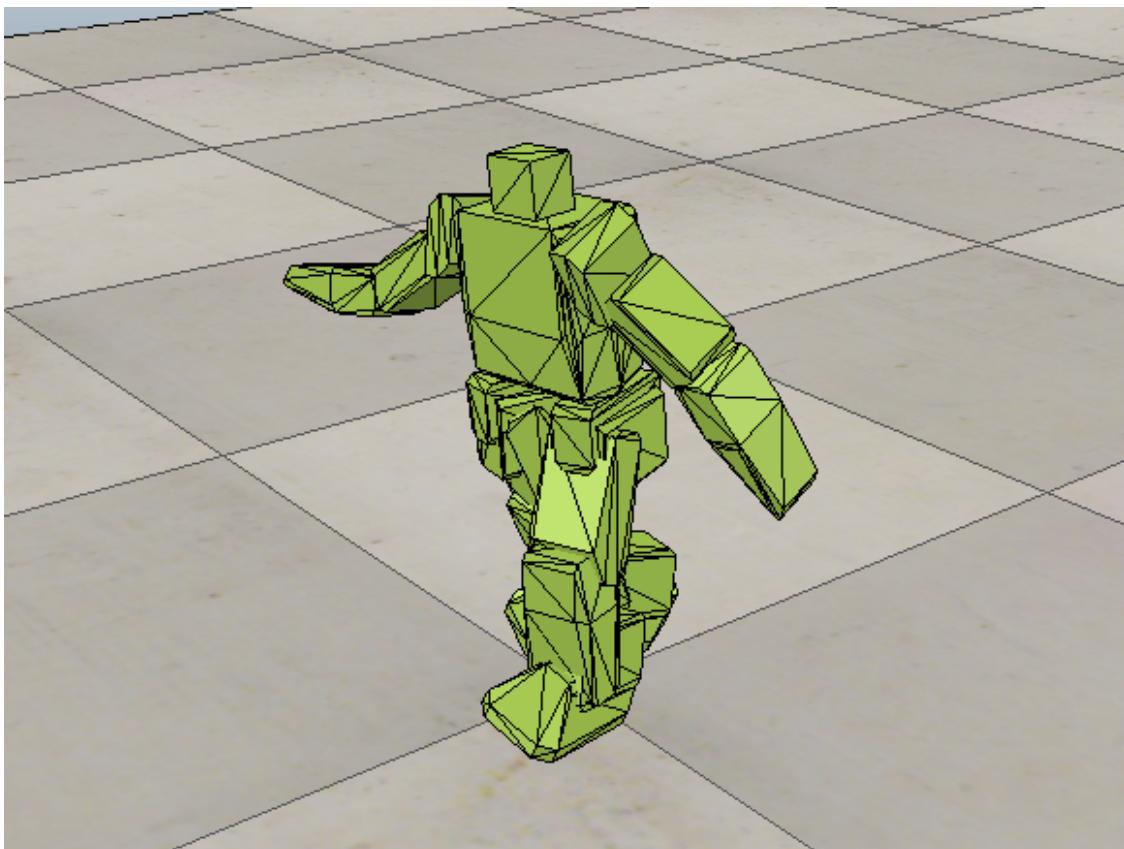




CHALMERS
UNIVERSITY OF TECHNOLOGY



EVOLVING A HUMANOID BEHAVIOUR IN A CPG ROBOT

Humanoid Robotics TIF160

NILS CARLSSON
JOHAN LARSSON HÖRKÉN
FREDDIE OGEMARK

Abstract

A walking cycle for a humanoid robot can be generated through means of reinforced learning algorithms, such as a genetic algorithm. Using data captured from human movements could potentially lead to a faster convergence of the algortihm along with more stable results. A simulation was set up consisting of a CPG network connected to a genetic algorithm, along with a 3D simulation of a humanoid robot. Accelerometer data from a human walking cycle was inserted as a starting point for the algorithm in order to see if the speed and accuracy of the simulation was improved. The results from the simulation was connected to a physical humanoid robot to evaluate the generated movement pattern. Although the results were found to be unstable, the method does show potential for improvement and refinement.

Keywords: CPG, Bioloid, Genetic Algorithm, V-REP, Walk Cycle, Humanoid Robot, Reinforced Learning.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Purpose | 1 |
| 2 | Theory | 2 |
| 2.1 | CPG Model | 2 |
| 2.2 | Optimisation Using Genetic Algorithm | 5 |
| 3 | Methods | 6 |
| 3.1 | Record Physical Model | 6 |
| 3.2 | Pre-Evolution | 7 |
| 3.3 | Implementation of the GA and Simulation | 8 |
| 4 | Results | 11 |
| 4.1 | Processed data | 11 |
| 4.2 | Pre-Evolution | 11 |
| 4.3 | Simulation | 13 |
| 5 | Discussion | 14 |
| 6 | Conclusion | 16 |
| 7 | Further Work | 17 |
| A | V-REP Models | 20 |
| B | Parameters for Walking Cycle GA | 20 |
| C | Processed Data | 20 |
| D | Results of Pre-Evolution | 21 |

1 Introduction

The field of humanoid robotics is a fast growing field of research and development wherein humanoid robots are becoming more and more advanced. There are different ways for a humanoid robot to interact with the world. Walking is one of the rudimentary characteristics of human movement, achieving a bipedal walking gait allows a robot to perform more complex tasks in a world adapted for humans, but constructing a robust walking gait is not trivial. There has been a lot of research and development using Central Pattern Generators (CPGs) to construct a walking gait using Genetic Algorithms (GA) as reinforced learning method. These algorithms are known to take very long to evolve a robust walking cycle. In most projects the GA starts with a random seed, which is one of the reasons for the long time needed to evolve a good walking cycle.

1.1 Purpose

The aim of this project is to investigate if using data gathered from a human walking cycle as the seed in the GA of a CPG will make the GA evolve faster and give a better walking cycle compared to using a random seed and letting it evolve from scratch. The robot used as the target platform for the movement cycle is the Bioloid robot [12], shown in Figure 1.



Figure 1: The Bioloid robot used throughout this project.

2 Theory

In the following the section the underlying theory of the models used in the project are introduced and explained, along with relevant background and examples.

2.1 CPG Model

2.1.1 Mathematical Model of Physical Behaviour

Biological behaviour in mammals, such as walking or swimming, are usually repetitive and oscillatory creating energy efficient movements. This can be seen in Figure 2, which shows the walking cycle of a human, measured as the acceleration in Z-direction whilst walking. Due to the rhythmic patterns in the behaviour, it is possible to construct a mathematical model to mimic this type of behaviour.

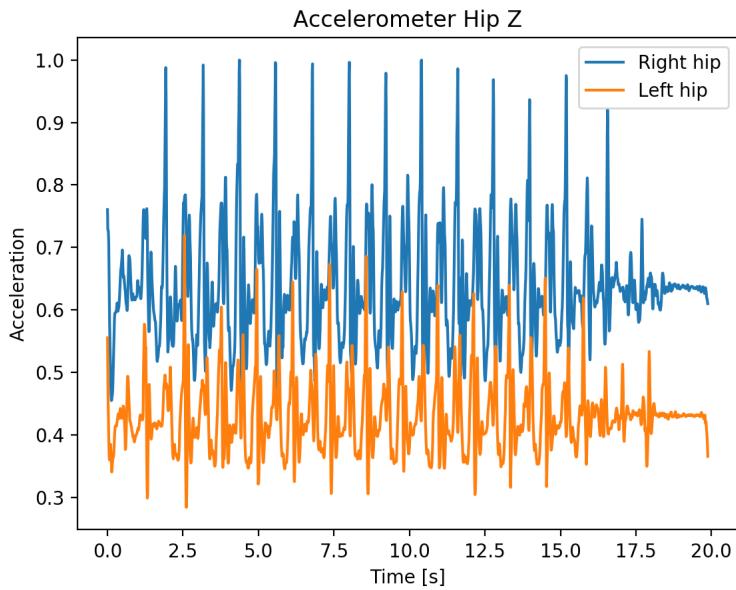


Figure 2: Walking cycle of a human. Where Z is the horizontal direction of the reference system.

A Central Pattern Generator (CPG) is a mathematical model of a neural oscillator, a neural network that produces rhythmic patterned outputs without sensory feedback [1]. The output of a CPG can be used as a model of biological rhythmic behaviour, for instance the activation of a certain muscle.

The activity of one muscle in an animal affect the activation of other muscles in its body automatically in order to distribute load and to achieve balance. Similarly CPGs can be modelled as a network, to account for more than a single muscle as well as the interaction between muscles.

2.1.2 Model of Neural Oscillator

There exists many different models of CPGs [3], the model to be examined throughout this project is the half-centre model. The half-centre model proposes to account

for the alternate activation of the extensor and flexor muscles in the limbs, originally inspired by a cat walking. A mathematical model was proposed by K. Matsuoka [4], using two mutually inhibitory neurons in each oscillator to account for the extensor–flexor behaviour. The model ensures that when one neuron is active, the other is suppressed, creating the oscillatory behaviour.

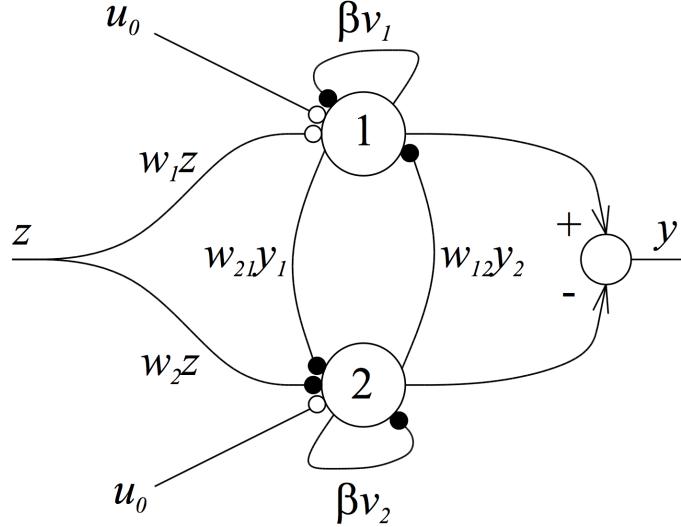


Figure 3: The Matsuoka model of a half-centre CPG [9], with parameters explained in Table 1.

The setup of the Matsuoka model is shown in Figure 3, with the parameters as described in Table 1. The behaviour of the model is described by the following equations [5]:

$$\tau_u \dot{u}_i = -u_i - \beta v_i + \sum_{j=1}^n w_{ij} y_j + u_0, \quad (1)$$

$$\tau_v \dot{v}_i = -v_i + y_i, \quad (2)$$

$$y_i = \max(0, u_i), \quad (3)$$

$$o = y_2 - y_1 \quad (4)$$

where the model is simulated by integrating over time using Euler integration and timestep Δt and \dot{u}_i , \dot{v}_i from above in:

$$u_i(t + \Delta t) = u_i(t) + \Delta t \dot{u}_i, \quad (5)$$

$$v_i(t + \Delta t) = v_i(t) + \Delta t \dot{v}_i. \quad (6)$$

2.1.3 Model of CPG Network

Coupling multiple CPGs into a network makes it possible to model more complex structures, such as an animal or a human. This can be done by connecting the

| Parameter | Description | Initial value | Comment |
|-----------|--|---------------|-------------|
| z_i | input of neuron i | | oscillating |
| u_0 | external tonic input | 1.0 | |
| u_1 | inner state of neuron 1 | 0.0 | fixed |
| u_2 | inner state of neuron 2 | 1.0 | fixed |
| v_1 | auxiliary variable measuring the degree of self-inhibition of neuron 1 | 1.0 | fixed |
| v_2 | auxiliary variable measuring the degree of self-inhibition of neuron 2 | 0.0 | fixed |
| β | modulation parameter | 2.5 | |
| w_{ij} | weights connecting neuron j to neuron i | -2.0 | |
| y_i | the output of neuron i | 0 | |
| τ_u | time constant | 0.025 | |
| τ_v | time constant | 0.3 | |
| o | the output of the oscillator | | |

Table 1: The parameters used in the Matsuoka model of a half-centre CPG.

output of one neuron to the input of another. This project investigate the walking behaviour of a human implemented on a humanoid robot, thus human behaviour can be modelled by looking at the movement in different joints of the body, corresponding to the joints of the humanoid robot. Since the output of a CPG is one-dimensional over time, and human movement is three dimensional, one can imagine modelling the movement in each dimension with a separate CPG. The output signals can then be superimposed to obtain the total movement.

2.1.4 Choice of Input to Oscillator

Each of the neurons in a CPG network have inputs (z_i) that are affected by the output of the other neurons (o). One can imagine that all neurons are fully connected, but by setting the weight between two neurons to zero, that weight is *removed*. The input to the neurons can be calculated in different ways, one way inspired by Liu et.al. [7] is to define the inputs to neuron i as

$$z_1 = \sum_{j \text{ neighbour of } i} a_{ij} u_j, \quad (7)$$

$$z_2 = 0. \quad (8)$$

Another implementation inspired by Shan et.al. [8] defines the inputs as

$$z_1 = \sum_{j \text{ neighbour of } i} a_{ij} [y_j]^+, \quad (9)$$

$$z_2 = \sum_{j \text{ neighbour of } i} a_{ij} [y_j]^-, \quad (10)$$

where

$$[y_j]^+ := \max(y_j, 0), \quad (11)$$

$$[y_j]^- := \min(y_j, 0). \quad (12)$$

Neurons can also be affected by an external tonic input (u_0). The amplitude of the oscillation in the CPG is proportional to the tonic input. If the input is oscillatory, the oscillator will lock to the frequency of the input. When the input is removed, the oscillator will smoothly return to the original frequency [9].

2.2 Optimisation Using Genetic Algorithm

A CPG alone might not achieve satisfactory results, as the parameters in the CPG or network might need to be tweaked and optimised to achieve the desired behaviour. A commonly used technique for optimising behaviour is to use reinforcement learning to improve the model based on an external measure of success. This project uses a Genetic algorithm (GA) as a reinforcement learning technique to achieve desired walking behaviour. A genetic algorithm is an evolutionary algorithm that takes inspiration from natural selection implemented as an optimisation problem [2].

2.2.1 Genome

The *genome* is a set of parameters that defines a proposed solution to the problem that the GA is trying to solve. One can imagine various different genomes when optimising the behaviour of a CPG, for instance the internal parameters in the CPG, shown in Table 1. The connection weights between neurons in a CPG network, as described in Section 2.1.3, could also be used to account relation between CPGs in a network.

2.2.2 Operators

The basic implementation of a GA is in summary; initialise the population, simulate the behaviour, evaluate success, selection process, mutation and permutations, repeat until converge. A more detailed description can be found in [6]. The evaluation process, also known as the *fitness function*, produces a fitness score which is the measurement of the success of the individual. Since this project investigates walking behaviour in a humanoid robot, a fitness function may include e.g.; distance walked, time walked, stability, falling, walking style, etc.

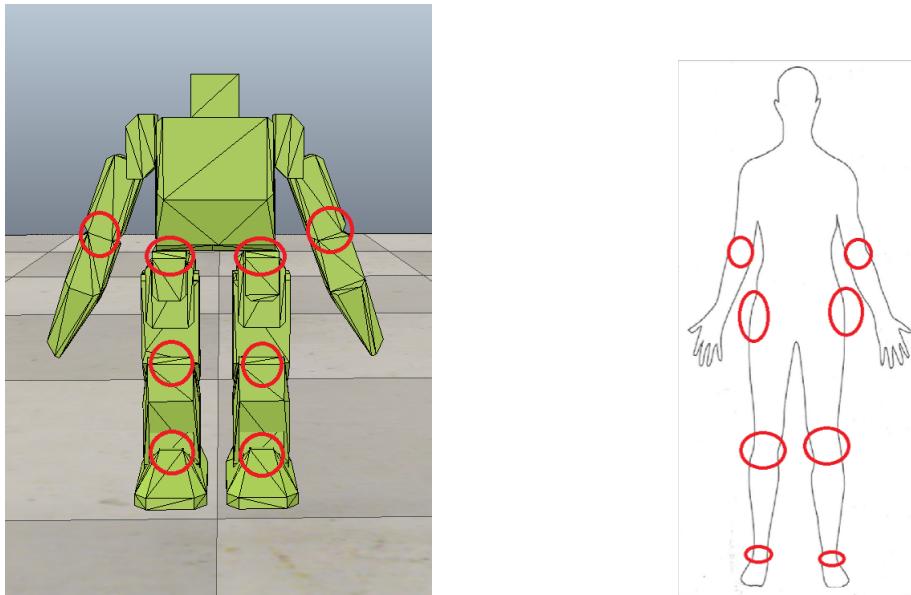
3 Methods

This section introduces the methods used throughout this project, along with a description of the tools and setup of the simulations. The approach is to collect data from human movement to use in a pre-evolution. Then use the pre-evolved model when performing simulations on a Bioloid robot model, and later in the physical robot. Much of the basic simulation environment was provided by a previous research project [11], which was further developed during this project.

3.1 Record Physical Model

3.1.1 Data Collection

The Bioloid robot has 18 Degrees of Freedom (DoF), which is divided on 8 different joints. These joints can be seen in Figure 4a. To gather data from a human during a walking cycle, accelerometers were strapped to the human in the joints corresponding to the joints of the Bioloid robot, as seen in Figure 4b. Smartphones with the app *Sensor Fusion* [13] was used to collect the accelerometer data. Using mobile phones enabled easy access to many high quality sensors, and the app provided an interface to export raw sensor data directly to CSV format. The phones were strapped to the joints shown in Figure 4b. A walking cycle was initiated with a jump, to find a clear correlation in initialisation between devices, making it easy to extract and align the signals. More than one walking cycle was recorded during the same session separated by the initialisation signal.



(a) Bioloid with joint locations indicated by red circles. (b) Human body with accelerometer locations indicated with red circles.

Figure 4: Joints examined during data collection

3.1.2 Data Processing

The recorded data from the walking cycles were processed to align the data between devices and extract walking cycles. All signals were normalised to account for different sensitivities between sensor models, thus removing bias.

3.2 Pre-Evolution

3.2.1 Evolve to Fit Physical Model

To leverage the behaviour of humans using the collected data presented in Section 3.1, a pre-evolution was constructed. The main goal of the pre-evolution was to adopt the output of a CPG network to fit the behaviour of the recorded data. Performing the pre-evolution in an environment not requiring any graphical interface or control of a robot model would hypothetically converge faster than using a full-scale simulation environment. Using an optimised model when performing later simulations on a robot model would then ideally evolve a better behaviour faster.

3.2.2 Model Construction

The initial intention was to use the same model in the pre-evolution as in the evolution in the robot simulation. But it was discovered that the existing simulation and evolution environment were too highly linked together, meaning that it was easier to construct an independent pre-evolution model. The model was highly inspired by the existing environment, using the Matsuoka model described in 2.1.2 to model the CPGs and initial parameters as presented in Table 1. The pre-simulation investigated two different types of genomes. In the first case only the connections between the neurons in the CPG network was used, while in the second case all the parameters of the Matsuoka model was used.

The pre-simulation was performed on a CPG network of 18 neurons, corresponding to the 18 DoFs on the humanoid robot. The neurons were initialised fully connected using a connection matrix with random values. The input model by Liu introduced in Section 2.1.4 was used, the outputs were initiated randomly, since it was required that the output values were initialised to non-zero values to start the evolution. The CPG model was simulated over 20 seconds with a timestep of 0.01 seconds, corresponding to the time of one recorded walking cycle, generating an output signal. The population size was varied between simulations between 20-40 individuals, and full-scale evolutions were made with 500-1000 generations.

3.2.3 Fitness and Reinforcement

When simulated, the CPGs were evaluated by assigning each neuron to a DoF, then comparing the output signal to the corresponding recorded reference signal. This was done by taking the *correlation* between the output signal and the reference signal. A correlation of 1 mean full correlation (identic signals) and a correlation of -1 indicates no correlation (signals shifted half a period). The fitness score was acquired by taking the absolute value of the *lowest* correlation between an output DoF and a reference DoF in an individual, the idea being to maximise the worst

correlation in each population. Other fitness functions can be imagined as measuring different similarities between the signals, analysing the moments of the correlation distribution, analysing symmetry between signals, evaluating each DoF individually instead of depending on the rest of the individual, etc.

Once evaluated, the next generation was chosen by tournament selection, with a tournament size of a fifth of the population size. Elitism was implemented not to lose the best individual, mutations and permutations were randomly applied to the rest of the population.

3.3 Implementation of the GA and Simulation

The method used to evaluate and iterate the weights for the CPG was a standard genetic algorithm, with elitism, tournament selection, mutations and crossovers. The specific parameters of the model can be seen in Appendix B. The fitness of each individual was evaluated by measuring the distance a simulated 3D model of the Bioloid could travel. The simulated Bioloid accepted input signals in the same way as the physical one. This made it possible to accurately measure and analyse the movement of the robot without having to connect it to the actual Bioloid, thereby speeding up the simulation and the algorithm. However, it came at the cost of discrepancies between the simulated environment and the real world. The chosen simulation environment in this project was V-REP [14], in which a model of a Bioloid robot was imported. Based on how far the simulated Bioloid was able to walk, each individual was assigned a fitness value and compared to the rest of the evaluated individuals. Several fitness functions were evaluated, and the one found to be the most useful was one that multiplied the greatest distance the robot travelled, with the lowest height of the robot. This was to prevent the robot from falling over while still rewarding behaviour that pushed it forward. However, based on the specifics of the simulation, such as stability and DoF, it could be varied for a more specialised approach, such as simply the distance travelled forward.

3.3.1 Configuration of V-REP

V-REP efficiently allowed for the import and simulation of the Bioloid robot. The 3D model of the Bioloid was the same as used in the previous project [11], and can be viewed in the V-REP environment in Figure 5. After loading the model into a specific V-REP scene, the GA automatically sent signals into the simulation as soon as it started. The V-REP platform also allowed for an arbitrary number of joints/servos to be simulated, meaning the algorithm could be focused to generate patterns only for a select number of servos.

3.3.2 Physical Implementation and Hardware

The physical Bioloid robot was pre-assembled into the standard configuration A [10], offering the highest degree of freedom and replication of the human body. The standard control board on the Bioloid was ignored in favor of connecting directly to the servos and controlling them through an application on the computer. This was accomplished by daisy-chaining all the servos together and connecting them

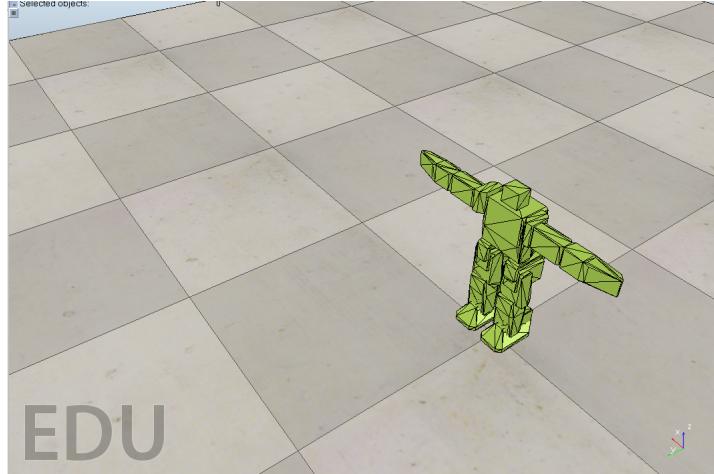


Figure 5: The 3D Bioloid model used in the V-REP simulation environment.

to both a power source and a *USB2Dynamixel* [15] dongle, which allows for direct computer-to-servo communication. As before, much of the software infrastructure regarding communication was taken from the previous student project [11], along with some specific software to communicate with the USB2Dynamixel dongle. As the simulated Bioloid in V-REP had higher friction against its surroundings, the physical Bioloid was modified in order to increase friction of its feet against the floor. This was accomplished with simple modelling clay attached to the soles of its feet.

3.3.3 Simulation Process

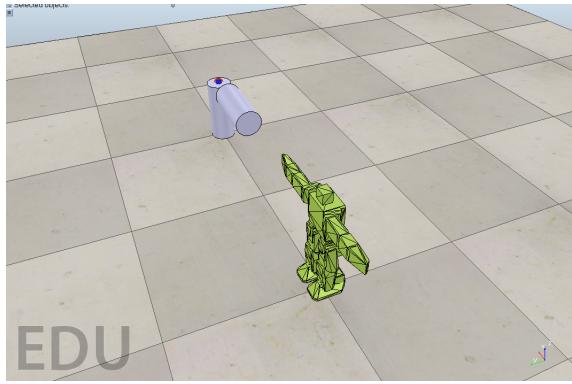
As previously mentioned, the number of simulated servos could easily be changed in the simulation. Two main approaches were used - one where all 18 joints of the Bioloid were simulated, and one where only 8 were considered. Based on the initial hypothesis, a robot with 18 DoF would be better able to replicate human movement and should thus be able to generate the best results. However, the increased degrees of freedom also led to an increased tendency for the robot to topple and fall. Initially, the simulation with all 18 joints simply let the robot orient itself without any external support and attempt to develop a walking gait from there. It quickly became apparent that the full 18 DoF case would be very slow to develop a walking gait, if it did so at all. In order to speed up the process support rods attached to the robot was used, first four, and then two, in order to prevent the robot from falling down but without interfering too much in the evaluation of the fitness. The 4-rod model can be viewed in Figures 6a, and the 2-rod model can be see in Appendix A, Figure 10. The purpose of these was to take the best result from the simulation with four support rods, and insert that into the simulation with two support rods, and then finally insert the best result from there into a simulation with no support, in order to finally reach a stable gait.

In the case of only 8 joints, the robot was constricted to only move in the forward direction by means of a support arm, seen in Figure 6b. The arm was used to prevent the robot from falling to either side. The purpose of this was to increase its stability

3 Methods



(a) The Bioloid model with 4 support arms in the V-REP simulation environment.



(b) The Bioloid model with the external support arm in the V-REP simulation environment.

Figure 6: Overview of the different support structures used in the simulation.

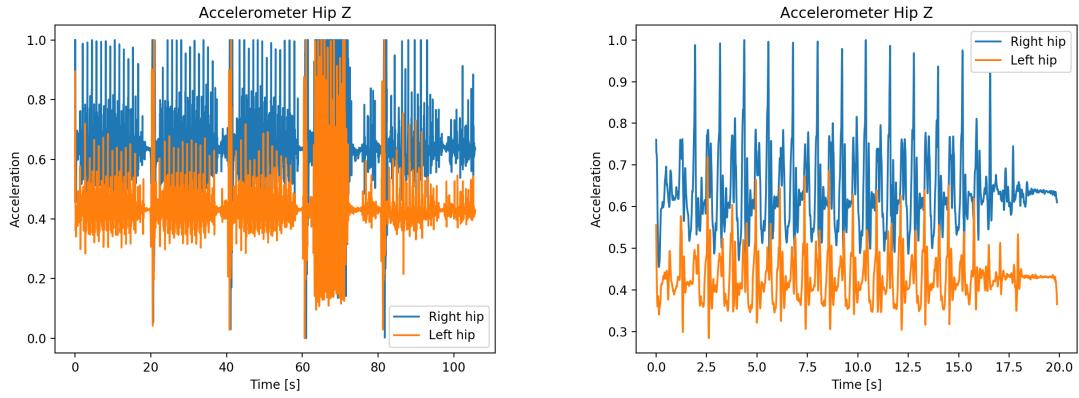
and speed up the simulation, sacrificing freedom of movement and possibly a more human-like gait in order to get stable results faster. Of course, the physical robot would then need to be attached to a support beam as well in this case. Based on preliminary results, the robot was also simulated with all 18 DoF and a support arm, to see if it gave any performance improvements in comparison to the 8 DoF case. Each simulation ran for around 50-70 generations with around 30 individuals in each population. The reasons for these parameters were somewhat arbitrary, as they seemed to give a good enough mix of convergence speed coupled with a reasonable runtime (around 5-7 hours).

4 Results

The results from both the pre-evolution process and the main simulation are presented in this Section. When possible, figures have been used to more clearly represent the data. In the case of the movement of the robot, care has been taken to describe the dynamics as accurately as possible.

4.1 Processed data

The processed data was plotted in a graph to identify walking cycles, see Appendix C. As this plot contained too much information to give a clear image and for clarity, two plots were made containing only the information from the left and right hip's acceleration in the z-axis which was enough to check for a pattern. These plots can be viewed in Figure 7, where walking cycles can be clearly identified.



(a) Plot of the gathered data from five walking cycles. (b) Plot of the gathered data from one walking cycle.

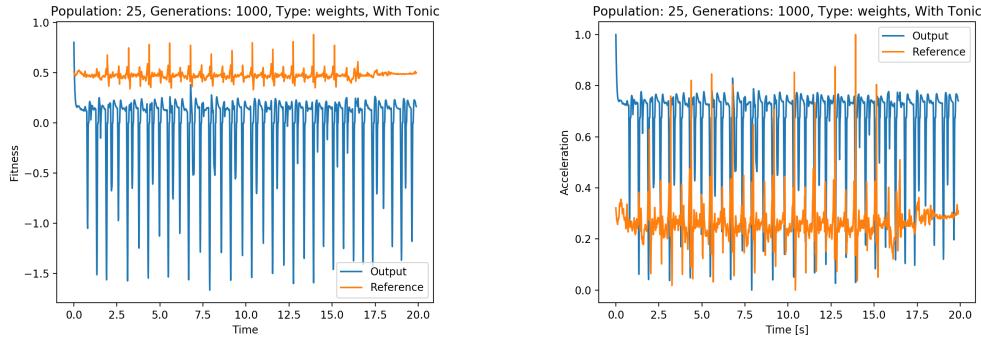
Figure 7: Processed Data from the recorded walking cycles.

4.2 Pre-Evolution

Simulation of the CPG model was done over 20 seconds with a timestep of 0.01 seconds, corresponding to the time of one recorded walking cycle. The population size varied between 20-40 individuals depending on simulations. Full-scale evolutions were made with 500-1000 generations.

As seen in Figure 8a, the amplitude of the output signal does not match the amplitude of the reference signal. To easier compare the signals, the output signals were normalised, which can be seen in Figure 8b. It can be seen that the pre-evolution simulation manage to create rhythmic output patterns, but does not manage to match the amplitude of the reference signals significantly. This is confirmed by the fitness scores, shown in Figure 9. The fitness is maximally around 4% correlation, which is quite low, since they ideally should be fully correlated. When using the entire Bioloid as type, it refers to using both the internal parameters in the CPG as well as the weights between them as genome. Further results are shown in Appendix D.

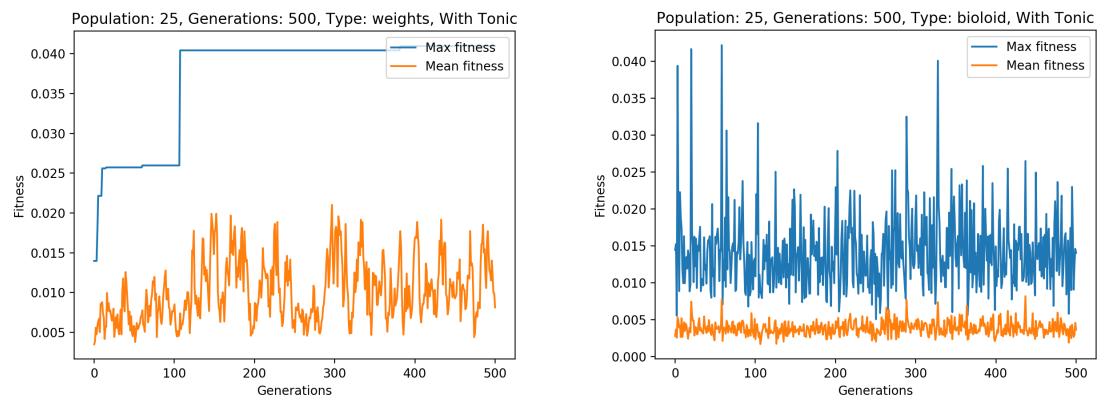
4 Results



(a) Original version.

(b) Normalised version.

Figure 8: Output signals of the CPG network.



(a) Only weights.

(b) Entire bioloid, both weights and internal parameters.

Figure 9: The evolution of the fitness function.

4.3 Simulation

In the case of the 18 DoF simulation with the support rods attached to the Bioloid robot, the Bioloid never achieved a walking cycle. From the start, the simulation with four rods did not properly converge into a stable cycle, which propagated to the case with two support rods, and further in the simulation with no support rods. The initial values were evidently too unstable for the robot to support itself, even if it did display some rudimentary capability to throw itself forward rather than backward. With these results, efforts were focused on the simulations where the external support arm was present, both with 8 and 18 degrees of freedom. Here the results are more closely overlapping, but as expected the 18 DoF simulation is more unstable than the 8 DoF simulation.

Neither simulation achieves a walking cycle comparable to the data gathered from the accelerometers, which means it is quite far off from a humanoid walking cycle. Both movement cycles seem to mainly exploit the fact that in the simulation, it is possible to inch the robot forward by extending one leg and wobbling the foot servo, generating a small amount of friction that slowly pulls the robot forward. However, this does not hold up when applied to the physical Bioloid, since the friction coefficient is completely different, caused by the reality gap. This can probably be tweaked by changing the friction material (modeling clay in this case) to something with less grip, possibly allowing the Bioloid to inch forward like in the simulation.

5 Discussion

When using the entire bioloid as a type during the pre-evolution generated a somewhat unexpected behaviour in the fitness function, as seen in Figure 9a. Since the implementation is using elitism, we would expect the maximum fitness to be non-decreasing. The fitness is measured at the end of a simulated generation, and the correlation is an aggregate of the similarity between the output signal and the reference signal over time. Hence, when using the internal parameters as genome, their performance depend on the earlier input values to the CPG, which are constantly changing between generations as the model is updated. An elite individual can therefore never be guaranteed to be the best individual both in current and in future generations. Compared to using only the weights as genome, where the internal parameters in the GPGs are reset between each simulation run, the individuals always have the same starting conditions for each generation, and the elite individual can be maintained.

Why are these results not better then? To start with, the simulations were only made over maximum 1000 generations with a population size of 30. Even if the fitness function seem to converge already between 100-200 generations, a more conclusive result could be achieved by simulating the model over longer time with higher population size. Also, since the output signal does not adapt to the amplitude of the input signal, one might reason that it could be some problems in the implementation of the CPG or GA models. Building a CPG model with a GA from scratch took a great amount of time. Much of the time was spent to understand the specifics of the models and algorithms and getting a working prototype, and less time was spent optimising the model. With more time, further investigations of the functionality and optimality of the model could be made. Furthermore, it would probably be more reasonable to optimise each DoF independently instead of all DoF collectively in an individual.

If we now look at the walking cycle GA, while the robot does manage to move forward, it is far from the walking cycle captured from the accelerometers. A simple solution might be to just increase the simulation time - after all, given infinite time, surely the simulation would generate a walking cycle? While this is true, there are several other factors that explain the discrepancy by the desired output and the achieved output. The first aspect is the differences between the simulated robot and the physical one. As mentioned before, the friction is a major difference, but also the movement and dynamics of the robot, creating a major reality gap. In real life, the servos of the Bioloid are heavy, quite stiff and unable to move particularly fast. The 3D model, by comparision, is lightweight, with joints that swivel and move far faster than the real robot ever could. Not only does this result in a 3D simulation that is more unstable (in terms of standing without falling, etc.), but also that the results gained from it is troublesome to translate in the real robot. In the current state, the simulation tends to exploit small, single-joint movements to propel itself forward. But looking at the overall scope of the project, we desire movements that can efficiently transfer the accelerometer data to the simulation, and finally to the real robot. That implies that we need movement cycles robust enough to translate between all three levels, and it is resonable to assume that

would be dynamics that utilize quite large movements of several joints together - like moving all the joints in one leg at once. In order to counteract this, it would be well worth the time to fine-tune the Bioloid model and the V-REP environment, and making sure the simulation environment aligns as closely to the real world as possible. However, not all efforts should be focused on the 3D environment either. More care needs to be taken in regards to the GA parameters, and the fitness function used. In the current GA, the fitness function does not sufficiently punish inefficient walking modes, such as favoring one leg over the other, not keeping the torso straight, or moving the arms in a way that counteracts the forward motion. Finally, a deeper understanding of both GA:s and CPG networks would be required to utilize the full potential of the methods. It is likely that a better choice of parameters in both the GA and CPG would have yielded better results, since they were chosen quite arbitrarily in the current implementation. The pre-processing for the CPG weights took a lot of development and simulation time, and in the end the results did not compare to the time spent developing it. It was also not until the later stages of the project that the oscillating input to the Matusoka was considered, where an earlier implementation could have lead to better results and perhaps allowed us to skip the pre-processing of the CPG altogether. It is also possible that a GA does not result in the best performance, and instead some other form of reinforced algorithm should be considered. However, despite all this the simulation showed clear signs of improvement during the runtime. Due to deadlines the simulation runtime was limited, but it would be interesting to see what results a week-long simulation would generate. A common behaviour of GA:s is to be stuck in a behaviour for a considerable amount of time, and then stumble upon a behaviour with considerably higher fitness. Such jumps were not observed in our simulations, reinforcing the idea that the GA needed more fine-tuning, but perhaps also more runtime.

6 Conclusion

In the end, the simulations did not generate the desired movement patterns, even if they moved the robot forward. Despite this, it is the opinion of the authors that the chosen methods are valid and can potentially result in faster and more "natural" walking cycles than what a randomly seeded simulation would. However, greater care needs to be taken when considering the simulation environment, runtime, parameters for both the CPG and the GA, and the fitness function. The authors encourage anyone interested to continue building on this project, as it is a great learning experience regardless of the generated results. The code base and this report should serve as a sufficient base for anyone considering to continue this work.

7 Further Work

In order to most efficiently improve on the project, CPG model and the fitness function should be the main priority. For instance, incorporating a oscillating tonic input in the CPG model, and a greater focus on punishing unwanted behaviour via the fitness function. The V-REP model should be modified to more accurately reflect the Bioloid robot, as should the interaction between the Bioloid model and the simulation floor. Beyond this, increasing the modularity of the codebase would greatly improve the flexibilty of the project and allow for greater changes to the base of the project. In its current iteration, the GA, V-REP, parameters such as number of used servos, the fitness function, and function used to replay and evaluate generated individuals are all heavily intertwined and in some cases hardcoded. As a result, it is not possible to evaluate other reinforcement algorithms, other simulation enviroments, etc. Evaluation and fine-tuning of the GA and CPG parameters would also be of great importance, as would a more accurate simulation envionment.

References

- [1] Sabine Hauert. *Bio-mimetic Robot Control*, Lectures in Bio-Inspired Artificial Intelligence at University of Bristol, February 16, 2016
- [2] Sabine Hauert. *Artificial Evolution*, Lectures in Bio-Inspired Artificial Intelligence at University of Bristol, February 9, 2016
- [3] G. M. Shephard. *Neurobiology*, 3rd ed. 1994, ch. 20, pp. 435–451.
- [4] K. Matsuoka. *Mechanisms of frequency and pattern control in the neural rhythm generators*. Biol Cybern, vol. 47, no. 2–3, pp. 345–353, 1987.
- [5] G. Taga, Y. Yamaguchi, and H. Shimizu. *Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment*, Biological Cybernetics, vol. 65, pp. 147–159, 1991.
- [6] Mattias Wahde. *Biologically Inspired Optimization Methods: An Introduction*. WIT Press, 2008.
- [7] Guang Lei Liu, Maki K. Habib, Keigo Watnabe, and Kiyotaka Izumi. *Central pattern generators based on matsuoka oscillators for the locomotion of biped robots*. Artif Life Robotics, 12:264–269, 2008.
- [8] Jiang Shan, Cheng Junshi, and Chen Jiapin. *Design of central pattern generator for humanoid robot walking based on multi-objective ga*. In Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, volume 3, pages 1930–1935. IEEE.
- [9] Krister Wolff, Jimmy Pettersson, Almir Heralic, and Mattias Wahde. *Structural Evolution of Central Pattern Generators for Bipedal Walking in 3D Simulation*, October 11, 2006
- [10] Robotis Inc. *Bioloid Premium Kit Assembly Manual*, http://www.robotis.com/download/doc/BIO_PRM_Humanoid_ASM.pdf, Accessed 30/10/2017
- [11] Helga Kristín Ólafsdóttir et al. *Evolving a human-like gait*, <https://drive.google.com/file/d/0By4BzP3-lonJTndmdGSVpJVDg/view>, Accessed 30/10/2017
- [12] Trossen Robotics. *Bioloid Premium Robot Kit*, <http://www.trossenrobotics.com/p/bioloid-premium-robot-kit.aspx>, Accessed 30/10/2017
- [13] Fredrik Gustafsson. *Sensor fusion*, Linköping University, <https://play.google.com/store/apps/details?id=com.hiq.sensor&hl=en>, Accessed 31/10/2017
- [14] Coppelia Robotics, *V-REP Virtual Robot Experimentation Platform*, <http://www.coppeliarobotics.com/>, Accessed 31/10/2017
- [15] Trossen Robotics. *Robotis USB2Dynamixel Adapter*, <http://www.trossenrobotics.com/robotis-bioloid-usb2dynamixel.aspx>, Accessed 31/10/2017

A V-REP Models

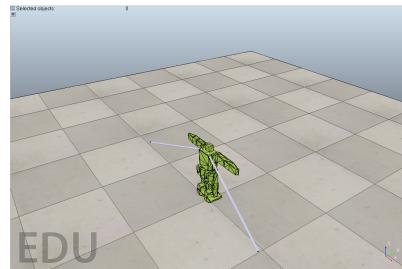


Figure 10: The Bioloid model with 2 support arms in the V-REP simulation environment.

B Parameters for Walking Cycle GA

| | |
|--------------|------------------|
| nIndividuals | 10 to 30 |
| nGenerations | 50 to 70 |
| pTournament | 0.75 |
| pMutation | $1/genomeLength$ |
| pCrossover | 0.5 |

C Processed Data

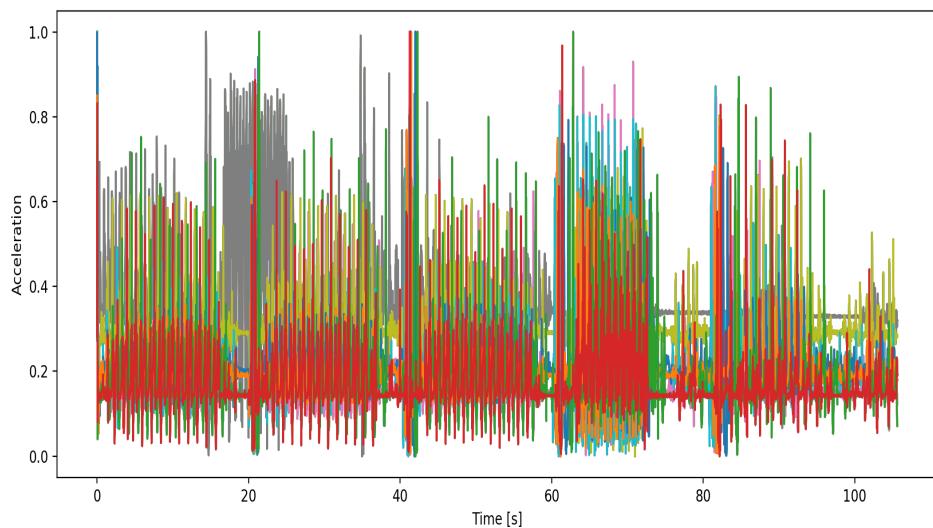


Figure 11: A plot of the collected accelerometer data where the peaks represents the initialisation signals and the different colours represents the different accelerometer signals.

D Results of Pre-Evolution

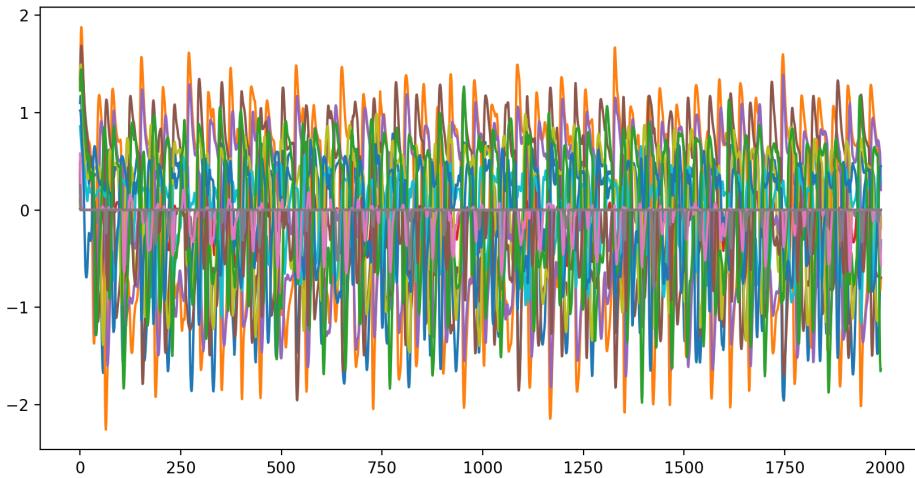


Figure 12: All output signals of the best individual after 500 generations simulated. The x-axis is the simulation time, and the y-axis is the output signal represented in acceleration.

D Results of Pre-Evolution

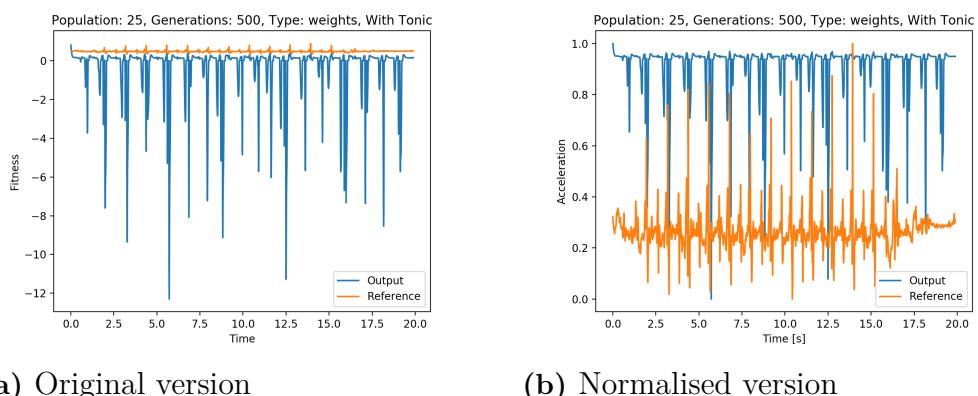


Figure 13: Output signals of the CPG network.