

Investigating the properties of jumping spider morphologies through in-silico brain-body co-optimisation

Ward Van Roeyen

Student number: 01903043

Supervisor: Prof. dr. ir. Francis wyffels

Counsellor: Dries Marzougui

Master's dissertation submitted in order to obtain the academic degree of
Master of Science in Computer Science Engineering

Academic year 2023-2024

Acknowledgment

The author would like to thank Francis Wyffels and especially Dries Marzougui for continued support and feedback during the project.

The ERPY framework, provided by Dries Marzougui along with the Mujoco engine was used to create and simulate the spider model.

The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO and the Flemish Government – department EWI.

Clarification with regards to master thesis

This master's dissertation is part of an exam. Any comments formulated by the assessment committee during the oral presentation of the master's dissertation are not included in this text.

Abstract

Jumping spiders possess the ability to walk, run and jump quickly in any direction to catch their prey. Their legs are not specialized for jumping unlike other arthropods such as grasshoppers and fleas, yet they can traverse up to 20 times their body lengths in a single jump. Research regarding the different species of jumping spiders is scarce and the exact behaviour and jumping technique for many jumping spider species is not well understood. In light of this lack of clarity, we create a simulation environment that emulates jumping spiders. In this environment different species of jumping spider morphologies can be created and simulated to analyse the difference in jumping technique and performance between these species. The first step is to create an in silico-model that mimics a jumping spider and give it parameters that change the morphology and jumping technique. These parameters are defined by examining the morphology and jumping behaviour of real jumping spiders and creating a parameterised morphological structure and control scheme that allow for different morphologies and jumping techniques to be represented with different parameters. Using evolutionary algorithms that perform brain-body co-optimisation, we can co-optimize these parameters for different types of spiders. These simulations and the parameters can be used to create a behavioural descriptor that describes the morphological landscape of different types of jumping spiders as cells in a 2D grid. Quality-diversity algorithms can be used to fill the grid with qualitative solutions of different types of spider morphologies. Analysing these results provides insight into the mechanics and the effects of different morphological features on the spiders jump behaviour without needing to capture and study live spiders. The simulation framework can also be used to test robot designs based on the jumping spider to create a spider-based robot with great mobility. However, this project only provides a simulation of a spiders jump so much future work needs to be done to create a physical robot that could walk, run and jump. The results of these simulations led to a possible new insight into how density distribution affect the jumping technique of jumping spiders.

The code for this project can be found in the following Github repository.

Keywords: Bio-Inspired robotics, Evolutionary Algorithms

Extended Abstract

Ward Van Roeyen
Student
Ghent University
Ghent, Belgium
ward.vanroeyen@ugent.be

Dries Marzougui
IDLab-AIRO Imec
Ghent University
Ghent, Belgium
dries.marzougui@ugent.be

Prof. dr. ir. Francis Wyffels
IDLab-AIRO Imec
Ghent University
Ghent, Belgium
francis.wyffels@ugent.be

I. INTRODUCTION

Jumping spiders are a very unique species that have the ability to walk, run and jump in many directions at very high speeds. Their eight legs and overall symmetry allow them to perform high speed manoeuvres with stability and efficiency [5]. This omni-directional, stable mobility is interesting for robotics as it can be used as a blueprint to create a mobile robot [2] [13]. Despite these traits, jumping spiders are not a well understood or researched species. Only a handful of papers discuss the jumping technique of jumping spiders and these mostly focus on a select set of species. However there are many different types of jumping spiders with many different types of morphologies and behaviours. Researching these creatures is difficult due to the cost of the equipment needed, such as high speed cameras to film their jump, and the specimens that need to be caught in the wild.

In light of this lack of clarity, techniques seen in robotics can be used to gain more insight into the behaviour and morphological traits seen in nature. Simulations can be used instead of researching real specimen that have to be captured and monitored. The simulation environment is created by using data from biology to make sure that the simulated creature behaves similarly to the real creature along with techniques from robotics in order to create and optimise the morphology and controllers that form the simulated creatures. By recreating the creature in a controlled environment, many more experiments can be performed that are otherwise impossible or very difficult. For example, research regarding the effect of a certain morphological trait on the behaviour of a certain creature is difficult to perform because very few creatures are identical aside from the researched trait. When creatures are not identical on all aspects except for the researched trait, it cannot be certain that changes in behaviour are not due to the other differences between the creatures. This principle is called ‘ceteris paribus’ [10]. In a simulated environment, the morphological traits of the creature can be altered directly, allowing for research to be performed with ceteris paribus intact. The results of the simulations can then be verified by researching the biological creatures and the information gained can be used to improve the simulated environment.

This work simulates jumping spiders performing forward jumps. Using these simulations, the behaviour of different species of jumping spiders can be researched in a more cost

effective manner using a test bed with more control that does not require much equipment nor the capture of live spiders. The simulation allows for many different types of morphologies and jumping behaviours to be explored without needing to manually define each one by extrapolating and optimising the behaviour from other simulated spiders into new morphologies. As discussed, the results from these simulations cannot be used as direct evidence and instead serve as a tool to create hypotheses which can be verified by researching real spiders. Results from these comparisons can be used to improve the simulation further resulting in better insights into general jumping spider morphology and behaviour. Additionally the simulations can be used as blueprints for physical robot designs with high mobility. Of course the controller would have to be expanded to include a walking and running controller, since the spider in this work can only jump. The abstracted morphological parts of the spider would also need to be mapped to real, physical components.

The goals of this work are to create a simulation environment for a jumping spider performing a forward jump. Evolutionary algorithms are used to explore the effects of morphological traits on the performance of the spiders and to find optimal jumping spiders that could be used as blueprints for robots. This environment allows us to investigate and speculate about the effects of morphological traits on the behaviour and performance of biological jumping spiders.

The code for this project can be found in the following Github repository.

II. METHODS

This section discusses the different methods used during this work to create the simulated jumping spider environment.

A. Spider Model and Controller

The first step is to create a model of a jumping spider. The model’s morphology has to be parameterised in such a way that many different species of real jumping spiders can be represented using different sets of parameters, while also allowing morphologies not seen in nature to be expressed.

The model is created in Mujoco, an open source simulation software. By examining and researching the morphology of jumping spiders, a model consisting of three types of parts is created. The main body is represented with an ellipsoid, which includes the central structure that connects all parts of the spider. The abdomen of the spider is also an ellipsoid that connects to the main body, this part is important for balance and changes the centre of mass of the creature. Lastly the spiders legs are created as a chain of capsules. Each capsule represents a segment of the spiders leg. These different parts are connected with hinges that allow rotation, the limits of these hinges are chosen based on real spider anatomy [7]. All hinges include actuators that apply torques on the connected body parts, allowing the spider to move. An example of such a spider model can be seen in Figure 1.

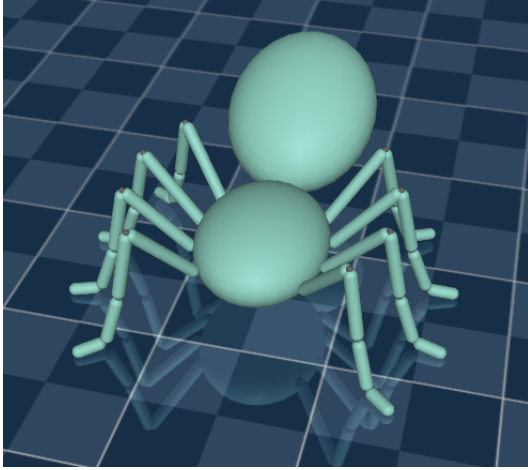


Fig. 1. A spider model in the simulation

The model has several abstractions that deviate from real spider morphology. These abstractions are made since some parts are too difficult to accurately recreate and/or did not contribute meaningfully to the jump mechanics. The main ellipsoid body of the spider includes the jaws and *pedipalps* (which are small arms near the mouth of the spider that aid in feeding and mating), although these parts are not represented as they do not contribute to the jump aside from minor aerodynamic differences [5]. The gripping hairs of the spiders legs are substituted by high friction values for the leg segments, since these hairs are difficult to replicate in simulations. Of all the joints between the 7 segments of a spider leg, only 3 joints are important for analysis of the kinematics [9], therefore the model of 7 segments is reduced to 4 segments. Spiders normally spin a silk thread during their jump to stabilise themselves, this is not included. This thread is important during the landing of a spider and therefore the landings of the simulated spiders are not accurate compared to real spiders [12]. The thread is not used during takeoff, therefore this aspect can still be researched.

The model has many parameters that allow for different types of spiders to be expressed. All of these parameters are shown in Table I.

Body parameters	
Size x	The size of the spiders body along the x axis
Size y	The size of the spiders body along the y axis
Size z	The size of the spiders body along the z axis
Leg angle	The angle between each leg
Leg connector angle	The angle between the body and each legs connector, indicating how high on the body the leg will be
Abdomen parameters	
Size x	The size of the spiders body along the x axis
Size y	The size of the spiders body along the y axis
Size z	The size of the spiders body along the z axis
Leg parameters	
Length	The length of the leg
Thickness	The thickness of the leg
Segment proportions	The proportional length of each segment (Usually kept at 40% <i>femur</i> , 30% <i>tibia</i> , 20% <i>metatarsus</i> , 10% <i>tarsus</i>)
Joint parameters	
Lower range	The minimum angle of the joint
Upper range	The maximum angle of the joint
Stiffness	The stiffness of a joint or how difficult it is to bend it beyond its current value
Damping	the damping of the joint or how much it will dampen forces

TABLE I
THE MORPHOLOGY PARAMETERS

The spider model needs a controller that manipulates its body to perform forward jumps. This controller is defined by studying videos and literature of jumping spiders. The final controller is defined in 5 phases: the pre-jump phase, the lean phase, the jump phase, the flying phase and the landing phase. Each phase is based on actions performed by real spiders. The pre-jump phase is when the spider puts its front legs forward before initiating a jump. The lean phase is when the spider leans into its jump to position itself. During the jump phase, the spider pushes itself off the ground to perform the jump. During the flying phase the spider takes a wide pose to stabilise itself and finally when landing the spider extends its limbs to catch itself. After the jump the spider returns to an idle pose. During these phases, the controller uses the actuators to achieve the same motions that real spiders perform.

This controller has to be parameterised to allow different jumping techniques to be represented by certain sets of parameters. For each phase of the jump, parameters can be set to alter the behaviour during that phase. All of these parameters are shown in Table II. These parameters are used to optimise the simulated spiders jump.

B. Optimisation techniques

Algorithms are needed to optimise the controller, since the initial jump parameters are manually defined by observing the behaviour and setting parameters based on observations. This

Pre-jump parameters	
Time	Duration of the pre-jump phase
Lean parameters	
Time	Duration of the lean phase
Strength	How much torque is exerted on the leg during leaning
Jump parameters	
3rd legs lateral torque	The lateral torque exerted on the third set of legs
3rd legs perpendicular torque	The perpendicular torque exerted on the third set of legs
4th legs lateral torque	The lateral torque exerted on the fourth set of legs
4th legs perpendicular torque	The perpendicular torque exerted on the fourth set of legs
Abdomen perpendicular torque	The perpendicular torque exerted on the abdomen of the spider
Flying parameters	
Time	When of the flying phase starts
Landing parameters	
Time	When of the landing phase starts

TABLE II
THE CONTROLLER PARAMETERS

work uses evolutionary algorithms to optimise the controller. These algorithms generally follow a four step cycle. First a generation of candidate solutions is created, each solution is then evaluated using a fitness function. The fitness function determines how well a candidate performs, for example how far it can walk, how fast it is, etc. After evaluation, the best solutions are kept and their features are recombined to create a new generation of candidates. That new generation is evaluated again and the cycle is repeated until an optimum or other stop criterion such as maximum runtime is reached [4].

To optimise the controller of the spider an evolutionary strategy is used. Evolutionary strategies are a subset of evolutionary algorithms that use sets of real numbers to represent their candidates rather than using a genetic code or other form of representation [1]. The specific evolutionary strategy used is CMA-ES, which stands for Covariance Matrix Adaptation Evolutionary Strategy. CMA-ES uses the covariance between optimal solutions to determine the variance between candidates in the next generation. When determining the next generation it adds random noise to the parameters of the best solutions, the variance of this noise is based on the covariance between the current best solutions [8]. The fitness function for the CMA-ES is defined as the distance jumped from the origin in a straight line. Using the fitness function and the control parameters discussed earlier, CMA-ES can optimise the controller so that the spider performs better jumps.

C. Quality-Diversity

The goal is to explore the relation between morphological traits and performance. To achieve this, qualitative solutions for many different types of morphologies have to be created and examined. Therefore quality-diversity algorithms are used to generate and optimise many different types of simulated spiders. Quality-diversity algorithms are a subset of evolutionary algorithms that focus on exploring the entire search space by providing qualitative solutions for different behaviours rather

than a single optimum. They define an additional behavioural Descriptor (BD) that describes what behaviour a solution shows. The goal of quality-diversity algorithms is to provide solutions for many different types of behaviour to gain more insight into how these solutions function [3].

This work uses the MAP-Elites quality-diversity algorithm. In this algorithm the features of the BD are used as axes on a grid, this grid is divided into cells and for each cell one or more optimal solutions are kept. Therefore an optimal solution for each type of behaviour is kept to illuminate the performance under different behaviours [11]. Using the BD, diversity in certain traits can be forced, for example if the BD has a leg length feature, the different cells in the grid contain spiders with different leg lengths. MAP-Elites uses random noise to copies of the candidate solutions in the cells to create new possible solutions to optimise further and explore the design space. To learn more about the relation between certain traits and their effect on jumping behaviour and performance, the BD is used to create a grid where each cell differs in the examined traits. The solutions in these cells are optimised for performance using the fitness function that measures jump distance. The behaviour in these solutions can then be examined to learn more about which morphological traits lead to which types of behaviour and performance.

III. RESULTS

This section discusses the results achieved during the work.

A. Controller optimisation

The controllers defined for the spiders need to be optimised to achieve ideal jumping behaviour. CMA-ES can be used to optimise the different parameters defined in Table II. The fitness function is the one based on distance jumped discussed in Section II-B. An initial solution for the algorithm is manually created by observing the behaviour for different parameters.

The results of the CMA-ES experiment showed unexpected behaviour and did not improve the controller much beyond the initial solution. The spider would tip itself over to gain a little extra distance. To mitigate this, the fitness function was altered to assign a very negative score when the spider tipped upside down. The new fitness function results in better optimisations that do not tip over, however the CMA-ES algorithm still cannot improve the controller much, even when allowing more runtime and a higher population size.

Presumably, the parameters for the spider model are too sensitive. A small change in jump timing may cause the spider to tip over completely, drastically changing the score. Future works could focus on changing the controller parameters to be less sensitive, making it easier to optimise. For example rather than directly defining the torques on the four legs during the jump, a parameter called ‘jump force’ could be introduced that automatically adjusts the torques on the legs

to achieve the desired force.

The slightly improved spiders created by CMA-ES are used as initial solutions for the experiments using MAP-Elites.

B. Exploring morphology and behaviour

To research the relation between certain morphological traits and the performance of the jumping spider, the MAP-Elites algorithm discussed in Section II-C are used. The fitness function is the same one used in the controller optimisation experiment, distance in a straight line with a penalty for tipping over. The initial solutions are the ones created in the controller optimisation experiments in Section III-A. The behavioural descriptor is set to be the size and relative leg length of the spider. These traits differ a lot between different spider species so it seemed interesting to examine their effects on performance. In a later experiment, joint power is used to see how well the algorithm adapts to stronger spiders and observe the effects of the joint power. The experiments generated and evaluated about 30000 spiders.

A full experiment is performed using the BD and fitness function, the results can be seen in Figure 2. Unexpected behaviour occurred, the simulated spiders would time their landing phase in such a way that they bounced upwards after landing, gaining a little extra distance. Real spiders do not perform such bounces so the fitness function is changed to penalize distance gained after landing. In the grid, a pattern emerged that showed two areas with increased fitness. These patterns cannot be matched to real spider behaviour nor can they be explained easily by examining the jumps performed. One possible explanation was found by examining the time lapse that contained the evolution of the grid. The centre of the areas with optimal solutions are roughly where the initial solutions started. It seemed that the initial solutions created optimal spiders and expanded their offspring in all directions, when the offspring of these simulations reached the same cells they started competing resulting in slower improvements. This video shows the time lapse of the grid for this experiment using a scatter plot rather than a heat map.

Using the changes made to the fitness function in the previous experiment, another experiment is performed. In this experiment the BD uses joint power rather than body size, since the spiders were not jumping as far as real spiders could and to test whether MAP-Elites is capable of training stronger spiders. The results of the experiment can be seen in Figure 3. These results showed unexpected behaviour as well. The simulated spiders tend to crash land by delaying their landing phase to gain as little distance as possible when landing.

The overall results showed that higher joint powers result in a further jump distance. This showed that the jump controller

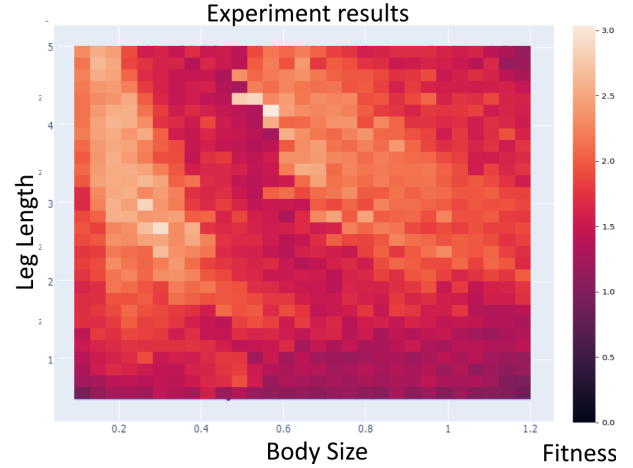


Fig. 2. The results of the full experiment, each square contains one spider controller and morphology with the color indicating the fitness of that spider

is capable of handling these increased powers and that MAP-Elites can optimise them. However, the grid is not completely filled and the solutions with the highest joint powers are not optimised after the experiment. This happens because the spiders with higher joint power need much more precise control to jump properly, making them harder to optimise.

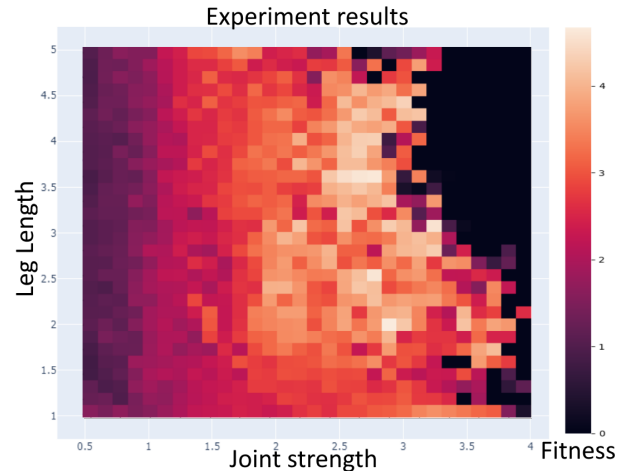


Fig. 3. The results of the full experiment that used joint power

C. Exploring biological relations between morphology and behaviour

To gain more insight into the behaviour and mechanics of real jumping spiders, the last experiment focuses on recreating several biological jumping spider species. The experiment achieves this by using the morphology parameters to create models for the spiders, optimising their jumps and using MAP-Elites to gain insight into the behaviour of species that had not been manually defined.

The first step is to pick some spiders as initial solutions and recreate them in the simulation using the morphology parameters. The species chosen are *Colonus Hesperus*, *Leptofreya Ambigua*, *Phidippus Mystaceus* and *Naphrys Acerba*. These four were selected for their varying leg size and abdomen roundness, which are the traits that the BD was based on in this experiment. The abdomen roundness is defined as how wide the abdomen is in relation to its length, this feature was picked since it varies between different spider species.

Through examination of literature and photos of these spiders, simulated models are created by selecting the morphological parameters. After the models are created, their jumps are optimised using manual selection of parameters and CMA-ES. During the optimisation of their jump controllers, a pattern emerged. The spiders with heavier abdomens usually have to push off the ground at a steeper angle than those with lighter abdomen. This might be because the centre of mass of the spider is further back, so the legs need to push at a higher angle to push through the centre of mass for stability and maximum distance.

After optimisation and with that hypothesis in mind, MAP-Elites is used to explore the behaviour and performance of many different types of spiders based on their abdomen roundness and leg length.

The results of the experiment can be seen in Figure 4. The results show that an abdomen roundness of 0.4 to 0.5, so an abdomen that is about half as wide as it is long, and smaller legs result in better jumping performance. This can be verified or disputed by examining the jump of spiders with these characteristics and seeing how they compare to other species. The hypothesis discussed earlier is not as present in the results of the MAP-Elites experiment, either due to premature convergence [6] or because the analysis that examined for the presence of this hypothesis was not precise enough. The angle of the jump and centre of mass were determined using the morphology and control parameters, however it might be better to improve the simulation to allow for direct observation of these properties.

The behavioural descriptor can also be improved, the different cells of the BD do not currently map directly to a species of spider. A spider with a certain roundness and leg length can still take many different shapes, so the optimal solution in the cell for that roundness and length may not correspond directly to a real spider species. This can be solved by expanding the BD with more features so that every spider species maps to a cell.

Future works could focus on improving different aspects of the simulation. The BD could be improved so that each species of spiders maps to exactly one cell, the fitness function could be altered to focus on different aspects of the jump, the silk thread could be implemented to improve accuracy, etc. All these experiments help improve our understanding of jumping

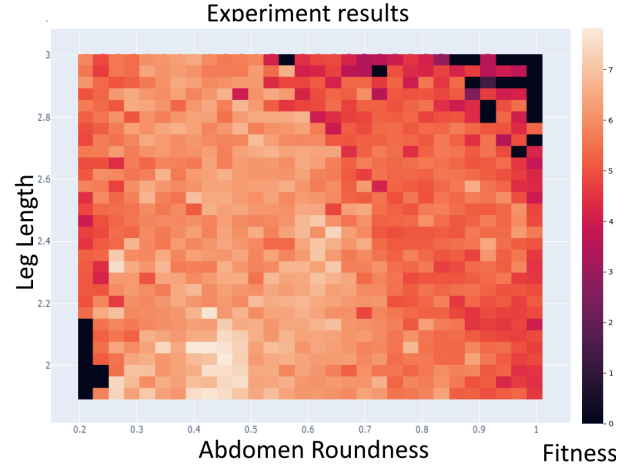


Fig. 4. The results of the experiment that examined biology

spider behaviour.

IV. CONCLUSION

The environment created during this work can simulate jumping spider morphologies. The morphology is parameterised to allow many different types of spiders to be simulated, including ones that relate to different biological species of jumping spiders and additionally allowing for hypothetical morphologies not seen in nature to be explored.

A jumping controller is defined that allows the simulated spiders to jump forward. The controller is based on observations of real jumping spiders and is parameterised to allow different jumping techniques to be expressed. The controller and morphology were co-optimised using evolutionary algorithms. These optimisations show interesting results that can be used for both robotics and biology.

For robotics, the simulation is useful for creating designs of jumping spider robots that cannot create a silk thread. For biology, the simulation is able to recreate the jump takeoff of real jumping spiders to investigate the effects of certain traits on their jump performance and has identified a possibly new hypothesis regarding jumping spider behaviour.

It is also possible to add new controllers to the system to give the spider the ability to walk, run and jump to achieve very high mobility in different environments. With these extra controllers the simulations can be used to model physical jumping spider robots that with advanced mobility and stability.

Biologists can verify the previous hypothesis by studying real jumping spiders. That research can then be used to improve the simulation for better accuracy. Resulting in a positive feedback loop of improvement that increases the

understanding of jumping spiders.

In conclusion the goal of creating a model to investigate jumping spider morphology and its effect on their forward jump was achieved. Future works could focus on adding the silk thread to improve biological accuracy along with additional improvements to the BD and model. Robotics works could focus on adding extra controllers to get a fully mobile spider robot that can act as a blueprint for a jumping spider robot.

REFERENCES

- [1] Schwefel Hans-Paul Beyer Hans-Georg. “Evolution strategies – A comprehensive introduction”. In: *Natural Computing* (2002). URL: <https://doi.org/10.1023/A:1015059928466>.
- [2] Mehmet Çavaş and Muhammad Baballe Ahmad. “A review on spider robotic system”. In: *International Journal of New Computer Architectures and their Applications (IJNCAA) vol 9* (2019), pp. 19–24.
- [3] Konstantinos Chatzilygeroudis et al. *Quality-Diversity Optimization: a novel branch of stochastic optimization*. 2020. arXiv: 2012.04322 [cs.NE].
- [4] Stephane Doncieux et al. “Evolutionary Robotics: What, Why, and Where to”. In: *Frontiers in Robotics and AI* 2 (2015). ISSN: 2296-9144. DOI: 10.3389/frobt.2015.00004. URL: <https://www.frontiersin.org/articles/10.3389/frobt.2015.00004>.
- [5] Brandt Erin E. et al. “Jump takeoff in a small jumping spider”. In: *Journal of Comparative Physiology A* (2021). URL: <https://doi.org/10.1007/s00359-021-01473-7>.
- [6] Luis Eguarte-Morett and Wendy Aguilar. “Premature convergence in morphology and control co-evolution: a study”. In: *Adaptive Behavior* 32.2 (2024), pp. 137–165. DOI: 10.1177/10597123231198497. eprint: <https://doi.org/10.1177/10597123231198497>. URL: <https://doi.org/10.1177/10597123231198497>.
- [7] Chantal Göttler et al. “Spider Origami: Folding Principle of Jumping Spider Leg Joints for Bioinspired Fluidic Actuators”. In: *Advanced Science* 8.5 (2021), p. 2003890. DOI: <https://doi.org/10.1002/advs.202003890>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/advs.202003890>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.202003890>.
- [8] Nikolaus Hansen. *The CMA Evolution Strategy: A Tutorial*. 2023. arXiv: 1604.00772 [cs.LG].
- [9] Stefan Landkammer et al. “Biomimetic Spider Leg Joints: A Review from Biomechanical Research to Compliant Robotic Actuators”. In: *Robotics* 5.3 (2016). ISSN: 2218-6581. DOI: 10.3390/robotics5030015. URL: <https://www.mdpi.com/2218-6581/5/3/15>.
- [10] George V Lauder. “Robotics as a Comparative Method in Ecology and Evolutionary Biology”. In: *Integrative and Comparative Biology* 62.3 (Apr. 2022), pp. 721–734. ISSN: 1540-7063. DOI: 10.1093/icb/icac016. eprint: <https://academic.oup.com/icb/article-pdf/62/3/721/52058146/icac016.pdf>. URL: <https://doi.org/10.1093/icb/icac016>.
- [11] Jean-Baptiste Mouret and Jeff Clune. *Illuminating search spaces by mapping elites*. 2015. arXiv: 1504.04909 [cs.AI].
- [12] Chloe Tenn. “Webless Jumping Spiders Spin Super Strong Silk”. In: (2021). URL: <https://www.the-scientist.com/webless-jumping-spiders-spin-super-strong-silk-69414>.
- [13] R. Vidoni and A. Gasparetto. “Efficient force distribution and leg posture for a bio-inspired spider robot”. In: *Robotics and Autonomous Systems* 59.2 (2011), pp. 142–150. ISSN: 0921-8890. DOI: <https://doi.org/10.1016/j.robot.2010.10.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0921889010001697>.

Table of contents

Abstract	iv
List of Figures	xiii
List of Tables	xv
List of Acronyms	xvi
List of Equations	xvii
1 Introduction	1
2 Background	4
2.1 MuJoCo	4
2.2 Evolutionary algorithms	4
2.2.1 Evolutionary strategies	5
2.2.2 Quality-Diversity	6
2.3 Biology research	6
3 Defining the spider morphology	8
3.1 Body morphology	9
3.2 Abdomen morphology	9
3.3 Leg morphology	10
3.4 Degrees of freedom between body parts	11
3.5 Parameters of the morphology	13
3.5.1 Body parameters	13
3.5.2 Abdomen parameters	13
3.5.3 Leg parameters	13
3.5.4 Hinge parameters	14
3.6 Limitations and abstractions of the morphology	14

4	Defining and optimising the controller	16
4.1	Defining the jump behaviour	16
4.2	Controllable aspects of the spider	19
4.2.1	Actuator controls	19
4.2.2	Timing	20
4.3	Chosen controls and parameters	21
4.4	Optimisation of the control parameters	25
5	Exploring the relation between morphology and jumping behaviour	28
5.1	Experiment Setup	28
5.2	Experiment runs	30
5.2.1	Preliminary Experiments	31
5.2.2	Experiment with bouncing behaviour	32
5.2.3	Experiment with crash landings	34
5.3	Discussion	36
6	Exploring biological relations between morphology and behaviour	37
6.1	Experiment Setup	37
6.2	Initial Spider optimisation	38
6.3	Results	40
7	Future works	43
7.1	Improving the parameterised spider morphology	43
7.2	Improving the Evolutionary Algorithms	44
7.3	Creating a physical robot with full mobility	45
7.4	Expanding the knowledge of biology	45
	Conclusion	47
	References	48

List of Figures

1.1	Ceteris paribus visualised: The compared feature is not the only difference between the examined creatures, therefore ceteris paribus is not fulfilled and researchers cannot be certain that differences in performance are only due to the fin length [1]	2
2.1	Visualisation of the four general phases of evolutionary algorithms [2]	5
2.2	An example of a MAP-Elites grid where the BD separates spiders by body size and leg length, the colour indicates the fitness.	7
3.1	Drawing of a real spider body [3]	8
3.2	Spider body visualised	9
3.3	Difference in body and abdomen shape visualised	10
3.4	Visualisation of the segments of a spider leg [4]	11
3.5	Difference in leg length and thickness visualised	11
3.6	The rotation axes of a leg visualised, the joints can rotate over the axes represented by the blue cylinders.	12
3.7	The rotation axes of the abdomen visualised, the joints can rotate over the axes represented by the blue cylinders.	12
4.1	The different phases of the jump, picture from the-scientist.com [5]	16
4.2	The idle position of a real spider	17
4.3	The pre-jump phase, where the spider aims before a jump, picture from this video	17
4.4	The lean (left) and jump (right) phases, pictures from this video	18
4.5	The flying phase, the third set of legs goes wide to stabilise, picture from this video (note that this spider is flying through the air)	18
4.6	The spider extends its legs to catch itself when landing (left) and then returns to an idle post (right), pictures from this video	19
4.7	Difference in abdomen position under different forces (left = -1 right = +1)	20
4.8	Difference in leg position with different lateral forces, the first, right leg has a force of -1, the other three, right legs have +1. The legs on the left have 0 force.	20

4.9	Difference in leg position with different perpendicular forces, the front 2 right legs have a force of -1, the other two right legs have +1. The legs on the left have 0 force.	21
4.10	Difference in leg position with forces on the segments, the front 2 right legs have a force of +1 on each segment, the other two right legs have -1. The legs on the left have 0 force.	21
4.11	The idle phase before a jump compared to real spiders idle pose. The last segments of the real spiders leg do not bend outward.	22
4.12	The pre-jump phase, where the spider lifts its front legs	23
4.13	The lean phase, where the spider leans forward to prepare for a jump	23
4.14	The jump phase, where the spider pushes itself off the ground	24
4.15	The flying phase, where the 3rd set of legs is widened to stabilise and prepare for landing	24
4.16	The landing phase, where the legs extend to catch the spider as it falls	25
4.17	The spider has landed and returned to an idle pose	25
4.18	The axes of the simulation environment.	26
4.19	The optimal spider after CMA-ES before (left) and after (right) changing the fitness function	27
5.1	An example of a MAP-Elites result. Every square contains one spider morphology and controller and the colour indicates the fitness of that spider. The black squares with 0 fitness were not filled in by the algorithm yet.	30
5.2	The final results of the preliminary experiments	32
5.3	The results of the experiment that showed bouncing landing behaviour	33
5.4	The results of the experiment that showed crash landing behaviour	35
6.1	The real spiders and their simulated counterparts. From top to bottom they are the <i>Colonus Hesperus</i> , <i>Leptofreya Ambigua</i> , <i>Phidippus Mystaceus</i> and <i>Naphrys Acerba</i>	39
6.2	The simulated <i>Leptofreya Ambigua</i> (left) and <i>Phidippus Mystaceus</i> (right), along with an approximation of their COM and the ideal force needed to jump. The spider on the right has a larger abdomen and thus has a centre of mass that is further back and needs to push at a larger angle from the ground.	40
6.3	The results of the biological takeoff behaviour experiment	41
6.4	A plot showing the relation between the abdomen roundness and the fitness of the simulated spiders, the pink was drawn manually to highlight that these results seem to follow a pattern.	41
7.1	A closer look at the spiders jaws and pedipalps, picture from https://squirrelsview.blogspot.com . . .	44
7.2	A scorpion mimic jumping spider, possibly <i>indopadilla sp.</i> Its front legs are much longer than the others [6]	44

List of Tables

3.1	The parameters for the main body	13
3.2	The parameters for the abdomen	13
3.3	The parameters for the legs	14
3.4	The parameters for the hinges	14
4.1	The parameters for the pre-jump phase	22
4.2	The parameters for the lean phase	23
4.3	The parameters for the jump phase	24
4.4	The parameters for the flying phase	24
4.5	The parameters for the landing phase	25

List of Acronyms

BD Behavioral Descriptor.

CMA-ES Covariance Matrix Adaptation Evolutionary Strategy.

EA Evolutionary Algorithm.

ES Evolutionary Strategy.

MAP-Elites Multi-dimensional Archive of Phenotype Elites.

QD Quality Diversity.

List of Equations

4.1	The initial fitness function	26
4.2	The fitness function that prevents the spider from tipping over	27
5.1	The initial MAP-Elites fitness function, the same as Equation 4.2, but normalised on body size . . .	29
5.2	New fitness function, the reward is now a percentage based on the amount of tilt using the max_tilt value which was a value between 0 and 1 where 0 indicated perfect horizontal stability and 1 indicated that the spider tipped over completely. It normalises on leg length and if it was flipped over at any points, the score is always -50.	33
5.3	New fitness function, the distance gained after landing (measured in bounced_distance) is now subtracted from the total distance	34
6.1	Fitness function for this experiment, there is no penalty for distance after landing. The variables do not change after the landing, so that only the takeoff and flying phases are scored.	38

1

Introduction

While many people fear spiders, these interesting and unique creatures have provided much inspiration for robot design [7] [8]. Mainly due to the creature's great stability and its ability to traverse many different kinds of terrain with speed and ease. A subset of spiders, the *Salticid* spiders or jumping spiders have another ability that sets them apart from other spider species. As the name suggests, they are able to perform fast, far jumps allowing them to walk, run and jump across many different types of terrain with ease.

Spiders also have a unique actuation mechanism for their legs. Unlike most animals, spiders do not use complementary muscles to extend/curl their legs. Instead they only have a muscle to curl their legs, but to extend their legs they instead use hydraulic pressure [9]. By increasing the pressure of fluids in their legs they can extend them very powerfully, which is why they can jump such great distances despite their legs not being specifically built for jumping like those of grasshoppers or fleas [10].

However, not much research has been done to investigate the behavioural differences between different species of jumping spider when performing jumps. Therefore the goal for this work is to investigate the dynamics of the jumping spiders jump and explore the effects of different morphological traits on their behaviour and performance using in-silico jumping spiders.

To investigate the relation between morphological traits and jumping behaviour, we need to optimise many spiders with different morphological traits to perform jumps. After optimisation, the controller can be examined to look for relations between the behaviour and morphology. To optimise different types of spiders, brain-body co-optimisation techniques are used. These optimise both the morphology and controller at once, resulting in many different morphologies being explored and optimised. This work uses co-optimisation techniques based on the evolution of creatures in nature called evolutionary algorithms.

Most evolutionary algorithms simulate the robot in its environment and let it perform a task, they assign a certain fitness value that determines how well the robot performs this task, for example how far it walks, how stable it is, etc. The algorithm selects a subset of the robots based on their fitness and combines their traits to create new robots. This new population can then be simulated again and the process is repeated. There are many ways

1 Introduction

to perform optimisation using evolutionary algorithms, mostly dependent on how you pick your fitness, how you represent your candidates and how you pick your next generation from the previous one [11].

Evolutionary algorithms are relatively simple since they don't require any gradients to be calculated due their simulation and evaluation strategy. They are also able to explore many different types of designs using random mutations and other techniques, reducing the probability to be stuck in a local optima and increasing diversity.

After performing optimisation, the resulting simulations of jumping spiders can be used for biological research. An important aspect of biological research is investigating the effect of certain traits in creatures and reasoning how these alter their behaviour and performance in the wild. This is to research why creatures evolved certain traits and helps us understand their evolution better. For example the effect of leg length on the distance that a spider can jump helps us explore how and why some spiders evolved legs of certain lengths.

A common problem in biology is that during research of morphological traits there is a lack of *ceteris paribus*. *Ceteris paribus* is a prerequisite for analysis, it dictates that while assessing the effect of one feature that is different in two creatures, all other features must be identical, otherwise it is not certain that the difference in behaviour is only due to the feature we researched. In nature it is nearly impossible to find creatures that are identical aside from one certain trait. For example, finding two fishes that are identical in all aspects except for their fin length in order to assess the effect of fin length on speed is extremely difficult. Therefore *ceteris paribus* cannot be guaranteed and the difference in behaviour may have an alternate cause [1]. The figure1.1 shows an example of *ceteris paribus*, two fishes are compared on certain traits yet the other traits are not identical.

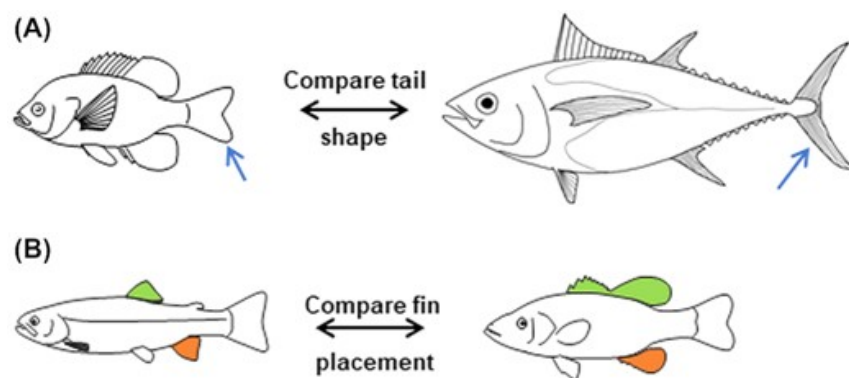


Figure 1.1: *Ceteris paribus* visualised: The compared feature is not the only difference between the examined creatures, therefore *ceteris paribus* is not fulfilled and researchers cannot be certain that differences in performance are only due to the fin length [1]

To analyse the effect of morphological traits on the behaviour of a creature, simulations can be used [12]. These simulated creatures can be adjusted in any way the researcher wants so it is possible to adjust only one trait and keep *ceteris paribus* intact using a controllable test bed. The jumping spider created can thus be used to research the effect of different morphological traits in jumping spiders on their jump performance.

1 Introduction

The simulated spider is not just useful for biology, but can also be used to create a bio-inspired robot. Bio-inspired robots are created by examining at how different organisms perform different tasks in nature and provide us a template for robot design, for example the trunk of an elephant is a very sturdy, yet flexible gripper and can form the basis for a robotic arm design[13]. Therefore the simulated jumping spider can also be used as a blueprint for a mobile spider robot. Other projects have already taken inspiration from jumping spiders to create a robot that can walk, run and jump [14], but this one uses only 6 legs rather than 8.

In summary, the goal of this thesis project is to use brain-body co-optimisation techniques to investigate the effect of different morphologies for a jumping spider performing a forward jump. The steps of this work are to create a simulation of a jumping spider and the using co-optimisation to optimise these simulations. The results can be used to research the effect of jumping spider characteristics on their jumping abilities for biological spiders. A model for a stable jumping spider-based robot can also be optimised to be used as a blueprint for physical robots.

All code for this project can be found in the following Github repository.

2

Background

This chapter provides information about the research, algorithms and programs used within this work.

2.1 MuJoCo

To simulate the spider, MuJoCo is used. MuJoCo is an open-source advanced physics engine and a proprietary of Google Deepmind. It can be used to define the spider model using geometric shapes to simulate body parts along with connecting hinges and actuators to move the different body parts. These different objects have many parameters to simulate many properties, for example the friction on the legs can be increased to mimic the adhesive properties of a spiders leg. Other important aspects of the hinges such as stiffness and damping can also be altered in MuJoCo to achieve a more realistic simulation.

2.2 Evolutionary algorithms

Evolutionary algorithms are based on the principles of evolution and perform brain-body co-optimisation for robot designs. Evolutionary algorithms are used in this work to optimise simulated jumping spiders. They were selected for their simplicity and their relation to biology.

The main principles of evolutionary algorithms are that they evaluate a generation of candidates based on some fitness value and recombine the properties of the best candidates to create a new generation. Evolutionary algorithms generally consist of four phases, which can be seen on Figure 2.1. An initial generation of candidates is created and evaluated using the fitness function. Then a subset of this generation is selected based on their fitness scores, the highest scoring candidates will usually be selected. The features of these selected parents are combined to create new children, which form a new generation of candidates. The algorithm may then add some random mutations to get more variety in the candidates and to explore the design space further. Finally the process is repeated until a stop criterion is reached, this could be that the maximum number of generations is achieved or that there is no improvement between generations, indicating that a (presumably) optimal solution

is found [2].

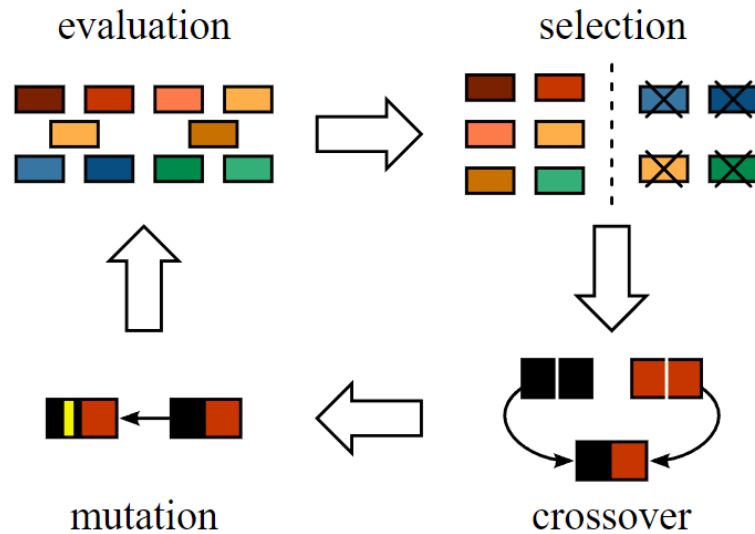


Figure 2.1: Visualisation of the four general phases of evolutionary algorithms [2]

2.2.1 Evolutionary strategies

Evolutionary strategies are a subset of evolutionary algorithms. The main characteristic of evolutionary strategies is that their candidates are represented as sets of real numbers (or parameters) rather than using a genetic code or other form of representation. These parameters can be altered by adding random noise to create new children [15].

CMA-ES

This work uses CMA-ES (or Covariance Matrix Adaptation Evolutionary-Strategy) to perform optimisation. CMA-ES is an evolutionary strategy that uses the covariance between top solutions to determine the variance between the solutions in the next generation. By examining how far apart the top solutions are within the parameter space, the algorithm makes a guess about how close its solutions are to an optimal solution. CMA-ES assumes that when the optimal candidates are close together the global optimum is nearby and if the solutions are spread further apart, the optimum is further away. It uses that knowledge by giving the next generation less or more variance depending on how correlated the previous generation was, that way it can explore further when the solutions are still far from the optimum, yet can also rapidly converge when it nears an optimum [16].

2 Background

2.2.2 Quality-Diversity

Quality-diversity optimisation is an alternative to general optimisation that focuses on illuminating the entire search space by providing multiple qualitative solutions rather than just a single optimum. This is achieved by defining a behavioural space separate from the parameter space. The algorithm will try to find optimal solutions for many different types of behaviour as defined by the behavioural space in order to give more insight into how different behaviours perform and act [17].

MAP-Elites

This work uses the MAP-Elites algorithm to perform QD optimisation. It is used to explore the entire morphological space of jumping spider species and find optimal jumping patterns for different types of spider species without having to manually define them. Like other evolutionary algorithms it uses a fitness value to determine the best solution and creates a new generation from these solutions. The MAP-Elites also defines the behavioural descriptor, which determines what behaviour the solution is showing. The behavioural descriptor maps the solutions to a behavioural space grid, where each cell is linked to a different type of behaviour. and the goal of MAP-Elites is to find an optimal solution within each cell of the behavioural space. By doing so, MAP-Elites will not only show the optimal solution but also shows solutions that are optimal for a certain behaviour in order to get a broader insights into the relation between certain behaviour traits and performance [18].

In this work, MAP-Elites is used to create a grid of different spider types using the BD. The behavioural descriptor is used to separate spider types by their body, mostly by the size of their legs and the shape of their abdomen. This BD allows that different species of spider with different body types fill different cells in the matrix. MAP-Elites will then find an optimal solution for those different types of spiders, allowing analysis of the jump technique that different spiders adapt to perform optimal jumps. An example of such a MAP-Elites grid can be seen in Figure 2.2

2.3 Biology research

In order to make sure the spider model is similar to a real jumping spider, research regarding the biology and behaviour of jumping spiders had to be consulted. Information about the morphology of the spider had to be consulted to create the spider model and to determine which abstractions could be made. Research regarding the general biology of all spiders gave a broad insight into general spider structure [19] [3]. Several sources regarding the joints of spider legs are used to determine the angle ranges of the joints in the spider model [20] [21]. In order to learn more about the specific types of jumping spiders, such as their precise measurements and body shape, different encyclopaedic websites such as the Dick Walton Natural History Services and BugGuide are used [22][23]. Lastly in order to create the controller that manipulates the model to perform jumps similar to real jumping spiders, several sources were used to get specific information about the jump mechanisms [24] [10], but

2 Background

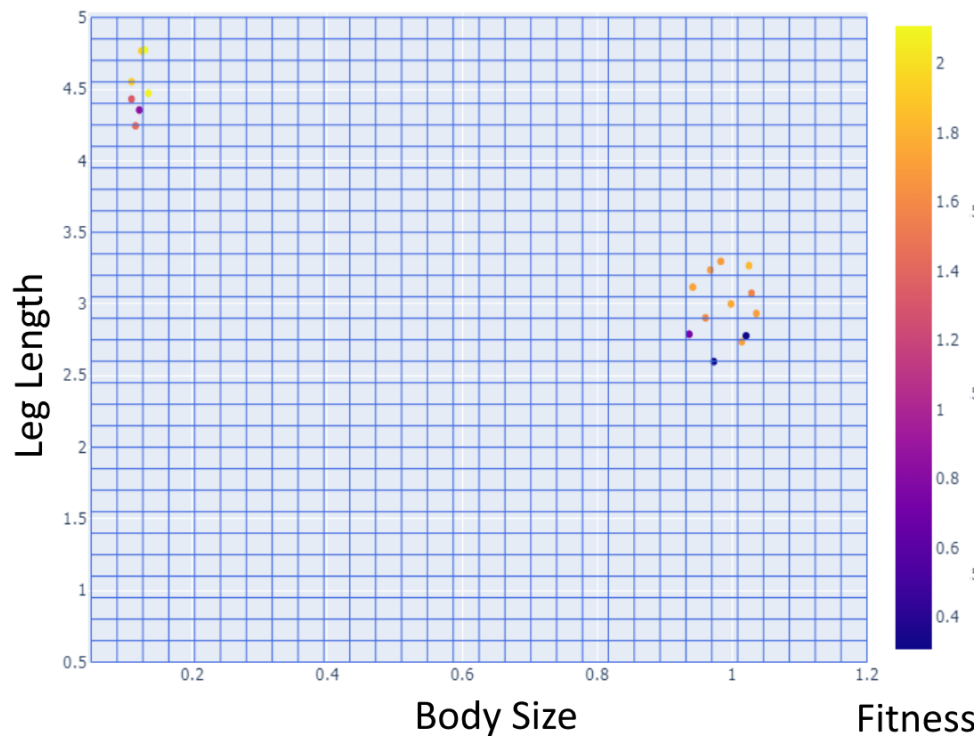


Figure 2.2: An example of a MAP-Elites grid where the BD separates spiders by body size and leg length, the colour indicates the fitness.

most of the insights were gained by watching videos of spider jumps such as this one, observing the behaviour shown in those videos and modelling that for the controller.

3

Defining the spider morphology

To perform the brain-body co-optimisation the body of the spider model has to be defined. The morphology of the spider needs to be highly parameterised to allow many different types of spider to be represented using different parameters. The model also has to closely match to an actual jumping spider morphology, however simulating an exact spider model is too complex so some abstractions need to be made.

In the following sections each part of the simulated spider will be described. Along with the parameters that can influence their morphology. All these parameters can be found in Section 3.5. Figure 3.1 displays a drawing of a general spider morphology for reference.

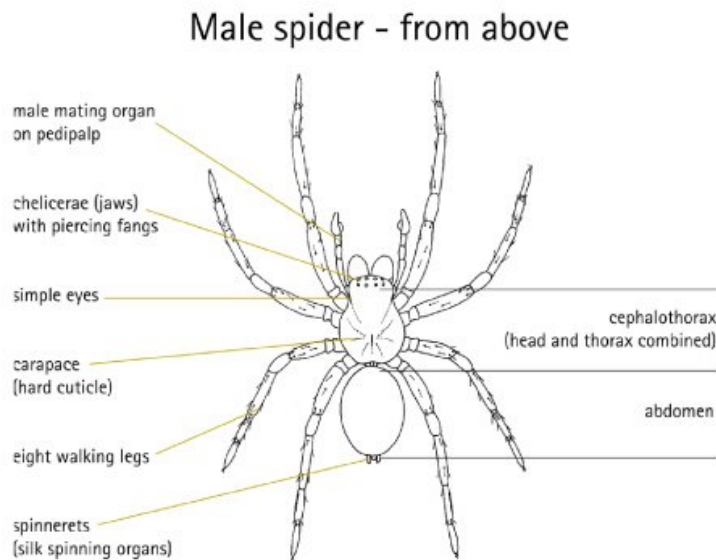


Figure 3.1: Drawing of a real spider body [3]

3 Defining the spider morphology

3.1 Body morphology

Unlike other insects, the head and body of spiders are combined into one part called the cephalothorax [19]. This is the central structure of the spider that all other body parts connect to. Their bodies usually have a somewhat rounded shape, but the exact shape varies depending on the species of spider. At the front of the body are the jaws and *pedipalps*, these do not contribute to the spiders jump or locomotion so they are not included in the spider model.

The main body of the spider is represented by an ellipsoid with adjustable shapes to represent the different species seen in nature. The shape of the ellipsoid can be altered by 3 parameters, its size along the y-axis and its size along the x- and z-axes relative to the y-size. To illustrate, if a spider's size along the y-axis is 2 and the x- and z-values are .5 and .25 respectively, the size of the ellipsoid would be 1, 2, .5 for the x-y- and z-diameters respectively. By keeping sizes relative to another, the spider bodies created by the evolutionary strategies will have a more realistic shape. Another parameter controls how far apart the legs are in radians.

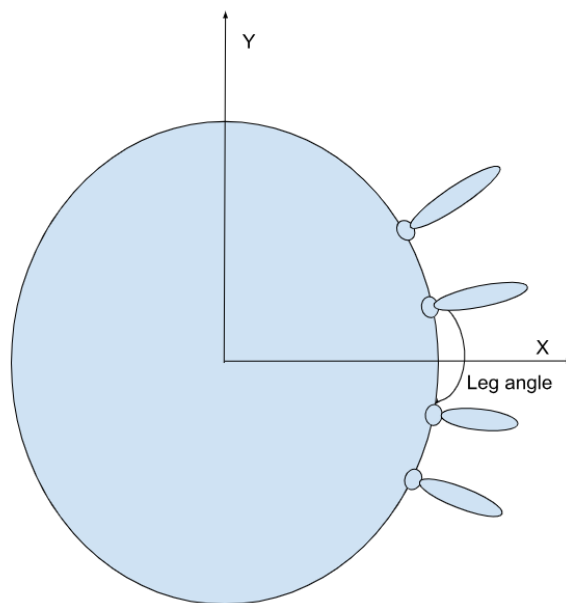


Figure 3.2: Spider body visualised

3.2 Abdomen morphology

The abdomen of the spider attaches at the back of the body and has a vaguely ellipsoid shape, its shape and size depends on the specific species of spider.

It is represented by an ellipsoid much like the body. The x- and z-sizes are relative to the y-size, while the y-size is relative to the main body's y-size to prevent the spider from having an abnormally large abdomen relative to

3 Defining the spider morphology

the main body. The abdomen attaches to the body through a hinge at the back and can be rotated in the x- and z-planes. it is important to include as it is vital in aerodynamics and balance during the jump.

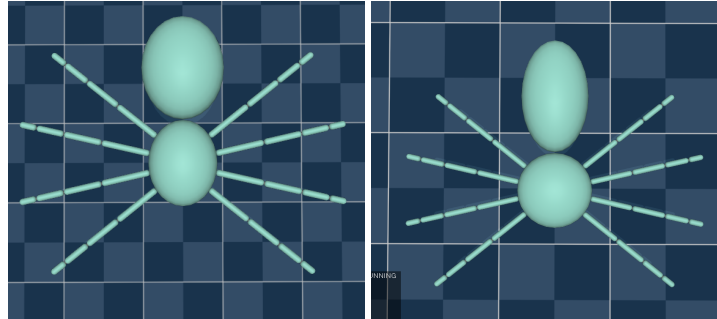


Figure 3.3: Difference in body and abdomen shape visualised

3.3 Leg morphology

Spiders have eight legs, four on each side spaced equally on the sides of the spiders body. Real spider legs consist of seven segments of different sizes attached to one another, as can be seen in Figure 3.4[19]. In order from the base of the spider to the tip of the leg these segments are named the *coxa*, *trochanter*, *femur*, *patella*, *tibia*, *metatarsus* and *tarsus*. These seven segments have seven joints between them and the main body. For functional analysis of the kinematics, only three joints are important, the hip joint between the leg and the main body, the *femur-patella* joint and the [?] joint [21]. Therefore the *coxa*, *trochanter* and *patella* segments are abstracted away and instead are fused to the main body, the *femur* and the *tibia* respectively. The *metatarsus*, *tarsus* joint is kept as it needed to improve the friction of the spiders leg when standing on the ground, this is discussed in Section 4.3.

Each segment is a capsule that is attached to the next segment by a hinge allowing it to move their pitch and yaw relative to the previous leg segment. The range of these hinges is quite large, going up to 160°, as real spiders have a specialized membrane between the femur and patella that allows their legs, despite being an exoskeleton, to bend at great working angles. [20]

The shape and size of the legs can be altered through their thickness and length parameters, see Figure 3.5.

As explained in the introduction, the actuation of real spider legs is performed by hydraulic pressure. However

3 Defining the spider morphology

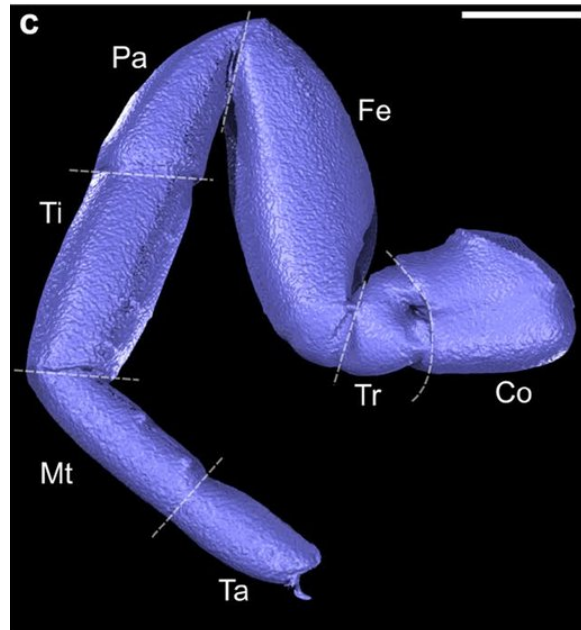


Figure 3.4: Visualisation of the segments of a spider leg [4]

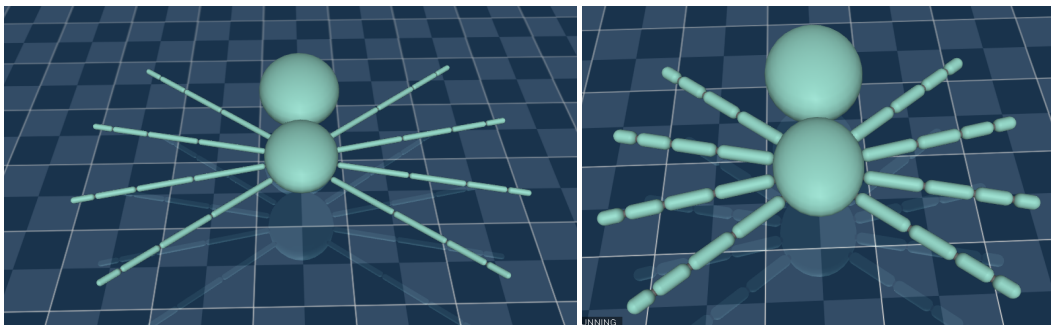


Figure 3.5: Difference in leg length and thickness visualised

for the simulation these forces are simulated as basic torques. [21]

3.4 Degrees of freedom between body parts

The different degrees of freedom in the joints are based on the joints of real jumping spiders [20] [21]. The joints connecting the legs to the body can rotate in 2 axes, allowing them to move laterally and perpendicularly to the main body. See Figure 3.6 for a visualisation of the rotation axes.

The only other degree of freedom is at the abdomen joint. This joint allows the abdomen to move in 2 axes, similar to the legs [3]. However it should be noted that the axis allowing lateral motion is never used during a jump as its movement is not symmetrical to the jump direction and thus always results in a less straight jump. See Figure

3 Defining the spider morphology

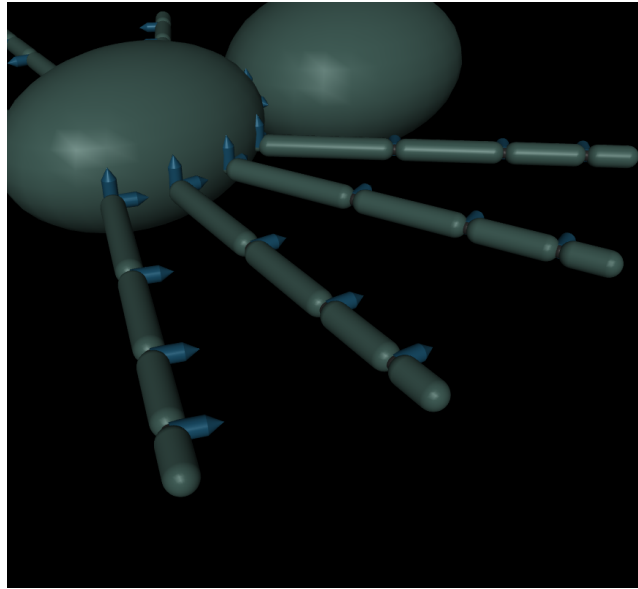


Figure 3.6: The rotation axes of a leg visualised, the joints can rotate over the axes represented by the blue cylinders.

3.7 for a visualisation of the rotation axes.

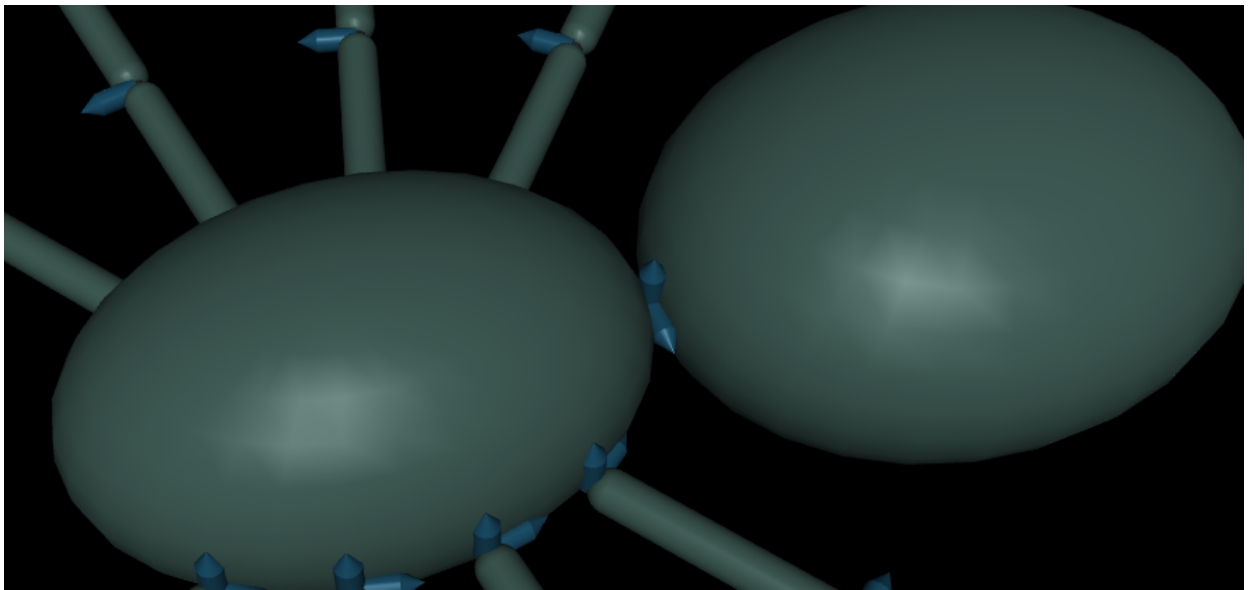


Figure 3.7: The rotation axes of the abdomen visualised, the joints can rotate over the axes represented by the blue cylinders.

3 Defining the spider morphology

The stiffness, damping and ranges for each of these joints can also be altered via parameters.

3.5 Parameters of the morphology

This section shows tables with all the parameters for each body part of the morphology.

3.5.1 Body parameters

Table 3.1 shows all the parameters for the main body of the spider.

During experiments, only the size parameters are altered. The leg angles are kept constant for simplicity.

Size x	The size of the spiders body along the x axis
Size y	The size of the spiders body along the y axis
Size z	The size of the spiders body along the z axis
Leg angle	The angle between each leg
Leg connector angle	The angle between the body and each legs connector, indicating how high on the body the leg will be

Table 3.1: The parameters for the main body

3.5.2 Abdomen parameters

Table 3.2 shows all the parameters for the abdomen of the spider.

Size x	The size of the spiders body along the x axis
Size y	The size of the spiders body along the y axis
Size z	The size of the spiders body along the z axis

Table 3.2: The parameters for the abdomen

3.5.3 Leg parameters

Table 3.3 shows all the parameters for each leg of the spider.

3 Defining the spider morphology

Length	The length of the leg
Thickness	The thickness of the leg
Segment proportions	The proportional length of each segment (Usually kept at 40% <i>femur</i> , 30% <i>tibia</i> , 20% <i>metatarsus</i> , 10% <i>tarsus</i>)

Table 3.3: The parameters for the legs

Lower range	The minimum angle of the joint
Upper range	The maximum angle of the joint
Stiffness	The stiffness of a joint or how difficult it is to bend it beyond its current value
Damping	the damping of the joint or how much it will dampen forces

Table 3.4: The parameters for the hinges

3.5.4 Hinge parameters

Table 3.4 shows all the parameters for each of the hinges between the different body parts of the spider.

3.6 Limitations and abstractions of the morphology

Some features of a jumping spider are not modelled for this work due to being either too difficult to implement or not important enough for the kinematics of a jump.

Real spiders use a silk thread when performing jumps as a safeguard when falling and to stabilise themselves when landing [5]. This work does not implement this thread and as such the landings of the simulated spiders are not as stable as real spider and thus not accurate. This means that the spider model can only be reliably used for takeoff mechanics when comparing to biological spiders, since the thread does not contribute to the jump during takeoff. However the model can still be useful for robot design as it would be difficult to implement a silk thread in a synthetic spider robot.

As discussed in Section 3.3, the 7 segments of the spider leg are abstracted to 4 segments, since only 3 joints are important for analysis of the jump [21]. However, the last segment is kept to help with friction during the jump, this is discussed in Section 4.3. Another aspect of spiders that is not modelled is the hair on their legs and body. These hairs are an important aspect of the spider that allows them to stick to walls and traverse in any direction. For this work the hairs are substituted by high friction values to mimic their stickiness.

Several body parts at the front of the spider, such as the jaws and *pedipalps* are not modelled in the simulation. Since these parts don't contribute to the jump aside from slightly different aerodynamics they were not deemed

3 Defining the spider morphology

necessary.

4

Defining and optimising the controller

The controller is responsible for manipulating the simulated body of the spider so that it performs a forward jump. In order to define this controller, real spiders are examined to determine which patterns are present in their jumps. These patterns are translated into phases of a control scheme. To make the spiders body move, actuators that exert torques on the joints are used. These actuators perform different actions during the phases in the control scheme. Parameters are implemented that change how and when these actuators act to allow for different jumping behaviours to be performed.

4.1 Defining the jump behaviour

The forward jump of the spider is manually determined by examining papers and videos of real spiders in order to create a control scheme that replicated their behaviour during a jump. The jump scheme is defined in five phases: pre-jump, lean, jump, flying and landing. Figure 4.1 shows the different phases for a real jumping spider. The phases are discussed in chronological order in the following sections.

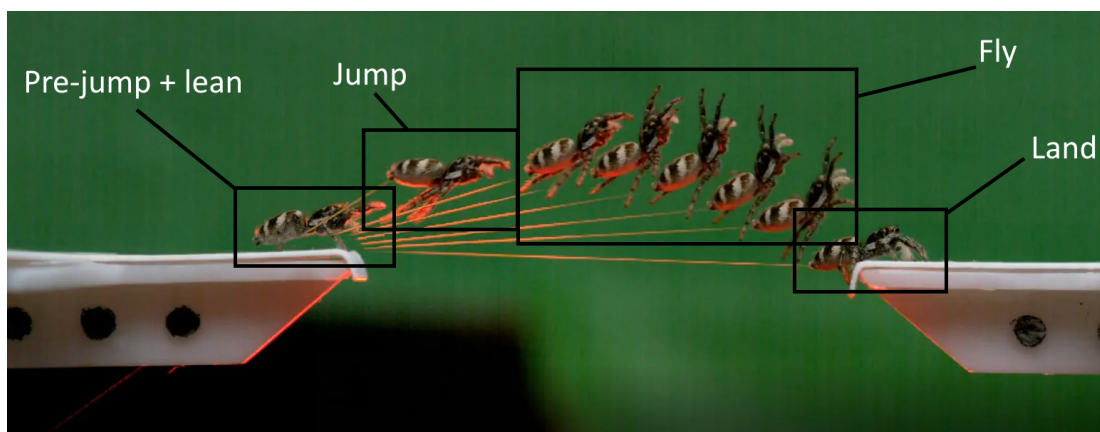


Figure 4.1: The different phases of the jump, picture from the-scientist.com [5]

4 Defining and optimising the controller

Idle start The idle pose that spider takes before initiating a jump, which is not a phase of the jump itself. It still needs to be defined for the simulated spider as it the starting position of any jump.



Figure 4.2: The idle position of a real spider

Pre-jump phase When the spider initiates a jump, it starts in the pre-jump phase. During this phase, the spider lifts its front set of legs. Nearly all spiders perform this action before initiating a jump. This behaviour is shown in Figure 4.3.



Figure 4.3: The pre-jump phase, where the spider aims before a jump, picture from this video

Lean phase The lean phase is performed just before a jump, during this phase the spider leans forward and starts lifting its second set of legs before commencing a jump. This phase determines at what angle the spider pushes off the ground. This behaviour is shown in Figure 4.4.

Jump phase The jump phase is the most important phase of the entire jump sequence. During this phase the spider extends both its hind legs to push off the ground and perform the forward jump. This behaviour is shown in Figure 4.4.

4 Defining and optimising the controller



Figure 4.4: The lean (left) and jump (right) phases, pictures from this video

Flying phase The flying phase commences very shortly after lifting off the ground. During this phase the spider widens its third set of legs in order to stabilise itself and prepare itself for a landing. This behaviour is shown in Figure 4.5.



Figure 4.5: The flying phase, the third set of legs goes wide to stabilise, picture from this video (note that this spider is flying through the air)

Landing phase The landing phase happens just before the spider lands on the ground. During this phase the spider extends its front legs in order to catch itself as it nears the ground. Afterwards the spider will simply return to the idle pose and the jump is complete. This behaviour is shown in Figure 4.6. It is important to note that real spiders use a silk thread drag line to stabilise their landing, this is not modeled in the simulation and as such the landings in the simulations do not accurately represent real jumping spiders, more information about this limitation can be found in Section 3.6.

4 Defining and optimising the controller



Figure 4.6: The spider extends its legs to catch itself when landing (left) and then returns to an idle post (right), pictures from this video

4.2 Controllable aspects of the spider

To make the in-silico spiders perform the same actions as real spiders, controllable aspects are needed that can be used to make the spider move its body in any way necessary to perform the jump. A set of actuators is added that can move the different body parts by applying torques on the hinges. The timing between these movements is also important during the jump, jumping too early or too late can result in drastically different results. The body movements and the timing between them need to be controlled by parameters that can be optimised by the evolutionary algorithms. These parameters should allow for different jumping behaviours to be expressed.

4.2.1 Actuator controls

The hinges between each body part of the spider have motors that can apply torques to move the attached body parts. These actuators are simple force controllers in Mujoco that apply torques to the hinges in a certain direction. These forces can be powered between a control range of -1 to 1 force to allow force in both directions and allow small forces by using decimal strengths. The strength of these actuators depends on the mass of the attached body parts, but Mujoco does not have built-in units so the forces. The power of these forces was determined experimentally by trying different values and seeing which values move the segments at a natural speed.

These actuators are present on each of the hinges that are discussed in Section 3.4. The effects of different forces on the hinge between the abdomen and main body are visualised in the following Figure 4.7.

The legs have two forces, one force allows it to move laterally to the main body and another allows it to move perpendicularly. The effects of these lateral forces can be seen in Figure 4.8 and the perpendicular forces in 4.9.

Between each two leg segment is a joint connecting both, these joints can be either curled or extended, by using

4 Defining and optimising the controller



Figure 4.7: Difference in abdomen position under different forces (left = -1 right = +1)

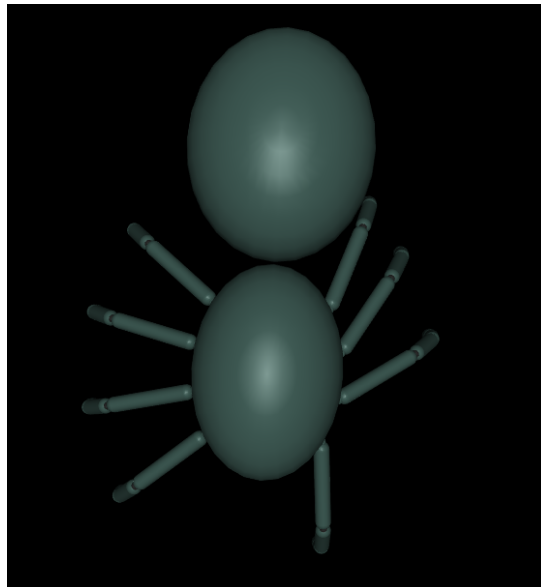


Figure 4.8: Difference in leg position with different lateral forces, the first, right leg has a force of -1, the other three, right legs have +1. The legs on the left have 0 force.

perpendicular forces. The same torque based forces used in other joints is used for these despite the hydrodynamic actuation present in real spiders to simplify the simulation. Real spider legs only have a single muscle controlling all segments to curl and the hydrodynamic forces to extend, so there is no need for lateral movement between these segments. The effects of these forces can be seen in Figure 4.10 [21].

4.2.2 Timing

The jump is a linear sequence of movements, thus the controller changes the torques on hinges in a sequence of phases, which are discussed in Section 4.1. The timing between these changes can be altered in the controller optimisation. Timing is very important during the spiders jump, being slightly too slow or too fast when performing

4 Defining and optimising the controller

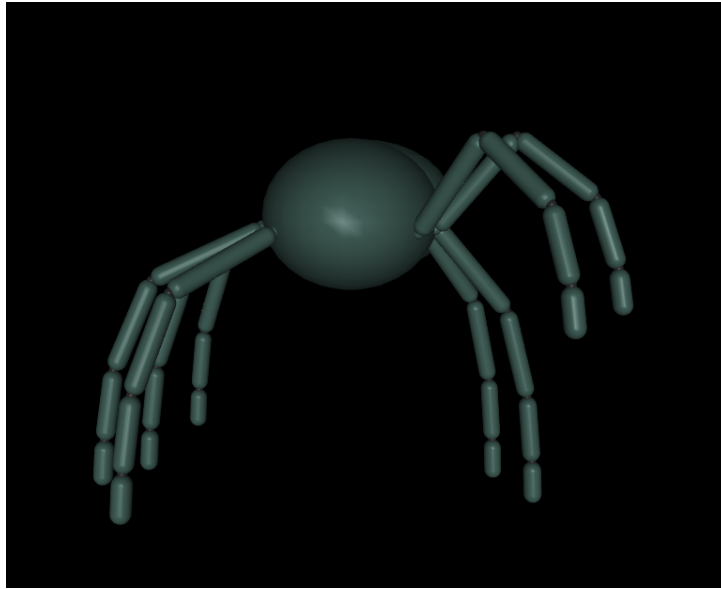


Figure 4.9: Difference in leg position with different perpendicular forces, the front 2 right legs have a force of -1, the other two right legs have +1. The legs on the left have 0 force.

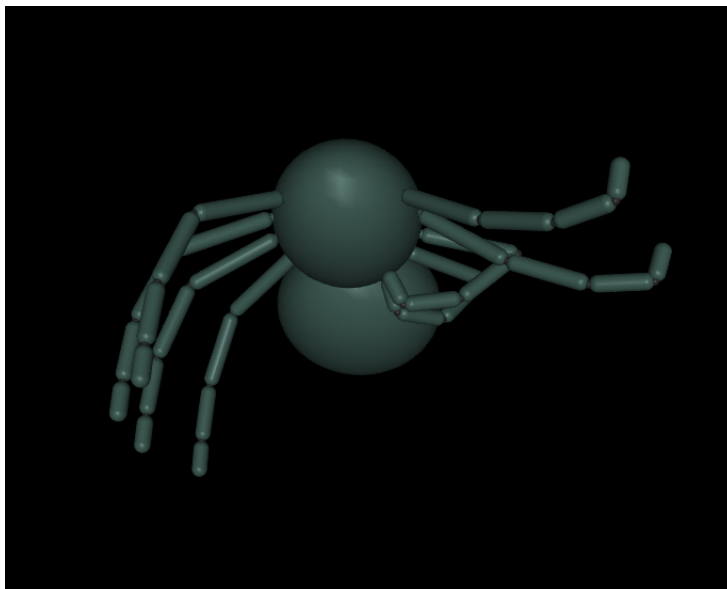


Figure 4.10: Difference in leg position with forces on the segments, the front 2 right legs have a force of +1 on each segment, the other two right legs have -1. The legs on the left have 0 force.

an action can lead to the jump failing entirely.

4.3 Chosen controls and parameters

This section discusses how the phases that are defined in Section 4.1 can be implemented in the simulated spider using the controllable aspects defined in Section 4.2. For each phase, the actuators apply torques so that the

4 Defining and optimising the controller

simulated spider performs the same actions a real spider would, for example in the pre-jump phase, the front leg actuators move the legs upwards. To allow different jumping techniques to be explored by the evolutionary algorithms, each phase has parameters that alter the behaviour during that phase. The change of torques between phases are linear transitions with a small transition time, to prevent sudden movements that could destabilize the simulation.

Idle start pose The idle pose is a simple idle pose that the spider takes before initiating a jump. One difference between the models idle phase and real spiders is that the final segments are bent outwards, this is to improve the grip with the floor as the spider would otherwise slip too easily, this difference is shown in Figure 4.11.

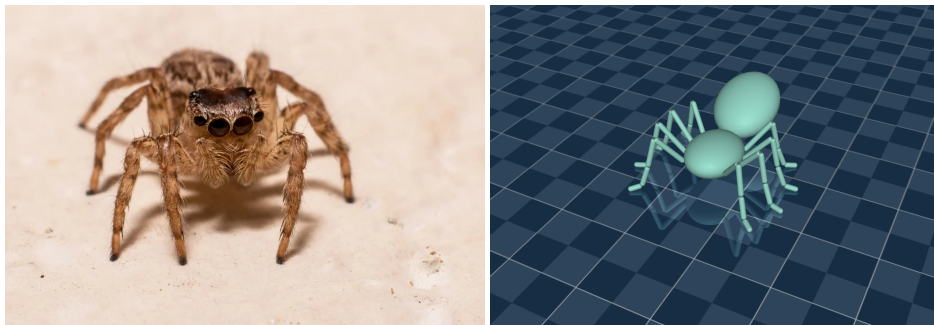


Figure 4.11: The idle phase before a jump compared to real spiders idle pose. The last segments of the real spiders leg do not bend outward.

Pre-jump control and parameters During the pre-jump phase, the actuators of the front legs drive these legs to lift upward and stretch out. The following Table 4.1 has all the parameters for the pre-jump phase. It is only the timing, as this phase does not contribute much to the actual takeoff of the jump. Figure 4.12 shows the actions of a simulated spider during this phase.

Time	Duration of the pre-jump phase
------	--------------------------------

Table 4.1: The parameters for the pre-jump phase

Lean control and parameters During the lean phase the 2nd set of front legs will start lifting upward and stretching slowly while the 4 hind legs are actuated to shift backwards. By shifting the legs backward they are set in position to push off the ground. The following Table 4.2 has all the parameters for the lean phase. This phase has a timing and strength parameter. These were chosen because the lean phase is important for determining the angle at which the spiders legs push off the ground. By picking timing and strength the exact position of the spider when jumping can be controlled more precisely. Figure 4.13 shows the actions of a simulated spider during this phase.

4 Defining and optimising the controller

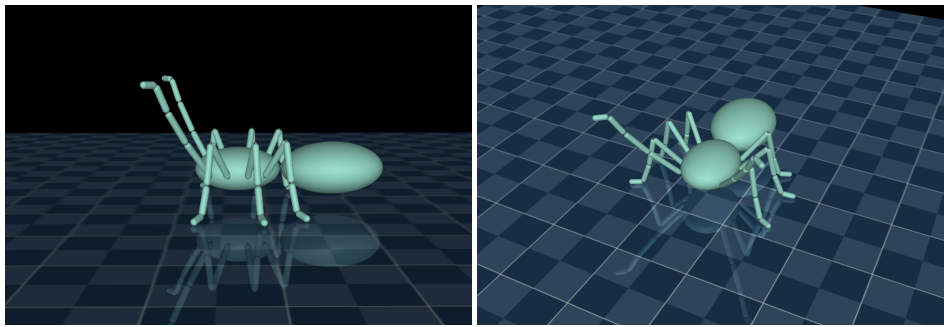


Figure 4.12: The pre-jump phase, where the spider lifts its front legs

Time	Duration of the lean phase
Strength	How much torque is exerted on the leg during leaning

Table 4.2: The parameters for the lean phase

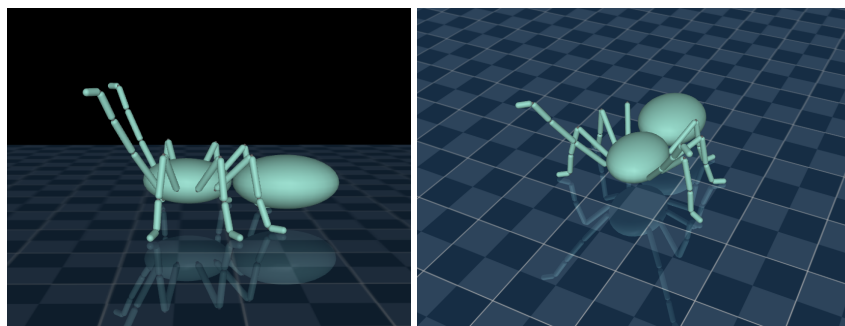


Figure 4.13: The lean phase, where the spider leans forward to prepare for a jump

Jump parameters During the jump phase the hind leg actuators quickly extend the legs in order to push the spider off the ground. The front legs curl up and bend inward to improve stability. The following Table 4.3 has all the parameters for the jump phase. Since this is the most important phase of the jump it has extra parameters to control the exact actuation of the hind legs and abdomen. Figure 4.14 shows the actions of a simulated spider during this phase.

Flying control and parameters During the flying phase, the actuators set the third set of legs to a wide pose to stabilise the jump. The following Table 4.4 has all the parameters for the flying phase. The only parameter for this phase is the time when it activates, expressed by how long it should wait after jumping. Figure 4.15 shows the actions of a simulated spider during this phase.

Landing control and parameters During the landing phase, the spider will first extend all of its legs in order to catch itself before falling. Afterwards it enters an idle phase again and the jump is complete. The following Table

4 Defining and optimising the controller

3rd legs lateral torque	The lateral torque exerted on the third set of legs
3rd legs perpendicular torque	The perpendicular torque exerted on the third set of legs
4th legs lateral torque	The lateral torque exerted on the fourth set of legs
4th legs perpendicular torque	The perpendicular torque exerted on the fourth set of legs
Abdomen perpendicular torque	The perpendicular torque exerted on the abdomen of the spider

Table 4.3: The parameters for the jump phase

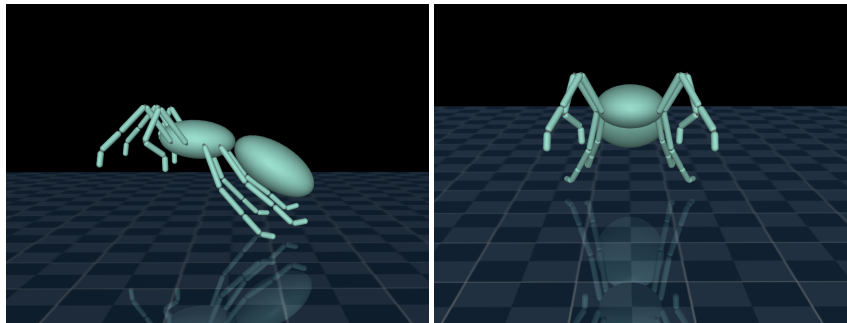


Figure 4.14: The jump phase, where the spider pushes itself off the ground

Time	When of the flying phase starts
------	---------------------------------

Table 4.4: The parameters for the flying phase

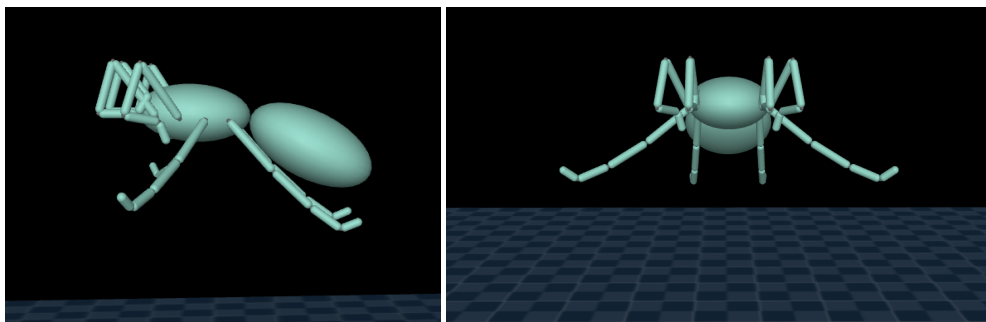


Figure 4.15: The flying phase, where the 3rd set of legs is widened to stabilise and prepare for landing

4.5 has all the parameters for the landing phase. The only parameter for this phase is the time when it activates, expressed by how long it should wait after entering the flying phase. Figures 4.16 and 4.17 show the actions of a simulated spider during this phase.

4 Defining and optimising the controller

Time	When of the landing phase starts
------	----------------------------------

Table 4.5: The parameters for the landing phase

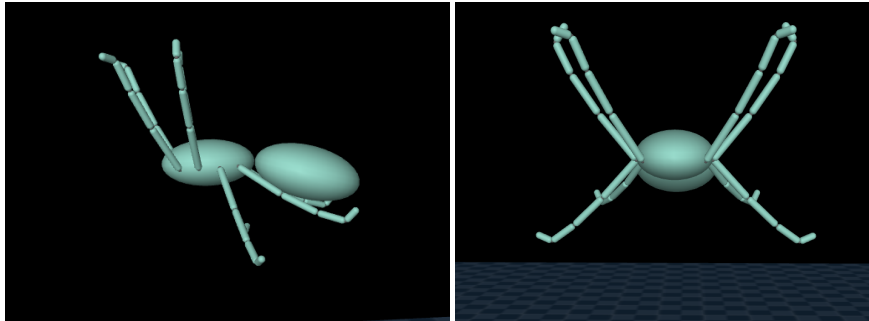


Figure 4.16: The landing phase, where the legs extend to catch the spider as it falls

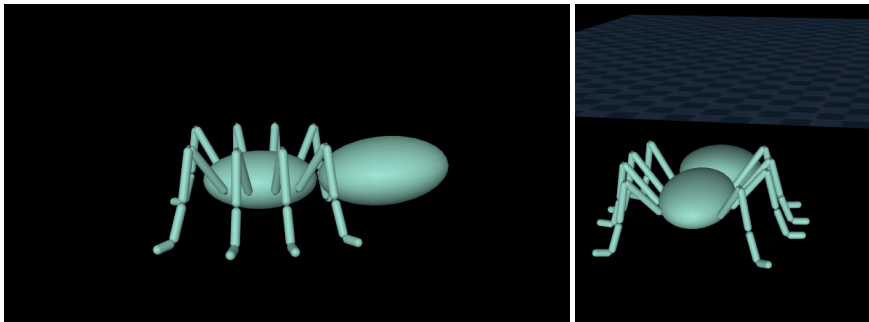


Figure 4.17: The spider has landed and returned to an idle pose

4.4 Optimisation of the control parameters

CMA-ES is used to optimise these control parameters. See Section 2 for more information about CMA-ES. The initial solution of the jump is manually defined by picking parameters and observing the jumping behaviour. CMA-ES is used to find an optimum for the parameters with maximum fitness, starting from the initial solution. CMA-ES can get stuck in local optima so the optimum found is not necessarily the global optimum. The fitness function was initially the distance jumped along the y-axis. To ensure that the spider jumped in a straight line, a penalty was added that gives a negative score depending on how far the spider strays from the y-axis, by measuring its position on the x-axis. The penalty for deviation from the y-axis was doubled, since these deviations were usually small. See Equation 4.1 for the full fitness function. Figure 4.18 shows the x- and y-axes, the distance jumped is measured on the y-axis and the deviation from a straight line is measured on the x-axis.

Unexpected behaviour showed up during initial experimentation. If the spider tips over after landing it gains extra distance from tipping, this lead to the spider tipping over in all optimal solutions. The fitness was adjusted to assign a negative score if the spider is upside down, see Equation 4.2. Figure 4.19 shows the results of the new

4 Defining and optimising the controller

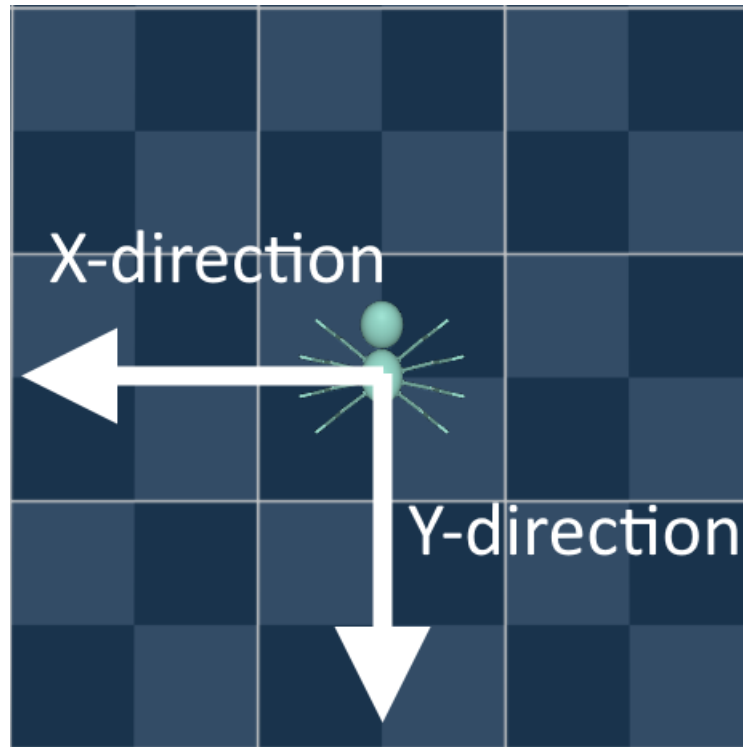


Figure 4.18: The axes of the simulation environment.

$$fitness = distance_to_origin - 2 * deviation_from_y_axis \quad (4.1)$$

Equation 4.1: The initial fitness function

fitness function versus the initial one.

The fitness function changed multiple times during experiments. More details about the changes and clarifications for why these changes are necessary is discussed in Section 5.2.

After performing CMA-ES, it had not altered the initial controller much. Aside from small changes in the abdomen position that provided a little extra distance, the jump remained mostly the same. The controller provided by the result of the final CMA-ES experiment is used in the MAP-Elites experiments as an initial solution, discussed in Chapter 5.

4 Defining and optimising the controller

$$\begin{aligned} \text{fitness} &= -50 \text{ (when upside down)} \\ \text{fitness} &= \text{distance_to_origin} - 2 * \text{deviation_from_y_axis} \end{aligned} \tag{4.2}$$

Equation 4.2: The fitness function that prevents the spider from tipping over



Figure 4.19: The optimal spider after CMA-ES before (left) and after (right) changing the fitness function

5

Exploring the relation between morphology and jumping behaviour

To examine the relation between the morphology, the controller and the performance of the simulated spider, experiments are performed to co-optimize the spiders morphology and controller using quality-diversity. The different aspects of the morphology that can be optimized were initially the body shape and size, the abdomen shape and size, the thickness and length of the legs. In later experiments joint power is also used as a parameter for optimization. The morphological parameters, discussed in Section 3.5 and the controller parameters, discussed in Section 4.2, are optimized in tandem. The main goal of these experiments is to explore the relations between jumping behaviour and certain morphological traits. While performing experiments the information gained was often used to improve the fitness function and simulation model itself by examining which jump patterns emerged when certain fitness scores are enforced.

5.1 Experiment Setup

For these experiments MAP-Elites is used since it is able to illuminate the entire search space by providing qualitative solutions for many different behaviours. These solutions are examined to learn more about the overall relations between certain morphological traits and performance. By defining the behaviour as the trait that we want to examine, MAP-Elites generates optimal spiders that are all diverse in that trait. The result is a grid where each cell contains an optimal spider with a different version of the examined trait based on its position in the grid. MAP-Elites optimized the spiders as a whole, adapting both controller and morphology at once.

The controller and morphology parameters that could be adjusted are the ones discussed in sections 3.5 and 4.2.

The MAP-Elites algorithm is implemented as follows. The fitness function was initially the one that had been used in the previous CMA-ES experiments in Equation 5.1 of Section 4.4. The fitness function returns the distance along the y-axis (the distance jumped), minus two times the distance along the x-axis (the deviation from a straight line) as the total score. Since this experiment included morphological changes, the fitness has to be normalised

5 Exploring the relation between morphology and jumping behaviour

$$\begin{aligned} fitness &= -50 \text{ (when upside down)} \\ fitness &= \frac{distance_to_origin - 2 * deviation_from_y_axis}{body_y_size} \end{aligned} \quad (5.1)$$

Equation 5.1: The initial MAP-Elites fitness function, the same as Equation 4.2, but normalised on body size

based on the morphology of the spider. Otherwise larger spiders always beat smaller ones, since they can jump further. The final score is divided by the y-size of the spider to normalise, the normalised function can be seen in Equation 5.1. The fitness determines the performance of the spider and a higher fitness should always indicate an improvement.

The behavioural descriptor is defined as the size and relative leg length of the spider. This behavioural descriptor forces the experiment to return spiders of many different shapes and sizes. All other morphological and control parameters can be freely adjusted, so it lays no restrictions on how the jumps should be performed, nor the shape of the body or the abdomen. Size and leg length are chosen since these seem to be vastly different in different species of jumping spiders. This BD can be determined directly from the morphological parameters and requires no extra computation. A grid is created where each cell corresponds to a certain range within the body size and relative leg length behavioural features. For each cell one optimal solution is kept and if a new child surpasses the optimum in that cell, it is replaced by that child. Figure 5.1 shows an example of such a MAP-Elites result, this grid is the result from the experiments in section 5.2.1. Each square represents one spider morphology and controller and the colour shows the fitness of that spider. Black squares do not contain solutions, because the algorithms children did not reach those features.

Our implementation of the MAP-Elites algorithm uses a previous generation to generate the next generation. Therefore, an initial generation of spiders has to be provided, these are then simulated to perform their jumps and a fitness score is assigned. The behaviour of these solutions is determined from their morphology parameters and they are placed on the grid. The initial solutions for the different MAP-Elites experiments are either spider models that had been defined and optimised using CMA-ES in the previous sections or models that emerged as optimal models during a previous MAP-Elites experiments.

After providing an initial generation, the MAP-Elites algorithm generates new children by randomly picking one of the current candidates in the grid. A copy of this candidates is made and changed by adding zero-mean random Gaussian noise to each parameter to create a new child. The variance of the noise and range of values the parameter can take is set individually for each of the parameters. After generating a new child using random noise, the child is evaluated using the fitness function and its behaviour is determined. If the new child outperforms the current candidate in the cell that corresponds to its behaviour, it replaces this candidate, otherwise it is discarded.

5 Exploring the relation between morphology and jumping behaviour



Figure 5.1: An example of a MAP-Elites result. Every square contains one spider morphology and controller and the colour indicates the fitness of that spider. The black squares with 0 fitness were not filled in by the algorithm yet.

This loop is repeated for several hundreds of generations, each generation consisting of 32 or 64 children that are evaluated in parallel. This results in several thousands of spiders being simulated and evaluated. After every 50 generations, the current grid with candidate solutions is saved to achieve a time lapse of the development during the experiment. These experiments run on the HPC clusters with multiprocessing, in the end most experiments ended with approximately 30 thousand spider designs simulated.

When completed the experiment provides a full time lapse of the grid where each cell contains the set of parameters for an optimal spider with different behaviours for different cells. Through examination of the grid, new insights into the jumping technique shown by the spiders and the effect of the traits in the BD are gained.

5.2 Experiment runs

The results of the different experiments are discussed in the following sections. Each experiment consisted of some initial, small scale testing to assure that any changes are working properly and prevent wasted compute time. Afterwards a longer run on a HPC node is performed to simulate many spiders and examine the results. The results are always periodically logged to assure that data was still collected in the event of a crash, to monitor

5 Exploring the relation between morphology and jumping behaviour

whether a long experiment is working correctly in the early stages and to gain a time lapse of the experiment.

5.2.1 Preliminary Experiments

The goal of these first experiments is to get an initial exploration of the search space to see whether more improvements had to be made to the model. Therefore, these are not run for as long as the other experiments, only generating about 5000 spiders. These experiments were performed before the timing parameters discussed in Section 4.3 were introduced, so the behaviour of the spiders during this run was more rigid and had fewer parameters to optimise. These experiments use the initial fitness function shown in Equation 5.1. If the spider tips over the reward is replaced by a negative penalty and if the spider deviates from a straight line when jumping forward, the distance to the straight line is also subtracted from the total distance travelled.

Unexpected behaviour occurred during the experiments due to the fitness function. Since the fitness is only penalized if spiders tipped over, they tend to tilt upward in order to fall forward when landing. This along with the controller aiming the front legs upwards before landing means the spider gained extra distance when it tilted. The following video shows the effect of this issue. Another issue is that if a spider performs a full somersault, the penalty for tipping upside down is not applied since the spider is not tipped over at the end.

The final result of these experiments can be seen in Figure 5.2.

The trend for these experiments is that spiders with longer legs have a better fitness, this is because the fitness is normalized on body size and not leg size. Bigger legs have increased power in their actuators and allow for further jumps. The spiders that are far to the upper-right of the BD have a decreasing fitness, due to the size boundary being too large. The spiders that are too big did not function properly in the simulation and thus could not jump forward correctly. A similar effect can be seen for the spiders that have small legs, the legs are so small that these spiders did not jump at all and instead paddled forward using their abdomen, resulting in a lower fitness. It also became clear that the control scheme (which does not include timings yet) is not sufficient to define jumping behaviours for all different types of spiders. The sizes of different morphological aspects are also not relative to other aspects yet. The x-, y- and z-size of the body are within ranges 0.1 to 2 without any regard for their relative size to one another, this resulted in spiders with unnatural shapes.

After the experiments the timing parameters are added to allow more control over the different phases of the jump. These control how much time separates each phase of the jump and are discussed in-depth in Section 4.3. Smoothing is added between phases since the speed of some motions causes the spider to jerk in a certain direction which could cause flipping. The smoothing is performed by changing some transitions to be linear over some time, which is also explained in Section 4.3. The sizes and proportions are limited by making most sizes relative to the y-size of the body to prevent unnatural designs that provide no useful information, such as spiders with too small legs that only their abdomen pushed them forward during a jump. The maximum size of models

5 Exploring the relation between morphology and jumping behaviour



Figure 5.2: The final results of the preliminary experiments

is also reduced since the bigger spiders are too big to properly simulate. Lastly the fitness function is changed to apply a penalty depending on how much the spider tilts upwards during its jump along with normalisation based on leg length instead of body length and returning -50 if the spider is tipped over at any point. The fitness function can be seen in Equation 5.2.

5.2.2 Experiment with bouncing behaviour

Thanks to the changes made after the previous experiment, the jumps performed by spiders became much more stable. The designs created by MAP-Elites are also more grounded in reality and within the bounds of what the simulation could reasonably handle. The goal of this next experiment is to use the previous changes and test if they are sufficient to generate good jumping spiders for many different morphologies. This experiment ran longer than the previous one, stopping after generating about 20000 spiders. The designs generated showed improvements, however a new strategy was abused to gain more distance. The spiders made sure that when they landed, they would use their landing pose to pull themselves forward a bit after bouncing on the floor. This results in an unstable, bouncy landing but since those bounces add to the distance, they are preferred. This behaviour can be linked to the fact that our model does not have a drag line to stabilise its landing, real spiders use their

5 Exploring the relation between morphology and jumping behaviour

$$fitness = -50 \text{ (when upside down at any point)}$$

$$fitness = \frac{(distance_to_origin - 2 * deviation_from_y_axis) * max_tilt}{leg_length} \quad (5.2)$$

Equation 5.2: New fitness function, the reward is now a percentage based on the amount of tilt using the max_tilt value which was a value between 0 and 1 where 0 indicated perfect horizontal stability and 1 indicated that the spider tipped over completely. It normalises on leg length and if it was flipped over at any points, the score is always -50.

silk line to stop themselves from bouncing further when landing. An example of the bouncing behaviour can be seen in this video, which shows the spider with the highest fitness of that experiment. It bounces after landing, to gain extra distance.

The final result of the experiment can be seen in Figure 5.3.

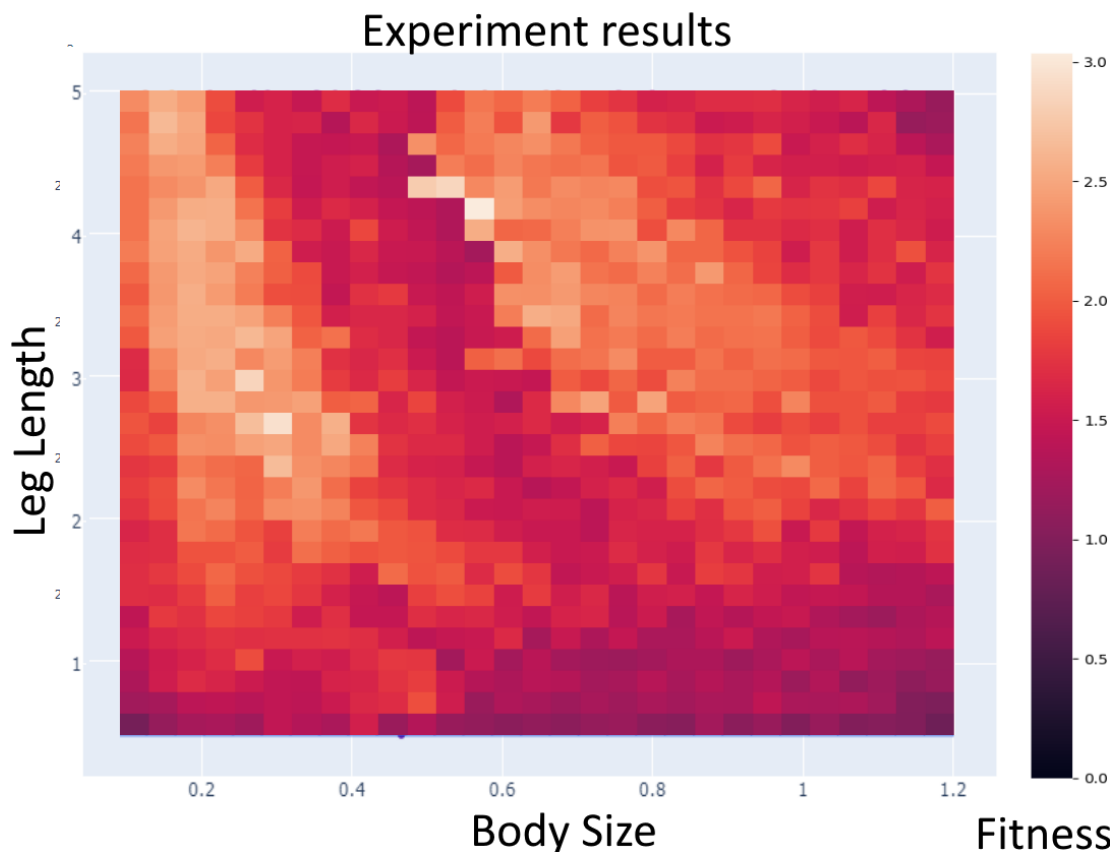


Figure 5.3: The results of the experiment that showed bouncing landing behaviour

Because the range of certain morphological traits such as size is tightened, the experiment does not generate

5 Exploring the relation between morphology and jumping behaviour

$fitness = -50$ (when upside down at any point)

$$fitness = \frac{(distance_to_origin - 2 * deviation_from_y_axis - bounced_distance) * (1 - max_tilt)}{leg_length} \quad (5.3)$$

Equation 5.3: New fitness function, the distance gained after landing (measured in `bounced_distance`) is now subtracted from the total distance

spiders that the simulation is unable to handle unlike the previous experiment. However this grid has some areas with better jump performance that are not easily linked to any real-life phenomena. By examining the time lapse, it seemed that the areas with lower fitness scores occur where the solutions generated by the initial solutions collided. To clarify, in the first few generations, most of the grid is empty so the first few generations that stem from the initial solutions are immediately accepted if they land in empty cells. One initial solution starts on the left side of the grid, another on the right. During these first generations their children expand outward into the grid and eventually they collide, it seems that the fronts where they collided had an overall lower fitness. This video shows the time lapse of the grid for this experiment using a scatter plot rather than a heat map.

After the experiment changes are made to the model to improve the jumping behaviour of the simulated spiders. Mainly the fitness function is changed to include a penalty for any movement made after the spider lands on the ground, any distance made by these bounces reduce the fitness greatly. The penalty is determined by figuring out when the spider has landed and all extra distance gained after landing becomes a penalty. This change can be seen in Equation 5.3.

5.2.3 Experiment with crash landings

In the previous experiments the spider models were not jumping nearly as far as real jumping spiders would. In order to get the simulation closer to real jumping spiders, the joint power is added as a parameter. This joint power parameter simply acts as a multiplier on all joint strengths, meaning the spider can exert more force and thus jump further. The joint power replaces the size in the behavioural descriptor, to examine the effect of stronger actuators on the jumping technique of spiders. The distribution of fitness values from this experiment should be predictable, higher joint powers should result in further jumps. The main goal is to see if the MAP-Elites algorithms is able to properly optimise spiders with higher joint power and see if the higher powers caused a change in jump technique. Before running the full experiment, a smaller run is performed to get an initial solution with a higher jump power for better exploration. This small run consists of a MAP-Elites experiment using the initial solutions of the previous experiments, these all had a joint power of one, but a different leg length so they did not occupy the same cell in the grid. After a while the grid is filled with solutions for higher joint powers, these are then used

5 Exploring the relation between morphology and jumping behaviour

as initial solutions in a longer run.

Due to the implementation of the bounce penalty introduced after the previous experiment, another strategy emerged. Spiders tend to make sure that their landing phase happened after they hit the ground. By doing so they stop all their momentum when landing in order to avoid the bounce distance penalty. An example of such behaviour can be seen in this video, which shows the spider with the highest fitness of that experiment. It touches the ground before it goes to a landing pose to stop all momentum.

The final result of the experiment can be seen in Figure 5.4.

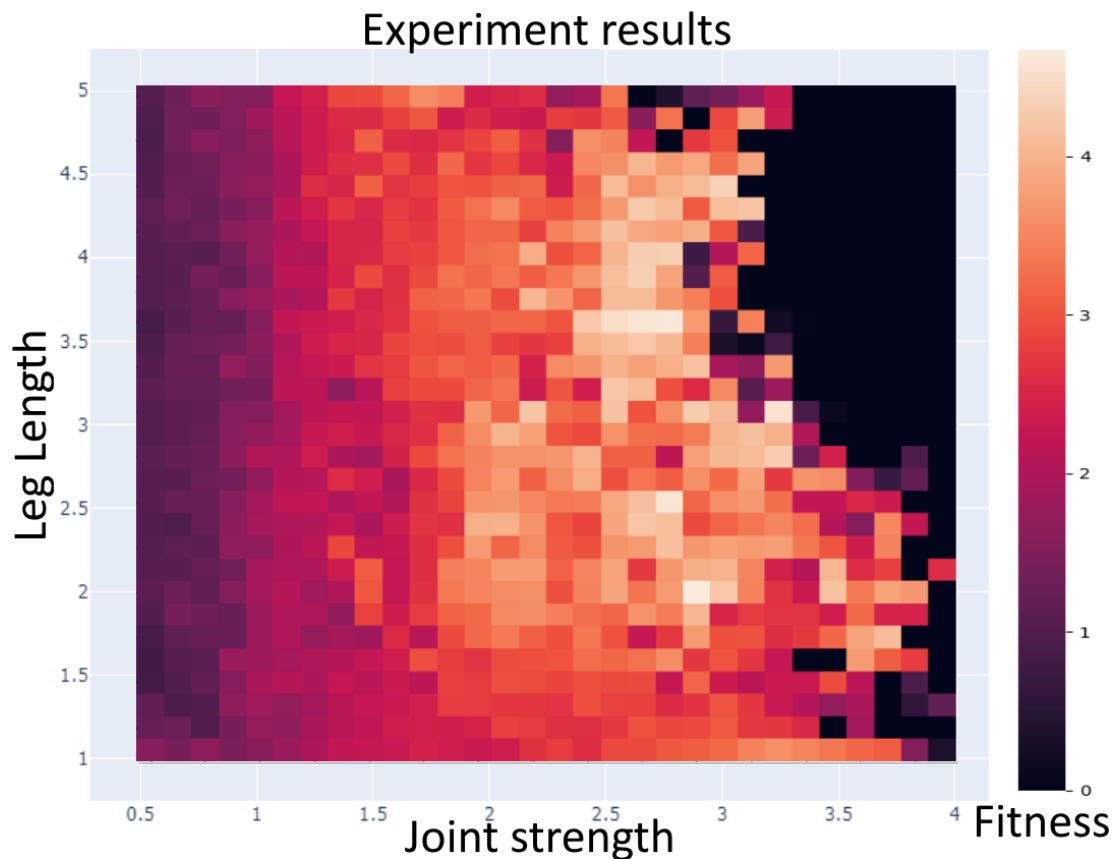


Figure 5.4: The results of the experiment that showed crash landing behaviour

The grid shows that there is a linear relation between jump power and the performance of the spider. Cells with higher joint power have a higher fitness scores. Due to the normalization on leg length, higher leg length did not seem to influence the overall fitness scores, resulting in a vertical fade of increasing fitness. However the simulation has more trouble optimizing spiders with higher joint strength. The cells with high joint strengths are filled slower and it takes more generations to fully optimise the solutions in these cells, resulting in the black edge at the right side of the figure. This could be due to the higher joint strength resulting in snappy movements that require very precise timing parameters. The overall morphology of the different spiders, such as their size

5 Exploring the relation between morphology and jumping behaviour

and body shape under different joint strengths is mostly the same. This could be a case of premature convergence, where the morphology is always the same because it is difficult to deviate from it without altering the controller drastically [25]. Or that specific morphology is optimal for jumping and is thus repeated in every cell.

5.3 Discussion

The MAP-Elites experiments show that the jump controller is very capable of making different types of spiders jump. Changes made in the fitness function are always correctly reflected in the performances of the different spiders. However, the system also excels at finding ways to fool the fitness function by gaining more distance in undesired manners. Mainly the landing is a problem due to the lack of a silk drag line that is present in real spiders. Unfortunately, the fitness changes that were made were combating symptoms one by one rather than improving the system as a whole. It could have been better to implement aerodynamics or landing force into the fitness to force the spider to jump and land better as a whole rather than mitigating one symptom at a time.

The morphologies that are created by the MAP-Elites also turn out to be mostly similar, especially in the last experiment since size was no longer part of the BD. The designs created are usually closely related to the ones defined initially, aside from small changes in the abdomen and the differences forced through the BD. The overall shape of the body, abdomen and thickness of the legs are usually very similar. This can be due to two reasons. The similar morphologies that emerged might be optimal for jumping as far as possible and the big differences in morphologies that emerges in nature happen due to other factors such as camouflage, mate attraction, body functions, etc. Another reason can be that MAP-Elites reaches a premature convergence, where morphologies cannot change easily because the controller cannot adapt quickly enough to the new morphology, this issue has been documented in many evolutionary algorithms [25].

The final jumps performed by different spiders are however very stable in the air and can reach far distances, these simulations can be useful as blueprints for robot designs. These experiments are difficult to link to biology, due to the absence of the drag line which is very important for the landing. Much of the fitness alterations were made for the landing, so these issues and the changes made to combat them cannot be linked to biology.

6

Exploring biological relations between morphology and behaviour

This experiment is performed to research the behaviour expressed by different types of simulated spiders during jump takeoff, in order to gain more insight into the jumping behaviour of real jumping spider species. Quality-diversity is used to illuminate a search space containing many different types of spiders in order to examine how they perform jumps. Only the jump takeoff can be examined, the landing would require the silk thread to be simulated. The experiment is performed by first selecting four spiders from nature, recreating their morphology in the simulation environment, optimising their jump with CMA-ES and using them as initial solutions for quality-diversity. After quality-diversity finishes creating new spiders of varying shapes and sizes, their performance and controller parameters are examined to learn more about their behaviour. The experiment resulted in a new hypothesis regarding spider behaviour during their jump takeoff. Namely that there is a relation between the centre of mass of the spider and the angle of the jumping force exerted during the jump.

6.1 Experiment Setup

To research the behaviour of many different spiders without needing to manually define each spider, the MAP-Elites quality-diversity algorithm is used to illuminate the entire space based on several initial solutions. The implementation of the MAP-Elites algorithm is similar to the one discussed in Section 5.1, with two alterations made. Firstly the BD uses relative leg length and abdomen roundness, rather than body size or joint power. The abdomen roundness is defined by the x-size of the abdomen (which is relative to the y-size of the abdomen), since this controls how wide the abdomen is. This change is made because the abdomen and relative leg length seem to vary a lot between different jumping spider species. The fitness function is also altered. Since this experiment only examines the takeoff, tipping over after landing no longer causes a penalty, only the stability while flying is important. The fitness function can be seen in this Equation 6.1.

Our implementation of the MAP-Elites algorithm needs some initial solutions to create the next generations.

6 Exploring biological relations between morphology and behaviour

$$\begin{aligned} fitness &= -50 \text{ (when upside down at any point before landing)} \\ fitness &= \frac{(distance_to_origin - 2 * deviation_from_y_axis) * (1 - max_tilt)}{leg_length} \end{aligned} \quad (6.1)$$

(variables cannot change after the landing phase)

Equation 6.1: Fitness function for this experiment, there is no penalty for distance after landing. The variables do not change after the landing, so that only the takeoff and flying phases are scored.

These initial solutions are based on real spider species and are ideally spaced far apart in the BD grid, meaning their leg size and abdomen roundness need to be diverse. Four spider species are selected as initial solutions, the *Colonus Hesperus*, *Leptofreya Ambigua*, *Phidippus Mystaceus* and *Naphrys Acerba*. These four are selected for their varying leg size and abdomen roundness. simulated morphologies for these spiders have to be manually created by using the morphology parameters to make sure that they are as similar as possible to the real biological spider. The spiders can be seen in 6.1 alongside their simulated counterparts.

These spider models first need controllers that are optimised to perform jumps. The initial controllers are created using manual parameter selection and then optimised using CMA-ES. Afterwards the spiders are used in the MAP-Elites experiment, which generates new spider models with varying types of abdomens and leg lengths. By examining the performance and control parameters of these spiders, new insights into the behaviour of the jumping spiders can be gained.

6.2 Initial Spider optimisation

The first step is to manually define and optimise four spider models to use as initial solutions in the MAP-Elites algorithm. These initial solutions are needed to increase the speed and variety of solutions produced by the MAP-Elites algorithm. The four chosen spiders, *Colonus Hesperus*, *Leptofreya Ambigua*, *Phidippus Mystaceus* and *Naphrys Acerba* are each created in the simulation by manually picking the morphology parameters, the results can be seen in Figure 6.1. Their position on the BD grid can be seen in Figure ??.

Their jump controllers are added by manually picking controller parameters and observing the resulting jump. CMA-ES is used to optimise these controllers. This proved difficult due to issues with CMA-ES. The controllers created by the algorithm did not converge on an optimum, usually the results it produced were worse than the initial, manually defined controller. The algorithm initially used the same fitness function as the last morphology experiment, see 5.3, rather than the new one described in Section 5.2.3. This proved problematic due to the landing being too unstable, resulting in many solutions crashing and returning a negative score. To fix this, several solutions were implemented such as raising the amount of children, changing the fitness function multiple times

6 Exploring biological relations between morphology and behaviour



Figure 6.1: The real spiders and their simulated counterparts. From top to bottom they are the *Colonus Hesperus*, *Leptofreya Ambigua*, *Phidippus Mystaceus* and *Naphrys Acerba*.

and letting the algorithm run longer. However the resulting controllers unfortunately did not improve much, although the amount of parameters and difficulty of the task should be within the capabilities of CMA-ES. In the end most of the spiders used the manually defined controllers that had only been very slightly improved by CMA-ES. These improvements were achieved during the first few generations, where the solution was still closely related to the manual solution, rather than being an optimum that the algorithm converged to.

6 Exploring biological relations between morphology and behaviour

While creating and optimising the different jump controllers, a pattern emerged. Spiders with a heavier abdomen had to push off the ground at a steeper angle. This is noticeable in the parameters that control the perpendicular torque during the jump, from Section 4.3. Higher perpendicular forces mean the spider pushes down more during the jump, increasing the angle between the upwards force and the floor. When the abdomen is larger, these forces are stronger since the centre of mass is further at the back, thus the force needs to have a greater angle to pass through the centre of mass. As an example, for the *Leptofreya Ambigua* the lateral forces are around 0.5, for the *Phidippus Mystaceus*, which has a much larger abdomen, these forces are 1. This is illustrated on Figure 6.2. From a physics perspective, this makes sense as it is best for the pushing force to go through the centre of mass for maximum stability and distance.

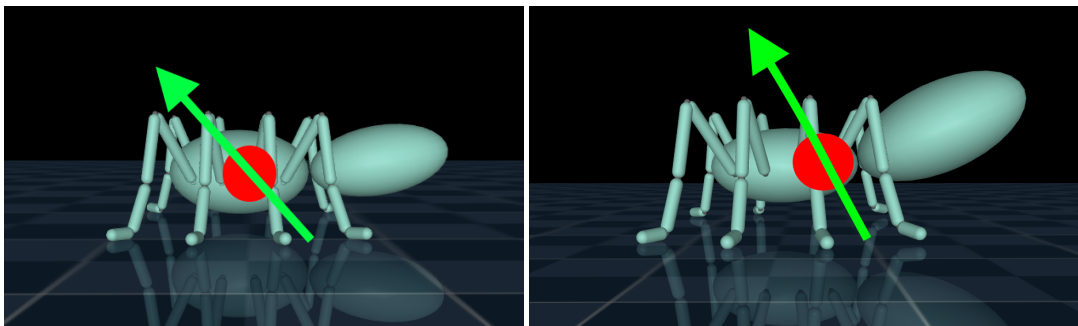


Figure 6.2: The simulated *Leptofreya Ambigua*(left) and *Phidippus Mystaceus*(right), along with an approximation of their COM and the ideal force needed to jump. The spider on the right has a larger abdomen and thus has a centre of mass that is further back and needs to push at a larger angle from the ground.

6.3 Results

Figure 6.3 shows the results of the experiment. Most notable is that the results between 0.4 and 0.5 roundness with low leg length have a significantly higher fitness. Figure 6.4 shows the relation between the abdomen roundness and the fitness. The relation seems to show a parabolic pattern where the area between 0.4 and 0.5 roundness contains the peak. Future biology research could be performed to investigate whether or not spiders with those types of abdomen perform better when jumping.

The hypothesis discussed in Section 6.2 is also explored using the results of this experiment. However the relations between COM position and the parameters that control the angle at which the spider pushes off the ground did not appear in the results. Either the hypothesis is not valid due to another factor influencing the takeoff angle or the analysis performed on the results is not correct. During the analysis the centre of mass and angle of the jump were calculated using the parameters that influence them rather than directly from the simulation. For example, the angle of the jump force is determined as the sum of the lateral forces, even though lean time and strength also influence this force. The centre of mass is also calculated without adding the masses of the legs

6 Exploring biological relations between morphology and behaviour

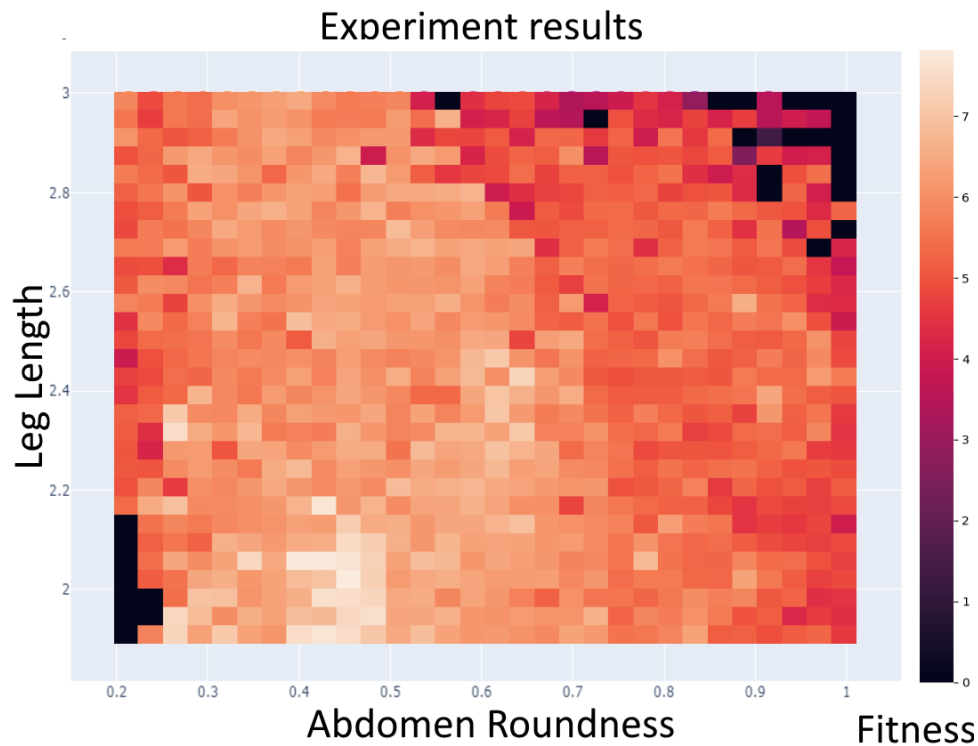


Figure 6.3: The results of the biological takeoff behaviour experiment

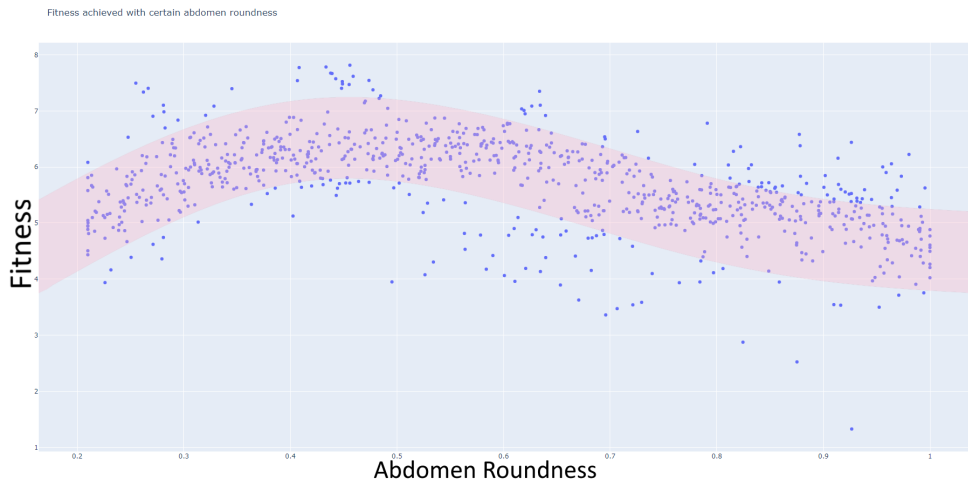


Figure 6.4: A plot showing the relation between the abdomen roundness and the fitness of the simulated spiders, the pink was drawn manually to highlight that these results seem to follow a pattern.

since these were hard to calculate from just the morphological parameters. Future works could focus on adding more precise methods to analyse the results of MAP-Elites experiments by adding more observable parameters to the simulation.

Linking the spiders in the grid back to real spider species proved difficult. While the abdomen roundness and leg

6 Exploring biological relations between morphology and behaviour

length are often different for spider species, they are not the only two aspects that can differ. A spider with a certain roundness and leg length can still take many different shapes, so the optimal solution in the cell for that roundness and length may not correspond directly to a real spider species. Future works could focus on improving the BD with extra features so that every spider morphology maps to a certain cell that can be easily recovered from the grid after quality-diversity algorithms are used.

Future works could focus on improving the analysis methods and exploring the morphology further by using different fitness functions, creating different BDs, changing the morphology parameters, etc. These different experiments can explore different morphological aspects and their relations to behaviour in many different ways, to gain more information about the jumping spider species.

7

Future works

This section discusses the aspects of the work that can be used for future works, either by improving certain parts of the work or by expanding it with new features.

7.1 Improving the parameterised spider morphology

Many different types of jumping spider species exists, all with significantly different morphology and behaviour. The bodies of these spiders are very diverse in their design and not every single type of spider morphology is represented properly due to the necessary abstractions made during the morphology design. This is in part due to the body size and species richness pattern, which poses that the number of species with small body sizes generally far exceeds those of large body sizes [26]. Spiders are small creatures, thus their diversity in species is very large and it is hard to capture that diversity for every single species. To solve this issue, more time needs to be invested in designing parameters to capture every single type of spider morphology with as few abstractions as possible.

The jaws (or *chelicerae*) and *pedipalps*, as seen in Figure 7.1, of the spider could be recreated as separate body parts, rather than considering them as part of the main body. While these parts do not directly contribute to the jump, they still determine the shape of the spider and its aerodynamics. The main body of real jumping spiders is also not an exact ellipsoid, a more precise mesh shape could be used to simulate their bodies more accurately. Another improvement would be to allow the legs to be different from another, some spider species have legs that are specialized, for example scorpion mimic jumping spiders seen in Figure 7.2.

Another important improvement is the inclusion of the guiding drag line that spiders spin during their jump. The drag line is used to facilitate the landing and keep the spider stable. Adding it improves the biological accuracy and allows landing behaviour to be researched from the simulation, which was not possible without the drag line.

Lastly, this work optimised the morphology of spiders only for jump distance. Real spiders will not necessarily need an optimal jumping morphology, but instead their body could be specialized for camouflage, mate attraction, resistance to the elements, running speed, etc. This could hypothetically be achieved by altering the fitness



Figure 7.1: A closer look at the spiders jaws and pedipalps, picture from <https://squirrelsview.blogspot.com>



Figure 7.2: A scorpion mimic jumping spider, possibly *indopadilla sp.* Its front legs are much longer than the others [6]

function and simulation environment to the point where it is as close to real evolution as possible. However achieving such a simulation is an enormous task.

7.2 Improving the Evolutionary Algorithms

Some problems occurred when using evolutionary algorithms during this work, primarily while using CMA-ES. When using CMA-ES to optimise a certain spider morphology the results were often not better than manually defined solutions. Even when using multiple different fitness function variants and increasing the generation size and runtime the system would not improve the design and stopped before reaching an optimum. This could be due to the sensitivity of the parameters, small changes in certain parameters can lead to big changes in the

7 Future works

fitness. For example if a spider jumps just a little bit too late, it might tip over when it otherwise would not have. These parameters might be too volatile for the CMA-ES to use resulting in lacklustre results. To fix this either the CMA-ES has to be forced to run for longer so it will eventually find an optimum or the parameters have to be changed to be less sensitive.

Another issue was how MAP-Elites would usually return the same general morphology for different types of spiders. This could be due to premature convergence, since the controller is very sensitive it might not be able to adapt to small changes in the morphology and therefore is unable to escape the initial morphology easily. It might also be due to the other factors that guide evolution mentioned earlier. The morphology reached may be optimal for jumping, but not for all other aspects of the spiders life. This was mitigated by forcing more diversity through the BD in the last experiments, but more improvements to the BD and fitness function could be made to achieve more diversity in the end results.

The BD could also be changed to ensure that every cell within the grid is more precise and has less variation so that each species of spider maps to one unique cell. The BDs used in this work were too broad and allowed too much variation within a cell. Extra features and dimensions could be added to ensure that every species of spider can be linked back to a cell in the BD grid.

7.3 Creating a physical robot with full mobility

The simulations of the jumping spider without the drag line can be used to help create a physical model of a spider robot that can walk, run and jump. The drag line is not included since it would be difficult to create a robot that is capable of spinning a line with the same features as spider silk. However, many more steps must be taken before this robot can be created. Firstly additional controllers must be added that allow the spider to walk and run, these must then be optimised to work on varied terrains rather than just the flat simulation surface. Ideally all these controllers would be optimised at once along with the morphology to create an optimal spider robot. Afterwards, the body parts, their hinges and actuators have to be mapped to actual physical parts, which would impose extra limitations. After those steps a jumping spider robot with full mobility can be created.

7.4 Expanding the knowledge of biology

The biological research for the topic of this work is limited and can be expanded using the simulation environment. Simulations could be used to find and examine possible hypotheses about the behaviour of spiders and which can be tested afterwards using real spiders to verify or disprove these hypotheses. From those tests the accuracy of the simulation can be evaluated again. If the hypothesis from the simulation was not correct, the information gained from the real experiment can be used to improve the accuracy of the simulation. Afterwards the improved simulation can again be used to find new hypotheses to create a positive feedback loop that improves

7 Future works

the understanding of jumping spider behaviour.

For example, the hypotheses found in Chapter 6, regarding the centre of mass and the jump angle and the abdomen roundness' relation to performance could be possible topics for biological research.

Conclusion & impact

This section highlights the results achieved and assets created during this work.

The environment created during this work can simulate jumping spider morphologies with abstractions as explained in 3.6. The morphology is parameterised to allow many different types of spiders to be simulated, including ones that relate to different biological species of jumping spiders and additionally allowing for hypothetical morphologies not seen in nature to be explored.

A jumping controller was defined that allows the simulated spiders to jump forward. The controller is based on observations of real jumping spiders and is parameterised to allow different jumping techniques to be expressed.

The controller and morphology were co-optimised using evolutionary algorithms. These optimisations can be used to create spider simulations with optimal jumping techniques. Changes in the fitness function were always reflected in the final optimal results of the co-optimised spiders, verifying that the jump controller is able to adapt to new fitnesses functions and able to express varied jumping techniques.

For robotics, the simulation is useful for creating designs of jumping spider robots that cannot create a silk thread. It optimised many different types of possible robots and explored the relations between their morphology and performance. The results of the morphological experiments can be used to check which morphological parameters lead to which performance. For example the results of the joint power experiment in Figure 5.4 show which joint powers and leg lengths are needed to achieve certain jump distances.

For biology, the simulation is able to recreate the jump takeoff of real jumping spiders in order to investigate the effects of certain traits on their jump performance and has identified a possibly new hypothesis regarding jumping spider behaviour. Spiders with larger abdomen and thus centres of mass more at the back will have to push more perpendicularly to the ground compared to spiders with smaller abdomen.

It is also possible to add new controllers to the system in order to give the spider the ability to walk, run and jump to achieve very high mobility in different environments. With these extra controllers the simulations can be used to model physical jumping spider robots that with advanced mobility and stability.

Biologists can verify the previous hypothesis by studying real jumping spiders. That research can then be used to improve the simulation resulting in improved accuracy, which in turn can lead to more discoveries between jumping spider morphology and behaviour. Additionally, a guiding silk thread can be implemented to research the landing aspects of real jumping spiders.

In conclusion the goal of creating a model to investigate jumping spider morphology and its effect on their forward jump was achieved. Future works could focus on adding the silk thread to improve biological accuracy and adding extra controllers in order to get a fully mobile spider robot that can act as a blueprint for a jumping spider robot.

References

- [1] G. V. Lauder, "Robotics as a Comparative Method in Ecology and Evolutionary Biology," *Integrative and Comparative Biology*, vol. 62, no. 3, pp. 721–734, 04 2022. [Online]. Available: <https://doi.org/10.1093/icb/icac016>
- [2] J. Cheng, "Evolutionary optimization: A review and implementation of several algorithms," 2018. [Online]. Available: <https://www.strong.io/blog/evolutionary-optimization>
- [3] "Spider structure," 2018. [Online]. Available: <https://australian.museum/learn/animals/spiders/spider-structure/>
- [4] M. Nabawy, G. Sivalingam, R. Garwood, W. Crowther, and W. Sellers, "Energy and time optimal trajectories in exploratory jumps of the spider *phidippus regius*," *Scientific Reports*, vol. 8, 05 2018.
- [5] C. Tenn, "Webless jumping spiders spin super strong silk," 2021. [Online]. Available: <https://www.the-scientist.com/webless-jumping-spiders-spin-super-strong-silk-69414>
- [6] W. Commons, "File:scorpion mimic jumping spider , possibly indopadilla sp.jpg — wikimedia commons, the free media repository," 2022, [Online; accessed 17-May-2024]. [Online]. Available: https://commons.wikimedia.org/w/index.php?title=File:Scorpion_mimic_Jumping_spider_,_possibly_indopadilla_sp.jpg&oldid=618248739
- [7] M. Çavaş and M. B. Ahmad, "A review on spider robotic system," *International Journal of New Computer Architectures and their Applications (IJNCAA) vol*, vol. 9, pp. 19–24, 2019.
- [8] R. Vidoni and A. Gasparetto, "Efficient force distribution and leg posture for a bio-inspired spider robot," *Robotics and Autonomous Systems*, vol. 59, no. 2, pp. 142–150, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889010001697>
- [9] L. Zentner, *Modelling and Application of the Hydraulic Spider Leg Mechanism*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 451–462. [Online]. Available: https://doi.org/10.1007/978-3-642-33989-9_34
- [10] B. E. E., S. Yoshan, E. D. O., and M. Natasha, "Jump takeoff in a small jumping spider," *Journal of Comparative Physiology A*, 2021. [Online]. Available: <https://doi.org/10.1007/s00359-021-01473-7>
- [11] S. Doncieux, N. Bredeche, J.-B. Mouret, and A. E. G. Eiben, "Evolutionary robotics: What, why, and where to," *Frontiers in Robotics and AI*, vol. 2, 2015. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/frobt.2015.00004>
- [12] G. V. Lauder, "Robotics as a Comparative Method in Ecology and Evolutionary Biology," *Integrative and Comparative Biology*, vol. 62, no. 3, pp. 721–734, 04 2022. [Online]. Available: <https://doi.org/10.1093/icb/icac016>

7 References

- [13] A. K. Schulz, J. S. Reidenberg, J. N. Wu, C. Y. Tang, B. Seleb, J. Mancebo, N. Elgart, and D. L. Hu, "Elephant trunks use an adaptable prehensile grip," *Bioinspiration & Biomimetics*, vol. 18, no. 2, p. 026008, feb 2023. [Online]. Available: <https://dx.doi.org/10.1088/1748-3190/acb477>
- [14] Y. Zhu, L. Chen, Q. Liu, R. Qin, and B. Jin, "Omnidirectional jump of a legged robot based on the behavior mechanism of a jumping spider," *Applied Sciences*, vol. 8, no. 1, 2018. [Online]. Available: <https://www.mdpi.com/2076-3417/8/1/51>
- [15] S. H.-P. Beyer Hans-Georg, "Evolution strategies – a comprehensive introduction," *Natural Computing*, 2002. [Online]. Available: <https://doi.org/10.1023/A:1015059928466>
- [16] N. Hansen, "The cma evolution strategy: A tutorial," 2023.
- [17] K. Chatzilygeroudis, A. Cully, V. Vassiliades, and J.-B. Mouret, "Quality-diversity optimization: a novel branch of stochastic optimization," 2020.
- [18] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," 2015.
- [19] R. Foelix, *Biology of Spiders*. OUP USA/Georg Thieme Verlag, 1996. [Online]. Available: <https://books.google.be/books?id=XUgyqxNKhyAC>
- [20] C. Göttler, K. Elflein, R. Siegwart, and M. Sitti, "Spider origami: Folding principle of jumping spider leg joints for bioinspired fluidic actuators," *Advanced Science*, vol. 8, no. 5, p. 2003890, 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.202003890>
- [21] S. Landkammer, F. Winter, D. Schneider, and R. Hornfeck, "Biomimetic spider leg joints: A review from biomechanical research to compliant robotic actuators," *Robotics*, vol. 5, no. 3, 2016. [Online]. Available: <https://www.mdpi.com/2218-6581/5/3/15>
- [22] Dick Walton Natural History Services website. [Online]. Available: <https://www.rkwalton.com/zuniga-sp/>
- [23] Online encyclopedia for different types of bugs. [Online]. Available: <https://bugguide.net/node/view/15740>
- [24] D. A. Parry and R. H. J. Brown, "The Jumping Mechanism of Salticid Spiders," *Journal of Experimental Biology*, vol. 36, no. 4, pp. 654–664, 12 1959. [Online]. Available: <https://doi.org/10.1242/jeb.36.4.654>
- [25] L. Eguiarte-Morett and W. Aguilar, "Premature convergence in morphology and control co-evolution: a study," *Adaptive Behavior*, vol. 32, no. 2, pp. 137–165, 2024. [Online]. Available: <https://doi.org/10.1177/10597123231198497>
- [26] G. E. Hutchinson and R. H. MacArthur, "A theoretical ecological model of size distributions among species of animals," *The American Naturalist*, vol. 93, no. 869, pp. 117–125, 1959. [Online]. Available: <https://doi.org/10.1086/282063>