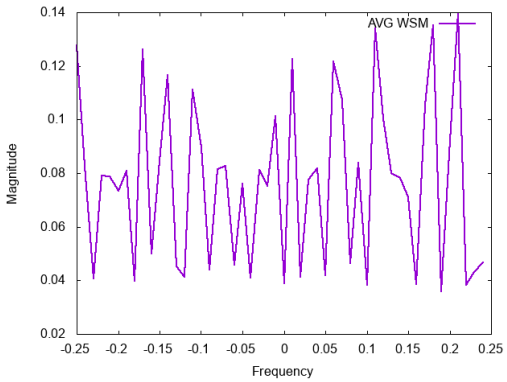
(b) Max Speed



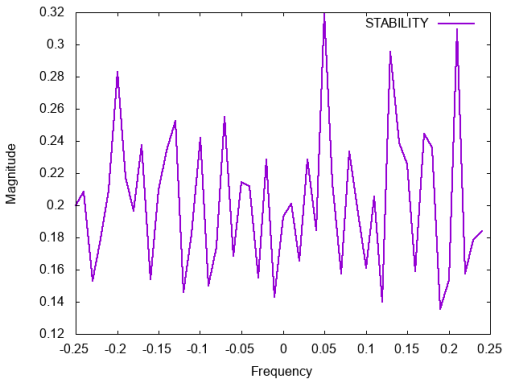
(c) Distance



(d) Average Acceleration

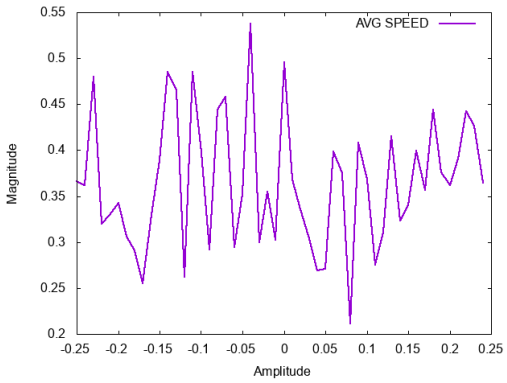


(e) Wide Stability Margin

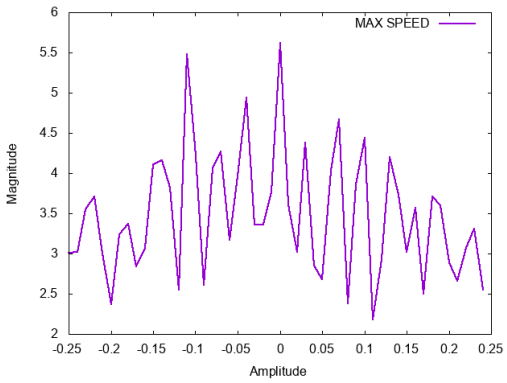


(f) Stability

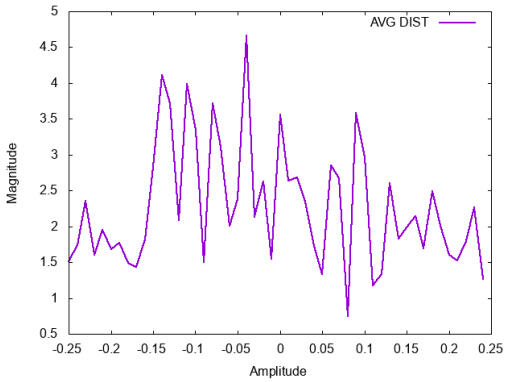
Figure 99: The effect of a frequency sweep on a walk gaits shoulder joints with friction



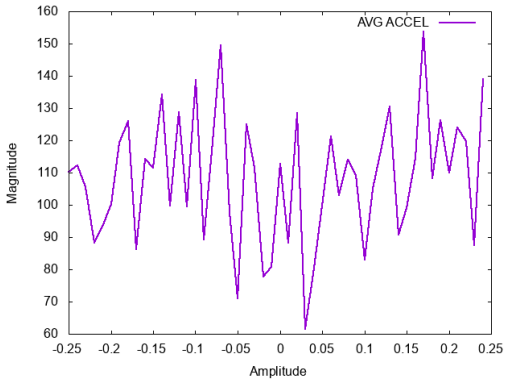
(a) Average Speed



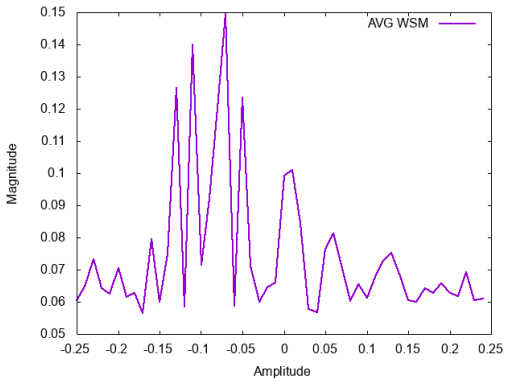
(b) Max Speed



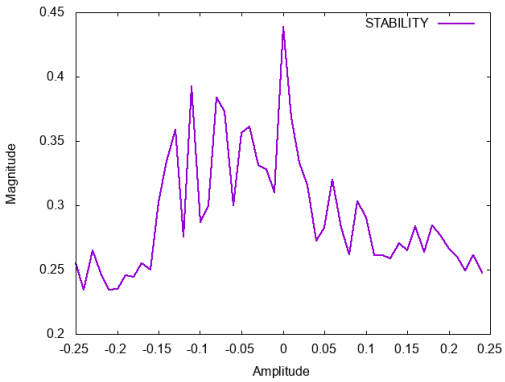
(c) Distance



(d) Average Acceleration

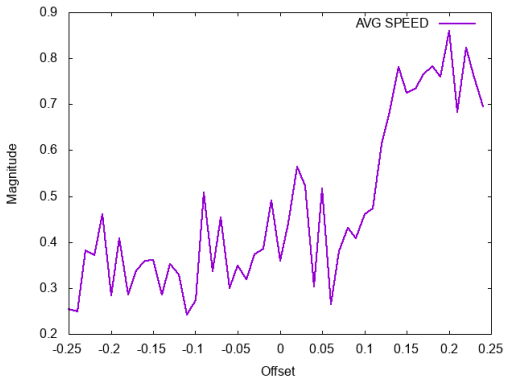


(e) Wide Stability Margin

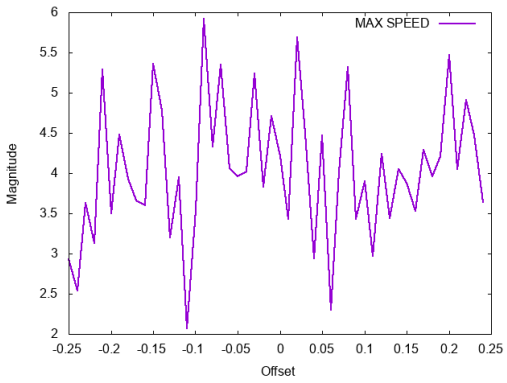


(f) Stability

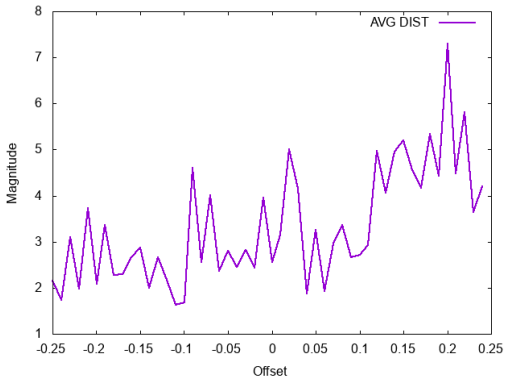
Figure 100: The effect of an amplitude sweep on a trot gaits shoulder joints with friction



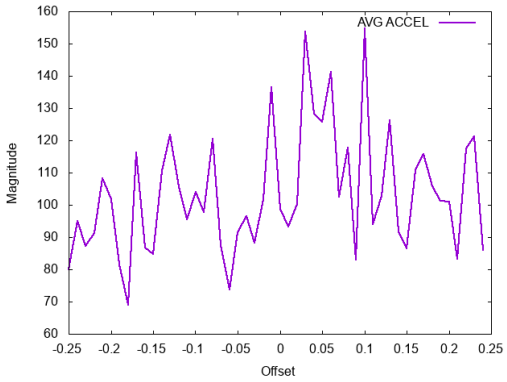
(a) Average Speed



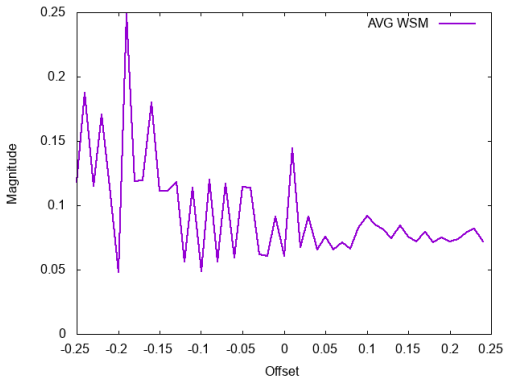
(b) Max Speed



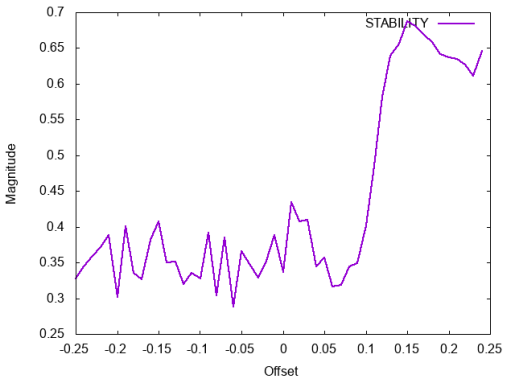
(c) Distance



(d) Average Acceleration

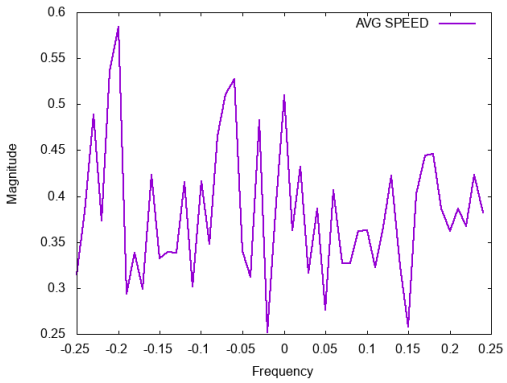


(e) Wide Stability Margin

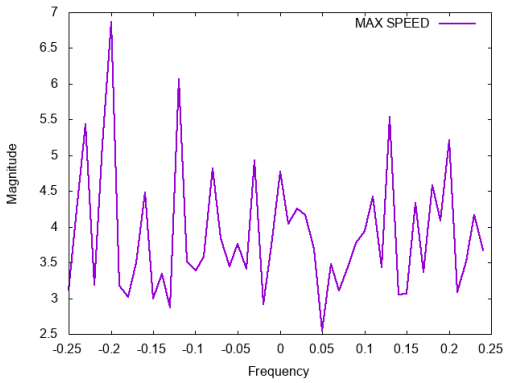


(f) Stability

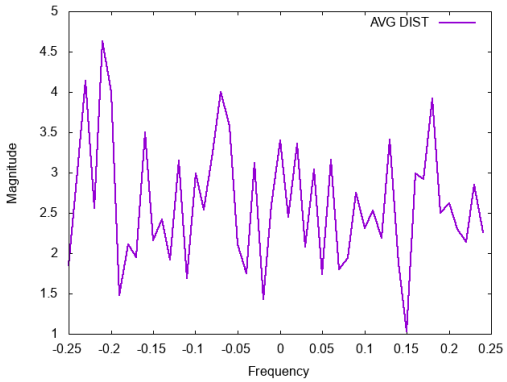
Figure 101: The effect of an offset sweep on a trot gaits shoulder joints with friction



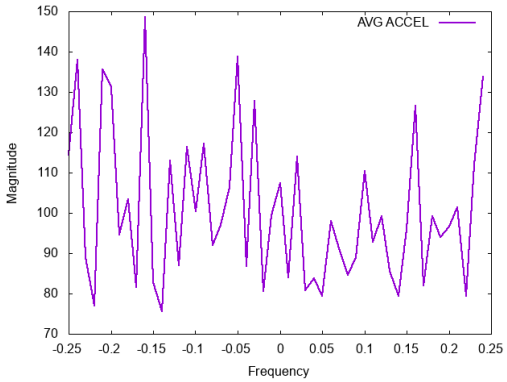
(a) Average Speed



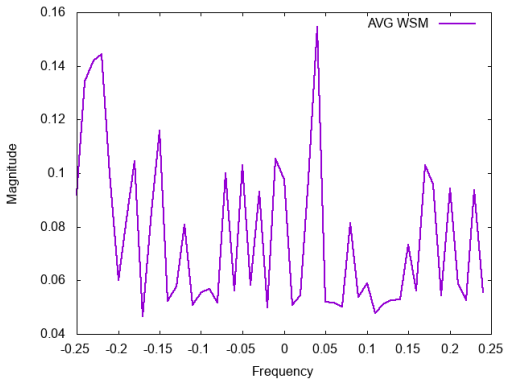
(b) Max Speed



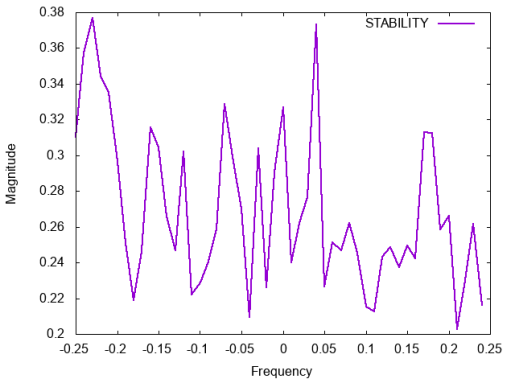
(c) Distance



(d) Average Acceleration



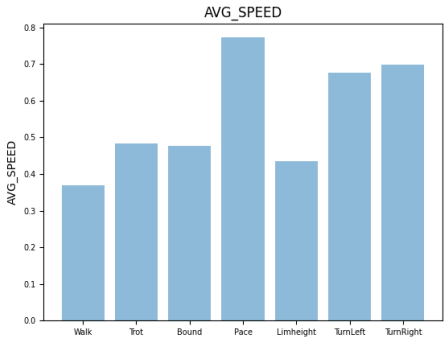
(e) Wide Stability Margin



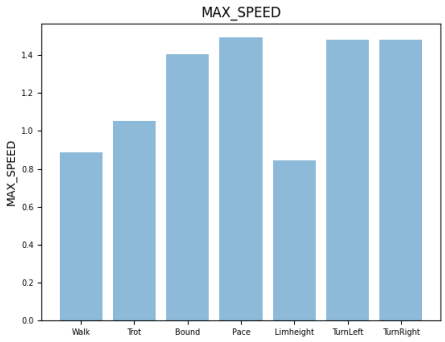
(f) Stability

Figure 102: The effect of a frequency sweep on a trot gaits shoulder joints with friction

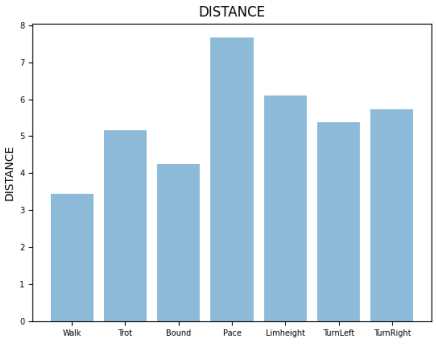
D.4 Friction Gait Comparison



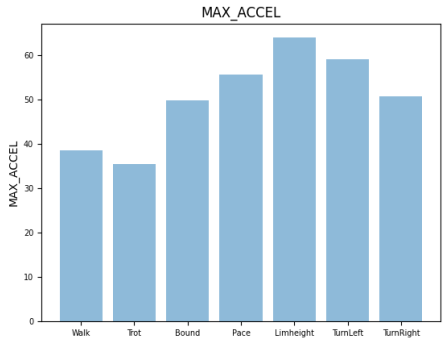
(a) Average Speed



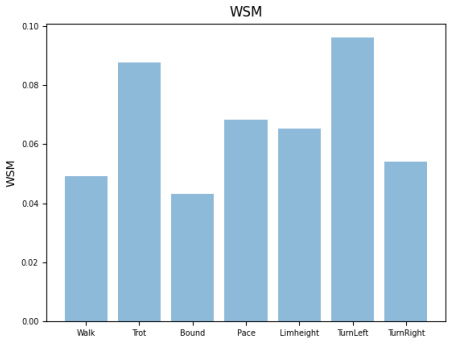
(b) Max Speed



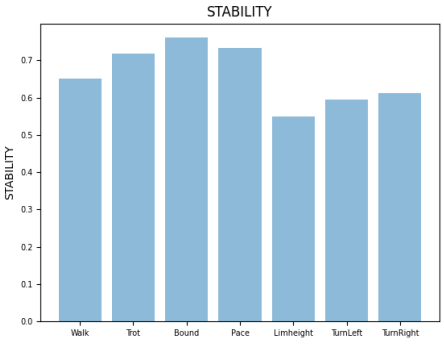
(c) Distance



(d) Max Acceleration



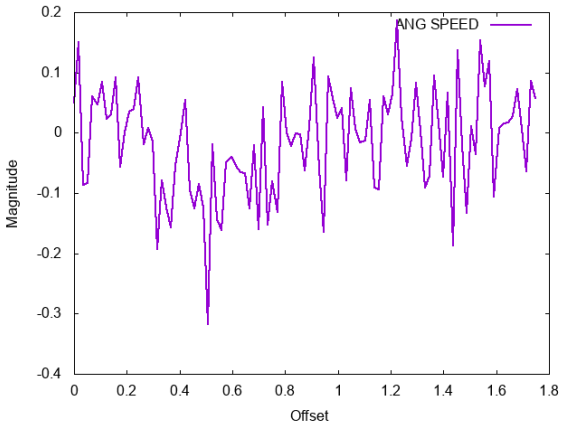
(e) Wide Stability Margin



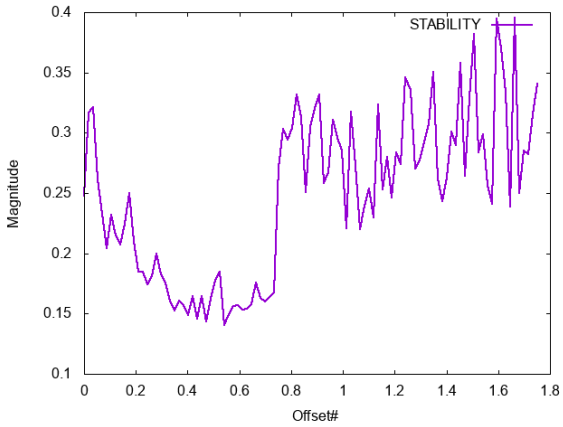
(f) Stability

Figure 103: Bar charts depicting the different gait performances with friction

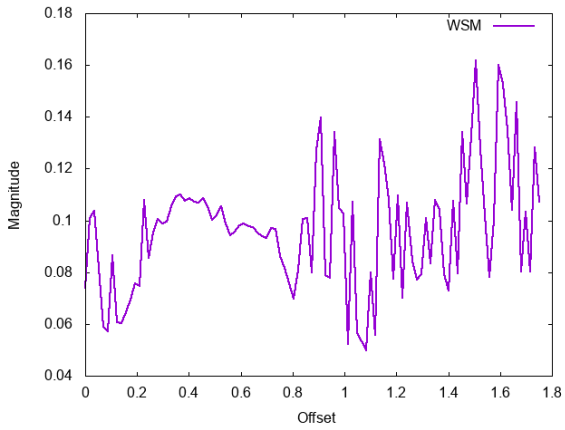
D.5 Turning Speed Experiment



(a) Angular Velocity

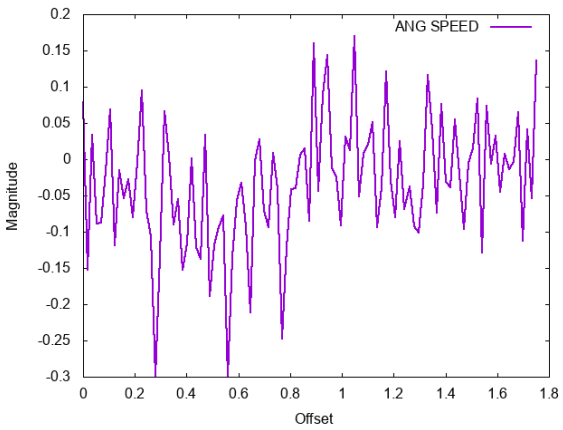


(b) Stability

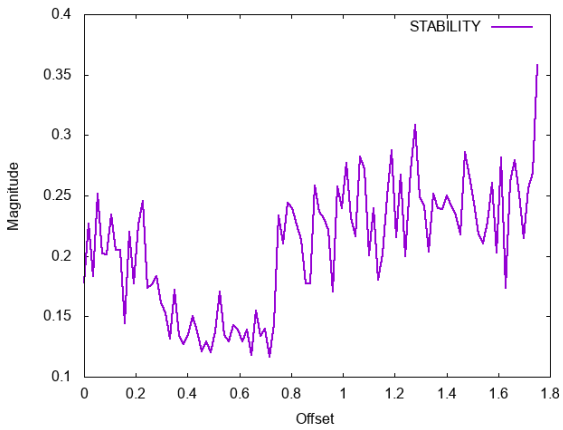


(c) Wide Stability Margin

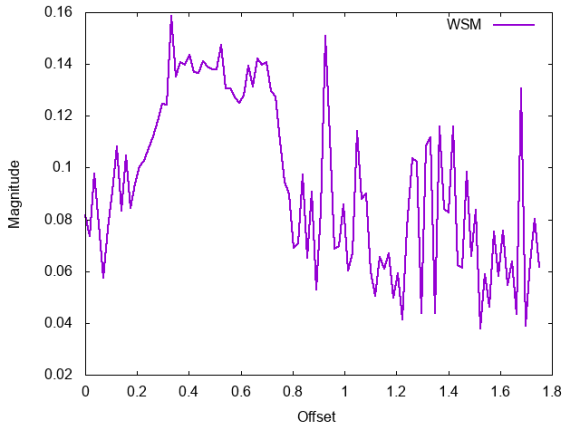
Figure 104: Turn Speed Experiment Results for trot with friction



(a) Angular Velocity



(b) Stability



(c) Wide Stability Margin

Figure 105: Turn Speed Experiment Results for walk with friction