Self-Reconfigurable Robots For Space Exploration

Effect of compliance in modular robots structures on locomotion

Vardi Adi

Semester project

Professor: Prof. Auke Jan Ijspeert

Supervisors: Stéphane Bonardi, Simon Hauser, Mehmet Mutlu, Massimo Vespignani

Micro-Engineering
École Polytechnique Fédérale de Lausanne
Switzerland
January 2016

Abstract

This paper presents a study on the effect of compliance in a modular robotic structure. The structure used is a quadruped robot where a compliant element is inserted between every two adjacent modules. The compliance of each element is varied individually resulting in a distribution distribution of compliance over the structure. The structure was built with the Bioloids kit and modeled in simulation using Webots, a physics simulation software developed by Cyberbotics Ltd. [16]. The effects were studied on two types of rough terrains in order to represent real environments, such as the Mars terrain.

As a first step, a Particle Swarm Optimization was used to optimize the gait of the robot in order to achieve fast, straight and stable gait on rough terrain. The robustness of the robot to external disturbances was tested and the robot's inherent stability was showed.

Systematic searches on the compliance values were used to test their effects. Later, a Particle Swarm Optimization was used to retest and validate the searches. Two main patterns were observed resulting in two hypotheses concerning the impact of compliance on the quadruped's locomotion. Hypothesis 1 claims that the inner compliant elements of the structure should be stiff. This hypothesis was supported by all the datasets. Hypothesis 2 claims that the outer compliant elements of the structure should be softer. This pattern was observed in a large part of the results, but is not very decisive and is not present in all the datasets. A lower limit for compliance was also observed.

The big impact of compliant elements on the locomotion's stability and speed was shown. This effect requires more research to be definitively established, using different structures and gaits.

Key words: Modular robots, Locomotion, Compliance.

Acknowledgments

I would like to acknowledge all the people who supported me during the work on this project.

First, to Prof. Auke Jan Ijspeert who welcomed me to the lab for my first master project.

To my supervisors Stéphane Bonardi, Simon Hauser, Mehmet Mutlu and Massimo Vespignani for their massive help during the entire project.

I thank Florin Dzeladini for his help with the optimization the code.

Finally, to all the members of the BioRob laboratory for your help in various subjects and questions.

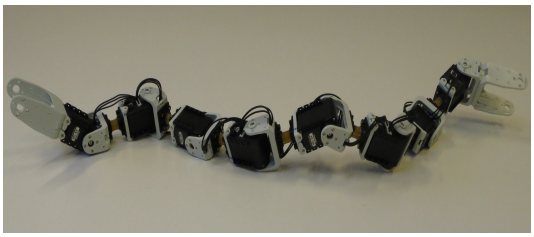
Contents

1 Introduction

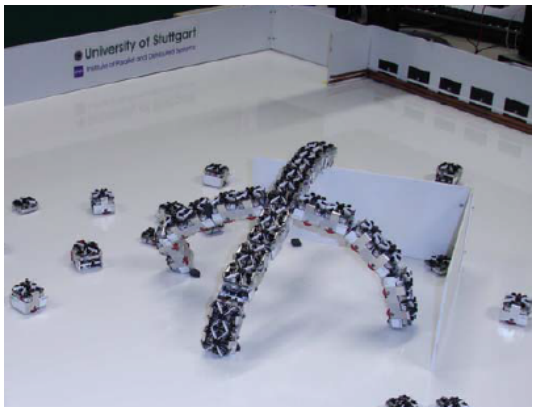
Since the creation of the first mobile robot, locomotion has been a key problem and an active area of research. Being the main feature of mobile robots, locomotion confronts us with many challenges and many aspects to consider. Exploring this field, one can concentrate on making locomotion more efficient in various aspects such as speed, stability, energy consumption and more. One could also work on improving the robustness of locomotion to external disturbance, to obstacles or to rough terrains.
Another field in robotics is modular robotics, as reviewed by Ahmadzadeh et al. [3]. Such robots are composed of multiple parts, or modules. Reconfigurable modular robots are even capable of changing their morphology by connecting and disconnecting their modules. This ability can be extremely beneficial in many applications since it allows the robot to handle a wide variety of tasks thanks to its flexibility and adaptability. However, modular robots present a big challenge; apart from the "usual" challenges of fixed-body robotics, modular robots add a second layer of complexity, which manifests itself in several aspects, such as mechanical design and control. For example, modular robots need to find the best configuration for a given task or the best way to move all the modules in collaboration. In addition, since each module might have to support other modules, the robots' structures are limited by the actuation of the modules. Several examples of modular robots can be seen in fig. 2.
When combining modular robotics and locomotion, we get a very interesting field of research. The locomotion of modular robots has been studied extensively over the last few years, but many of the challenges are yet to be overcome.
The aim of this project was to explore a relatively new trend in modular robots' locomotion. That is, using strategically placed compliant elements inside the robot's structure in order to improve the locomotion. This has been already done in several studies. Vespignani et al. studied the effect of compliance on a modular snake robot. In [7] they studied the effect of compliance on a snake robot rolling on different horizontal pipes. They showed that using compliant elements the snake can adapt to the pipe size. This results in a big improvement in the power consumption. In [9] and [8] the existence of an optimal stiffness value was tested on different types of rough terrains, looking to maximize speed and minimize energy consumption. Using sidewinding gait, comparable results were obtained for compliant and stiff structures. Using rolling gait, the gait was improved when using a rigid structure. Note that all these studies have used a homogeneous structure where all the compliant elements have the same stiffness. In another work [6], the effect of compliance was studied using Roombots - self reconfigurable modular robots. Using different configurations they showed that compliance can improve the speed of the gait and that different compliance structures can result in different locomotion strategies.
All those studies have shown very interesting results. Even though the effect of compliance has not been proven to be absolutely beneficial in all cases, it had been showed to improve certain aspects of the locomotion.

An important question to ask is how to characterize good locomotion. This question can have many answers, depending on the application, task, structure and other characteristics of the robot. One aspect for improvement can be high speed. We can also consider low energy consumption, stability, maneuverability and more.
For this project, we have chosen to concentrate on the stability aspect of locomotion. The term stability is used to describe how much the robot tilts during the locomotion and its sensitivity to external disturbances. While the speed is still optimized, a bigger weight is given to good stability, even if the speed is compromised. In addition, we have decided to choose a single robot morphology - a quadruped with 8 degrees of freedom, two in each leg. This bio-inspired morphology can be very interesting for many applications (such as exploration, rescue and transportation).
This project is mainly intended as a first research for the use of modular robots for space applications. In such environment, where spare parts are scarce, the flexibility and adaptability of modular robots

(a) Roombots.



(b) Lola-OP.



(c) Symbrion.

(d) Smores.

Figure 2: Examples of modular robots. Images from [15, 9, 12, 14]

are highly appreciated. Such robots could be used to assist a big rover in exploring difficult terrains such as crevices, cliffs and caves. We have chosen to look into the Mars environment. Mars poses many engineering challenges due to very rough terrains and extremely fine dust. Even though the final results of this project are not yet ready to be deployed in the next mission, I have used this goal as a motivation and a source of inspiration for this project.

2 Definition of the project

There are many directions to explore in the subject of compliance elements and locomotion. Considering our ambition, we decided to focus on one part of it, mainly the the effect of compliance on the stability of a quadruped robot.
For the gait optimization, we have considered many possible metrics to be optimized, such as speed, power consumption and robustness.
Power consumption estimation using simulation is highly unreliable, thus so we have decided to exclude it.
As we are mainly interested in the stability of the locomotion, the speed of the gait is of less importance. Higher speed is considered as better gait, but with a lower weight than the stability.
We have defined a number of steps to match the necessary milestones of the study.

Step 1 - Modelling in simulation
The first step is to model the quadruped in Webots, including the compliant elements. I also need to implement a controller for the movement.

Step 2 - Optimization of gait parameters
This part involves optimizing the controller parameters to achieve a betgter gait on a slope. Tough the compliant elements are inside the robot's structure, the robot is entirely stiff in this step. The running time of each repetition is 30 seconds. Each repetition will result in a value of a fitness function that quantifies the characteristics of the gait to be optimized. This is done by maximizing the fitness value.

Step 3 - Robustness to external disturbances
Moving on flat terrain, with no obstacles, the robot will have to resist external forces. In 100 repetitions, three forces are applied to the center of the robot in random directions and at random moments in time. The evaluation will use the same fitness function, since it includes a measurement of the robot's stability. The results for this step can be later compared to the same test, this time using the best compliant elements configurations found. As later shown in chapter 6, I found the gait to be inherently stable, so running this test again with compliant elements would not give very useful results. Thus, I decided to not run it again but to concentrate on the compliance introduced in the step 4.

Step 4 - Rough terrain with compliant elements
In this step the compliant elements are introduced. The structure of the robot doesn't change, but the stiffness values of each element are varied between three values (low, medium and high). A systematic search is used to find the compliant configuration which gives the highest performance (measured by the fitness function). Note that the stiffness can very well change between different elements in the structure.

3 Preliminaries

3.1 Central Pattern Generators (CPG)

As in nature, there are many approaches to control robotic locomotion. In this work, Central Pattern Generators (CPG) were used. As described in [11] and [2], CPG networks are used to control a robot containing sub-structures, taking into account the symmetries of the structure. This results in Bio-Inspired periodic gaits. A CPG can be expressed as a network of non-linear coupled oscillators with as many oscillators as degrees of freedom. The main advantages of CPGs are the small number of control parameters needed (thus saving optimization time), their ability to generate many different gaits and their ability to keep the gait smooth even in the face of disturbances.
Each CPG, or each motor, is formed by the following equations:

$$\dot{\phi}_i = 2\pi f + \sum_j \omega_{ij} r_j sin(\phi_j - \phi_i - \psi_{ij}) \tag{1}$$

$$\dot{r}_i = a_i(R_i - r_i) \tag{2}$$

$$\dot{\theta}_i = r_i * sin(\theta_i) + X_i \tag{3}$$

i and j are the index of the motors, θ is the oscillator output, r_i is the amplitude of the oscillator and ϕ_i is its phase.
A fixed frequency for all the motors is (0.5Hz) and all the coupling weights (w_{ij}) are set to 2. The CPG have a number of open parameters for each motor: the desired amplitude (R_i), the offset (X_i) and the phase between every two coupled oscillators (ψ_{ij}). As will be described later, symmetry and only nearest-neighbor coupling can reduce the number of open parameters.

The CPGs were implemented using the Codyn framework [13]. Using Codyn, the parameters are defined in a separate .cdn file, called the network file. This file is read by the robot's controller which then uses them for the motors.

3.2 Particle Swarm Optimization (PSO)

PSO is a population based optimization algorithm. It is used to find the extrema combinations of parameters in a search space. In each iteration, the particles are evaluated by simulation and a value corresponding to a fitness function is returned (see more details in chapter 5). Using their own evaluations and those of their neighbors, the particles are moved to new locations in the search space. After a certain number of iterations, the final solution representing the best combination of variables the algorithm has found. More details can be found in [10].

3.3 Optimization Framework

In order to implement the PSO, and later the systematic searches, an optimization framework was used. A single .xml file, called the job file, is created. This file includes all the required data for the optimization. Among its fields are the number of particles and iterations, the location of the Webots simulation, the open parameters and the fitness function.
When an optimization is run, the job file is read before the network file, if certain parameters are defined in both, the job file takes precedence. Thus, the optimization framework can change the values of the parameters defined in the network and to optimize them.

4 Step 1 - Modelling in simulation

4.1 Robot Modelling

The first task I had to accomplish was to model the robot in simulation. As already stated, I am using Webots.
From previous works done in the lab, I had models of a quadruped without compliant elements and of a snake robot with compliant elements. The existing quadruped model was built using Connector nodes. These nodes connect different modules like two magnets snapping together.

The robot could be modelled using different approaches. The two main ones are:

- **Serial approach:** The entire robot is modelled as a single entity in the simulation. All the motors and joints are included inside a single Robot node. Only one controller is needed (plus a Supervisor). This approach yields more stable results and is computationally cheaper.
- **Modular approach:** Using the Connector nodes. The robot is composed of several modules. Each module is one servo motor (a hinge joint) and one compliant element (a ball joint). The modules snap together in the beginning of each run. Each module is controlled by a controller and there is an additional supervisor.

As the robot's structure is not to be changed during the entire project, we don't need it to be composed of modules. Considering the benefits of the serial model, I have decided to rebuild the robot without using the connector nodes. This was not easy as the structure is fairly complex and, being modelled as a single entity, far more complicated than the modular approach.

For the compliant elements, as done before with the LOLA-OP robot [7], ball joints with non-zero spring constants are used. For simplicity, I have not included a solid to represent the actual compliant elements, but their function is successfully implemented by the ball joints. I have added bounding objects between the modules to prevent impossible situations where an object can pass between the modules.

I have also started working on adapting the controller to work with the quadruped robot. I had a controller designed for a tripod, but since the quadruped has a different morphology and is modelled differently, the controller had to be adapted.

After successfully modeling the robot using this approach I have discovered a bug in the simulation. When the spring constant of certain ball joints is set above a certain value, the model starts vibrating and falls apart. After consulting Cyberbotics this problem turned out to be a bug in Webots. For some specific rotations, the ball joints are unstable when introducing a high spring constant. This can be seen in fig. 3.

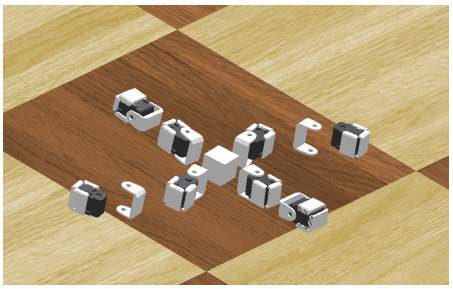


Figure 3: Failure due to a bug in Webots with the serial model - Two joints are vibrating and falling apart.

This bug has forced me to reconsider my modelling approach.
Thus, I have remodeled the robot's structure using the modular approach.

Using the new structure, adapting the controller was far easier. I have also adapted the network file to work with the new structure. The final model can be seen in fig. 4.

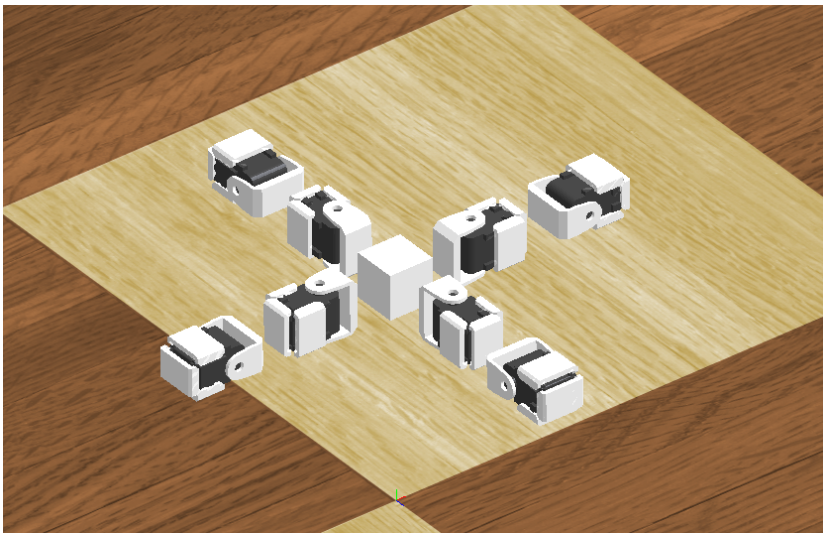


Figure 4: The final modular model.

4.2 Terrain Modelling

After modelling the robot in Webots, I had to model its environment. In this project, three different terrains are used.

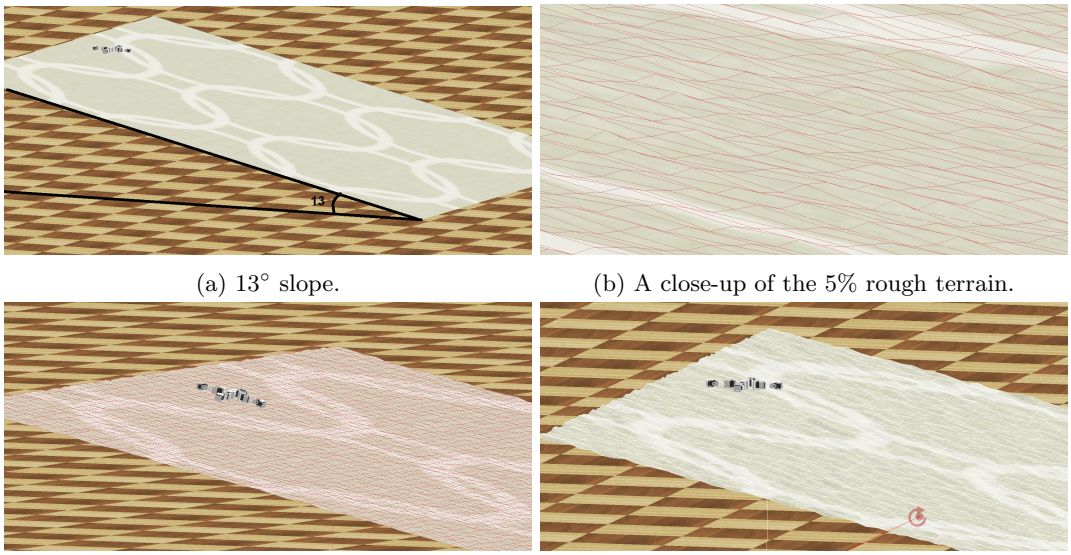
The first is a simple flat ground. This terrain is used to test the robot's robustness to external disturbances in the step 3. This terrain is simply the flat arena object of Webots.

The second terrain modelled is the flat slope. This is a simple smooth slope of a certain angle. This terrain is necessary due to another bug in Webots - some directions in Webots incur less friction. If the robot is moving on flat ground, it is impossible to know if its chosen direction is due to the controller or to the bug. Since the controller is open loop, it is not possible to command the robot to move in a certain direction. Thus, a slope is used in order to favour a direction of movement and to prevent this problem.
The slope is modelled using an ElevationGrid node. This node defines an array of points in the horizontal plane and assigns a height for each. Only 4 points are needed for the smooth slope. Two are touching the ground so their heights are zero and two define the highest end of the slope. Their heights can be calculated with simple trigonometry as:

$$H = horizontalLength * tan(angle) \tag{4}$$

I have created four different slopes with different angles: 5°, 13°, 20° and 25°. For next steps I have chosen to work with the 13° slope as it seems suitable for the robot's size and capabilities. The robot is situated at the beginning of the slope (the highest point), but a bit into the slope so it won't fall over the edge.

The third environment is the rough terrain. This environment will be used for the rest of the project. Again, a slope to impose a predetermined direction of movement is used. Roughness is added to the slope to create a rough terrain. The roughness is created by changing the heights of many points on the slope by random values. To create the rough terrain, I have changed the ElevationGrid to include many more points. The total slope size is 6x6 meters and the new grid spacing is 60 mm. The spacing is chosen to be less than the length of one module (68 mm without the compliant element), so at any point of time, two different modules will not be at the same height. The height of each point is the sum of its smooth slope height (using simple trigonometry as before) and a random number between



(a) 13° slope.

(b) A close-up of the 5% rough terrain.

(c) 5% rough terrain.

(d) 10% rough terrain.

Figure 5: Terrains used in this project.

0 and *max_roughness*. The *max_roughness* is defined as a function of the robot's leg length (sum of 2 modules lengths). I have created two different rough terrains, one where *max_roughness* is 5% of the leg length and the second with 10%. The final terrains used in this project are presented in fig. 5.

5 Step 2 - Gait optimization

After finishing modelling the robot on Webots and adjusting the network file and controller, the first study step can begin. In this step, the aim is to optimize the gait parameters in order to achieve the highest straight line speed possible.

5.1 Gait parameters

The robot's motors are controlled by a CPG network. In order to achieve a gait, and to optimize it, the CPG parameters must be optimized. This optimization depends on the robot's structure.

Using the CPG equations and nearest-neighbor coupling (see chapter 3.1), each motor has three open variables. For the quadruped morphology, we can reduce the number of open parameters using symmetry. For the amplitude and the phase shift, I used similarities inside each leg. Thus, the amplitude and the phase of the pair of motors belonging to the same leg are the same.
For the coupling, I decided to use neighbor coupling. In each leg, the two motors are coupled. In addition, the first motor in each leg is coupled to the other first motors. This can be seen in fig. 6. There is also symmetry between the front legs and between the hind legs.
These considerations are expressed in both the network file and the job file (see subsection 3.3). They result in reducing the number of open parameters from 24 to 13 (if each motor has three open parameters.)

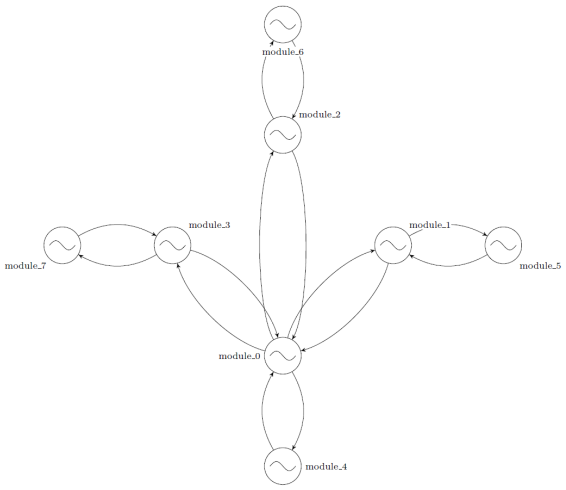


Figure 6: Illustration of the CPG neighbor coupling.

5.2 The optimization (PSO)

PSO is used for the gait optimization. Each particle encodes the values of all the parameters to be optimized, in this case the CPG parameters. A fitness function is used for the particles evaluations (see later in this chapter). For all the PSO in this work, 50 particles and 200 iterations are used. In addition, each particle is evaluated three times for each iteration, in order to reject problems due to numerical instability.

5.2.1 The job file

In order to run the PSO, the job file fields had to be completed. The open parameters are set as explained before. For the fitness function, it is composed of three measurements: the advance along the direction of movement, the deviation from this direction and the instability of the robot. Those three and the final value of the fitness function are defined as the fitness variables. The fitness function value is defined as the expression to be maximized.
Next step is to calculate the fitness function, which is done in the robot's controller.

5.2.2 The fitness function

The fitness function is calculated in the supervisor controller and then sent back to the optimization framework. Calculating the advance and deviation is quite simple. The axis of movement is defined as the z-axis (Note: In Webots the vertical axis is the y-axis, so z is horizontal). This is because the slopes are oriented according to this axis. Thus, the advance of the robots is simply the difference between its original position and its final position according to the z-axis. The robot's position is found by calculating its center of gravity. Similarly, the deviation is the distance the robot has traveled in the x-axis (the other horizontal axis).The stability evaluation isn't as straight forward.

The initial fitness function had the following form:

$$F = \frac{D/(1+d)}{e^R} \tag{5}$$

$$R = \sum_{timestep} |g_y - v_y| \tag{6}$$

where D is the advance along the axis of movement, d the horizontal deviation and R represents the stability (or vertical deviation). In this equation, R is calculated as the sum over the entire run of the deviation of the robot's vertical axis from the gravity axis. But, this expression is hard to find since it requires finding the robot's vertical deviation. For this, we need to introduce a new sensor to the robot (in this case to the middle cube module). There are three options for this sensor:

- Accelerometer: This is the obvious choice for this job. The accelerometer measures the accelerations in the different axis. When the robot is not moving, the only acceleration is the gravity. By knowing the decomposition of the gravity on the different axis, we can find the robot's tilt. However, the accelerometer suffers from a big drawback. If the robot is moving, other accelerations are added to the gravity, so we cannot extract the real decomposition.
- Gyroscope: The gyroscope measures the rotation speed around its axis. The gyroscope doesn't suffer from the accelerometer drawback, but introduces another issue. Since it only measures the rotation speeds, we cannot actually tell the tilt of the robot (since the speed can also be zero while the robot is on its side).
- Inertial measurement unit (IMU): This unit combines both gyroscope and accelerometer to give directly the roll-pitch-yaw angles of the robot. At first glance, this is perfect, but I soon realized that the robot tilt is a composition of both roll and pitch angles. Also, in order to actually find the robot's vertical axis, we would need to convert from the angles to vector. While feasible, this isn't the simple mathematical problem and might impact the robot real-time performance. The IMU also adds a complexity to the model.

I have decided that the simplest option will be to use the gyroscope to measure the robot's speed around the two horizontal axis instead of its tilt. The resulting R is zero if the robot is perfectly stable and higher if the robot falls over rapidly. The new R value is then given by:

$$R = \sum_{timestep} (|\omega_x| + |\omega_z|) \tag{7}$$

After trying this approach, I have realized that the gyroscope's capability of only measuring the rotation speeds is not enough. This value of the R function doesn't penalize the gait if the robot is constantly tilted. Thus, I have decided to change the fitness function once again. Using the IMU, the supervisor controller receives the roll-pitch-yaw angles. If the roll and pitch angles are zero, the robot is perfectly vertical. Thus, my new R function is calculated by:

$$R = \sum_{timestep} (|roll| + |pitch|) \tag{8}$$

Since the roll and pitch angles are never really zero, R grows relatively fast. To prevent the exponential in the denominator of the fitness function from growing to a very high value the fitness function is changed to:

$$F = \frac{D/(1+d)}{1+R} \tag{9}$$

I have also tried running the PSO with a R function given by:

$$R = \sum_{timestep} (|roll| + |pitch| + |yaw|) \tag{10}$$

This function penalizes the robot from turning around its vertical axis.

Since choosing a fitness function can have a big effect on the locomotion, I decided to try one more expression, one that only cares about the gait. I have eliminated the R function from the value of F. So,

$$F = D/(1+d) \tag{11}$$

The R function values are still logged since it shows us the stability of the robot.

The Flipped variable

I have also added one last variable to the fitness function - the Flipped variable. This is a Boolean that signifies if the robot was flipped or not. The robot is marked as flipped if either the roll or pitch are bigger than $\pi/2$.

5.2.3 Implementation

The IMU is added to the middle cube. To send the IMU measurements to the supervisor controller I have created a new controller for the cube. Its only role is to send a message containing those measurements. This message is then received by the supervisor.In the supervisor controller, every time step, R is increased by the sum of the roll and pitch (in absolute values). Since the robot is starting by free falling to the ground, it might not be very stable in the first seconds of each run. The CPG also needs a certain time to converge to its stable state. Thus, R is not increased in the first three seconds of each run.

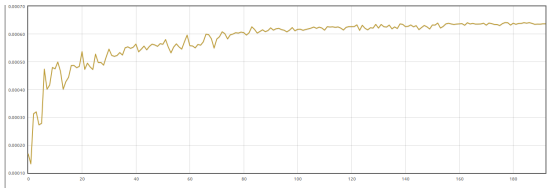


Figure 7: Example of the evolution of the fitness value during the optimization by iteration. The units are rad^{-1}.

After everything had been implemented into the controllers and the supporting files, I ran the optimization framework to find the best gait parameters.

5.3 Results

After running the PSO once on a smooth slope, I have decided to run it again, this time on the rough slope. I ran the PSO three times on the rough slope, for both expressions of the R function (with and without the yaw) and without the R function. I compared the three resulting movements (videos can be found in the project CD). When the yaw is not penalized the robot turns more around its axis. However, this movement doesn't affect its stability and results in a much faster gait (the stability

and horizontal deviation are practically the same). Thus I have concluded that the gait that doesn't penalize the yaw is better.
Concerning the fitness function without the stability term, the resulting gait is showing easily the effect. The movement is much more dynamic - The robot lifts higher its legs and the ground clearance is higher. The gait is also better (advance and horizontal deviation). The only problem is the stability which is roughly two times lower - the R function is two times higher. (It can be seen on the video that the robot tilts much more).

I have decided to keep both functions and run the next step (robustness to external disturbances on both). I plugged the resulting gait parameters into the network file. The next steps of the project will use those parameters to create the robot's movement.
Fig. 8 roughly shows the position of the robot when walking. See videos for a better illustration.

Figure 8: The quadruped walking on a flat slope.

6 Step 3 - Robustness to external disturbances

Now that we have the gait parameters for the robot and it is capable of walking over both smooth and rough terrain, we need to establish its robustness to external disturbances. The results of this test can be used as a control group. Later, after optimizing the stiffness of the compliant elements a similar robustness test could be concluded. Comparing the final results with the results obtained in this step will tell us if the compliant elements have improved the robustness of the robot and by how much. This step is also used to decide on a final gait parameters and fitness function, as I still have two possibilities from last step.
The robustness test is carried out as followed: Moving on flat terrain, three forces are applied on the center of the robot at three random times. The directions of the forces are completely random but their norms are set to a specific value. The forces are applied for a fixed duration. For each different value of norm, 100 repetitions are run. The fitness value (and all the fitness variables) are logged. Those values are the results of this test and will be later compared to the results using compliant elements.

6.1 Numerical values

I had to find appropriate values for the forces' norms and the duration. The robot weighs approximately 900 grams. Thus, a force of 10N could lift it off the ground. Even a lesser force, if applied in a certain direction, could drag the robot on the floor. Thus, I have decided to limit the forces to 3, 4 and 5N and the duration to 300ms. I have experimentally checked these values by running the simulation multiple times and visualizing the results.
In order to let the robot some time to start (It also has to fall to the ground at the beginning of each run) the forces can only be applied between 5 to 25 seconds (The entire run is 30 seconds).

6.2 Implementation

The forces generation is implemented using a Physics Plug-in. This plug-in is a .cpp file used by Webots to affect the physics of the world. During the initialization part, the three timings are generated randomly between 5 to 25 seconds. The three forces are also generated and their components are normalized so the forces will have a norm equal to the desired norm (defined as a constant). During the run, if the current time is between one of the generated times and 300ms later, the force is applied.

In order to run the simulation 100 times, I have used the already existing optimization framework. I have adapted the job file used for the PSO from last step. This time, the PSO is only used to run the repetitions as we don't actually want to maximize anything or to change any parameters. To do so, I have changed the number of particles to 1 and number of total iterations to 100. In order to prevent the parameters from changing, I have set the boundaries to be extremely close and practically equal to the results from last step. This way, even when the PSO tries to optimize the parameters, they cannot really change from their set values.

6.3 Results

The results of this step are used for two goals. One, to decide on final gait parameters and two, as a control group. They will serve for a comparison with the results of the same test, this time applied on the compliant robot.

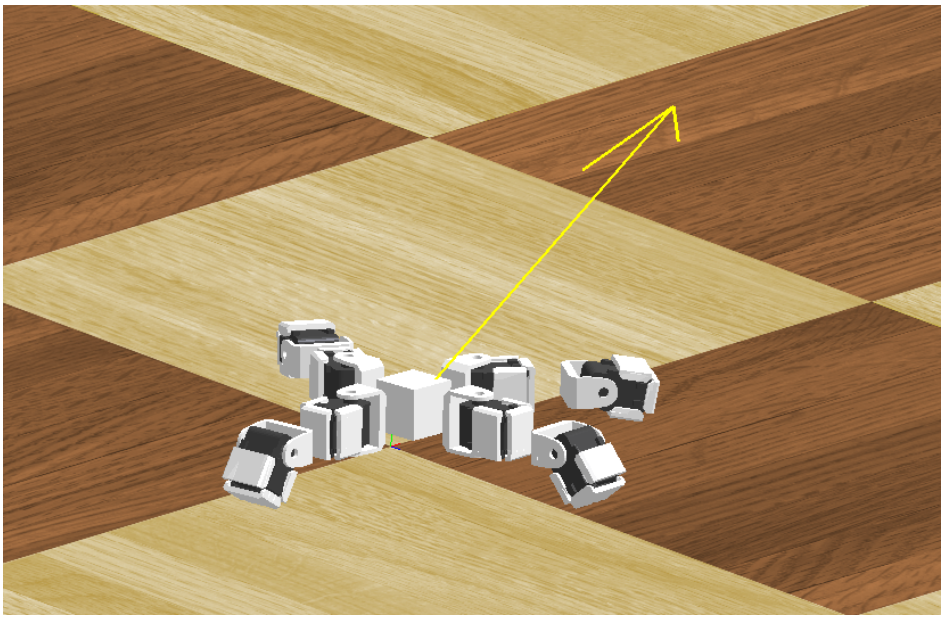


Figure 9: A force (yellow arrow) applied to the quadruped.

First, I have compared the results for both gait parameters. Using the gait obtained with no stability function, the stability of the robot suffers. For the same norm, the stability function values are higher and have a bigger variance using this gait (higher values mean the robot is tilting more). Thus, I have decided to let go of this gait. This is because in this study I am mainly interested in the stability of the robot. Not having the most dynamic and fast gait is not as important as having a stable robot. So, the final fitness function is given in equations 9 and 8. The fitness function unit is rad^{-1}, and will be omitted from now on. The final gait parameters are:

Parameter \ Element	0	1	2	3	4	5	6	7
Amplitude R	0.781	0.781	0.181	0.181	0.426	0.426	0.326	0.326
Offset X	-0.780	-0.780	0.0036	0.0036	0.724	0.724	0.233	0.233

Phase between coupled oscillators:
$\psi_{0,4} = 1.445$, $\psi_{1,5} = psi_{04}$, $\psi_{2,6} = -0.244$, $\psi_{3,7} = psi_{26}$, $\psi_{0,1} = 0.622$, $\psi_{0,2} = -1.339$, $\psi_{0,3} = 0.713$

Using those gait parameters, the resulting gait is very stable (see also the gait videos, in the Project CD). For forces of norms of 3,4 and 5 N, the robot is not greatly disturbed. This is probably due to the large support polygon of the robot and to the chosen gait, which has very low ground clearance. For this reason, we have decided to not retest the robustness to external disturbance with the compliant elements. Instead, I have chosen to do a deeper analysis on the influence of compliance, as will be reported in the next chapters.

7 Step 4 - Rough terrain with compliant elements

After establishing a control group for the stability, it is time to introduce the compliant elements to the structure. As noted before, the model used so far includes those elements, but their stiffness was set to a very high value (1000 Nm/rad) so they are practically rigid. The structure of the quadruped includes eight compliant elements. In order to be as exhaustive as possible, I wanted to check every possible configuration of stiffness value. This will result in a heterogeneous structure, where every element's stiffness is set individually in order to achieve the best locomotion of the entire robot. Using a systematic search, simulations are run for each possible configuration and the results are logged. The gait used for this step is the gait obtained in the previous steps. As before, the results are the values of the fitness function, since it includes all the locomotion characteristics to optimize.
In order to reconfirm the systematic search results, I have later run a PSO on each terrain. The PSO is done exactly as the systematic search, but the stiffness variables can take continuous values. In addition, as the PSO evolves, the stiffness values are adjusted to get the highest fitness value.

7.1 Implementation

The optimization framework previously used for the PSO also includes a systematic search option. In a systematic search each parameter can only take discrete values. Thus, apart from defining maximum and minimum values for each parameter, the number of steps between is also defined. Instead of modifying the gait parameters, the search modifies the stiffness of the compliant elements. Using Webots, this is done by changing the "springConstant" field of the Ball Joints. Each configuration is repeated three times to avoid instability problems.
Similarly to the PSO, the supervisor controller reads the job file and changes the stiffness values accordingly. Since the job file doesn't modify the gait parameters, the supervisor reads them from a network file (see subsection 3.3).

7.2 Numerical values

With eight compliant elements, increasing the number of possible stiffness values exponentially increases the number of possible configurations. With only three stiffness values, there are already $3^8 = 6'561$ configurations. With three repetitions per configuration, this results in 19'683 simulations. Four possible values result in 65'536 configurations and 196'608 simulations. Thus, it is impossible to run a systematic search with too many values.

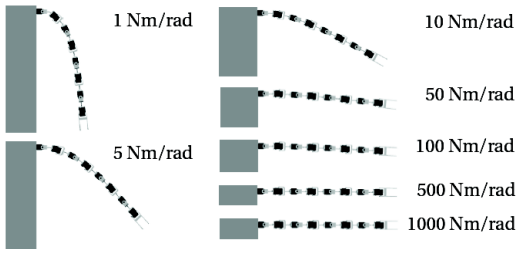


Figure 10: Effect of compliance on a Lola-OP snake robot with 8 DOF. The bending is only due to the compliant elements. Image adapted from [9].

Since a very broad search is impossible, I have run multiple narrower searches. Though this is not as exhaustive, since not all possible configurations are tested, this still shows the effect of many different stiffness values. For the systematic searches, all the values are powers of ten due to the implementation (changed for the PSO).
For each rough terrain, 5% and 10% roughness, I have run three systematic searches with the following possible values: (10, 100, 1000) , (1, 10, 100) and (1, 3.162, 10). For the 5% rough terrain, I have also run a search with four values (1, 2.154, 4.642, 10). For the final PSO, the compliant elements can take values between 1 to 50. All those values are in Nm/rad.

8 Results and Analysis

N.B: *In order to facilitate the text, the word "statistics" is used to represent the box plot, mean and standard deviation of a dataset. In all the figures containing such data, the mean and standard deviation are plotted in green and the other colors belong to the box plot.*

Running the systematic search results in a database containing the fitness variables values and stiffness parameters by iteration. In each iteration the stiffness parameters change, so each iteration represents a different configuration of compliant elements.
Each dataset was subjected to the same analysis. First, in order to visualize the fitness variation, it is plotted by iteration. Then, the data is clustered into two parts using a fixed threshold. This threshold can be chosen by the user, but for this analysis I have left it fixed as the median of the fitness values. From each cluster, 15 cases are chosen. Those cases correspond to the best five, medium five and worst five configurations. The stiffness of those cases is then printed. The statistics of the stiffness of each compliant element for each set of five cases are also plotted. Since each statistics is only based on five cases, this is not representative enough and I have not based the analysis on it.

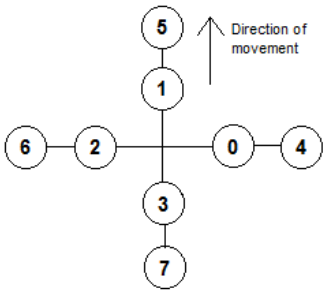


Figure 11: The structure and numbering of the compliant elements. The numbering is the same for the modules, motors and compliant elements.

The next step of the analysis is to divide each cluster into three subclusters of equal size. Thus, the dataset is divided to a total of six subclusters. For each of these subclusters, the statistics of each compliant element are plotted.
At this step I have started noticing an interesting phenomenon. According to the data, it seems that the best results are obtained with configurations where the inner compliant elements are stiff (elements 0 to 3) and the outer are soft (elements 4 to 7). The only leg that doesn't seem to conform to this behavior is the front leg. This pattern leads me to formulate two hypotheses:

1. The inner compliant elements of the structure should have higher stiffness.
2. The outer compliant elements of the structure should have lower stiffness.
Both hypotheses exclude the front leg.

In order to investigate theses hypotheses further I have grouped together the inner elements and the outer elements, excluding the front leg. This is done for each subcluster. Then the statistics of each group are plotted.
The next step was to explore a bit more the inner-outer phenomenon. In order to do that, the data is re-clustered differently. This time the clustering is done by comparing the inner and outer elements. Excluding the front leg, if the average stiffness value of the inner elements is bigger than the average of the outer elements, the configuration belongs to the first cluster. Otherwise, it belongs to the second cluster. Combinations where the the two averages are equals were excluded in order to have clusters of the same size.
The last step is to analyze these new clusters. For this end an ANOVA test is run on the two clusters. In order to use an ANOVA, the dataset must verify a couple of assumptions:

- Independence of observations - Since the observations are different simulations we can assume their Independence.
- Residuals normality - The histograms of the residuals are plotted in order to verify their normality.
- Homoscedasticity - This can be verified by running a Levene's test on the dataset. The results of this test show the assumption is weakly verified.

After these verifications an ANOVA test is run on the two clusters to check if they are significantly different. The significance level (or p-value) used is 5%.

To recall, for each rough terrain, 5% and 10%, I have run three systematic searches with (10, 100, 1000) , (1, 10, 100) and (1, 3.162, 10) Nm/rad as possible values and a PSO with stiffness between 1 to 50 Nm/rad. After some thought, I concluded that there is not much difference between 100 and 1000 Nm/rad, since both values are practically completely rigid. I have thus excluded these datasets from the study. For the 5% rough terrain, I have also run a search with four values (1, 2.154, 4.642, 10) Nm/rad.

8.1 Dataset 1: Systematic search, 5% Roughness, 1-10-100 Nm/rad

This dataset was generated by a systematic search with possible values of 1, 10 and 100 Nm/rad on a rough terrain of 5% roughness.
Fig. 12 shows the fitness value per configuration. This dataset shows that a very large fraction of the configurations result in low fitness values. We can see several peaks which suggest that particular configurations give much better results. This clearly indicates the big impact of the compliant elements on the gait.

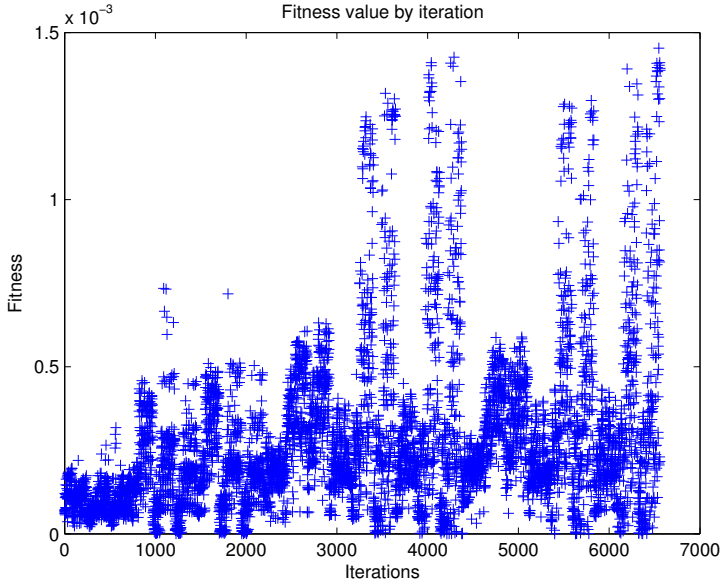


Figure 12: Dataset 1: Fitness values by iteration.

The median of the fitness values is 0.213×10^{-3}. Using this value as a threshold for clustering, we get the clusters shown in fig. 13.

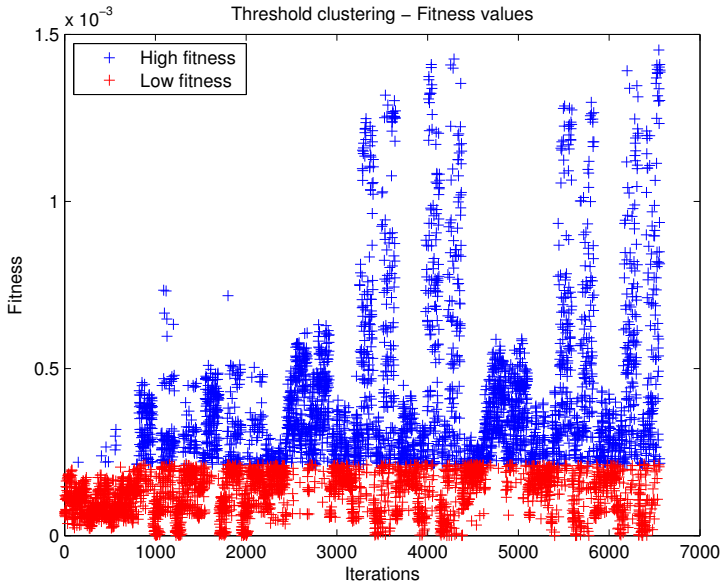


Figure 13: Dataset 1: Clustering by fixed threshold.

The next table shows the stiffness values in Nm/rad for the best five configurations (A table containing all the 30 chosen configurations can be found in Appendix B.1):

Configuration \ Element	0	1	2	3	4	5	6	7
1	100.0	100.0	100.0	100.0	100.0	10.0	10.0	10.0
2	10.0	100.0	100.0	10.0	100.0	100.0	10.0	100.0
3	10.0	100.0	10.0	10.0	100.0	10.0	100.0	10.0
4	100.0	100.0	100.0	100.0	100.0	100.0	10.0	100.0
5	10.0	100.0	100.0	10.0	10.0	10.0	10.0	100.0

Fig. 14 shows the statistics of the stiffness value of each leg for the best and worst subclusters. Remember that each subcluster contains a sixth of the configurations. Both the table and fig. 14 show quite well the pattern I have observed. In the table we can see that the best cases tend to have stiff inner elements (0 to 3). Fig. 14 support this pattern even stronger. We can see that practically all the best configurations have stiffness values of either 10 or 100 Nm/rad in the inner elements. The main exception is the front leg (elements 1 and 5).

Even more clearly, this phenomenon can be seen in fig. 15 which shows the statistics of the compliant elements by two groups for the six subclusters. The first group includes elements 0, 2 and 3 (inner elements) and the second includes 4, 6 and 7 (outer elements). Here, we can clearly see that the inner elements of the best configurations are stiff, while they are soft in the worst configuration. Configurations with medium stiffness in the inner part of the structure score in the middle of the scope. These observations corroborate Hypothesis 1 that the inner compliant elements should be stiff.

For the outer elements, we do not see a very clear tendency. Even though fig. 14 might suggest that they should be stiff, this is not very decisive and is not supported when plotted in groups. For all the subclusters, the outer elements' means are around 10 Nm/rad, which is the middle value and the box plot covers all the possible stiffness values. Thus, Hypothesis 2 is not very well supported, though not entirely rejected.

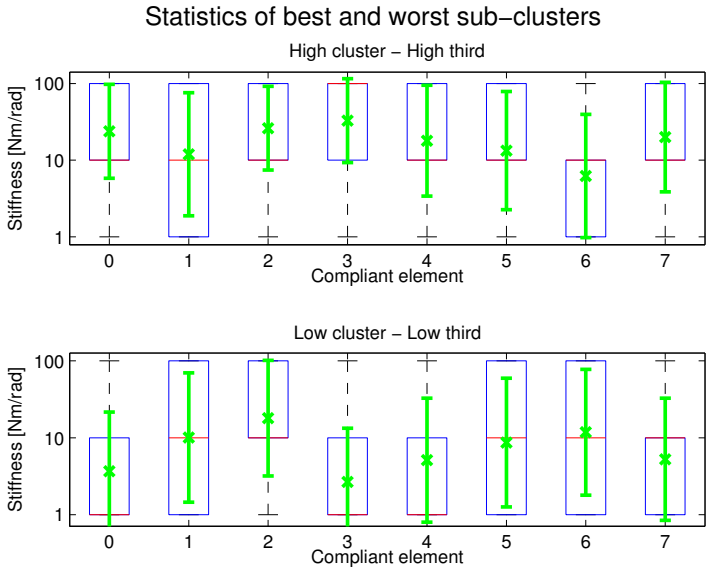


Figure 14: Dataset 1: Statistics of compliant elements for the best and worst subclusters.

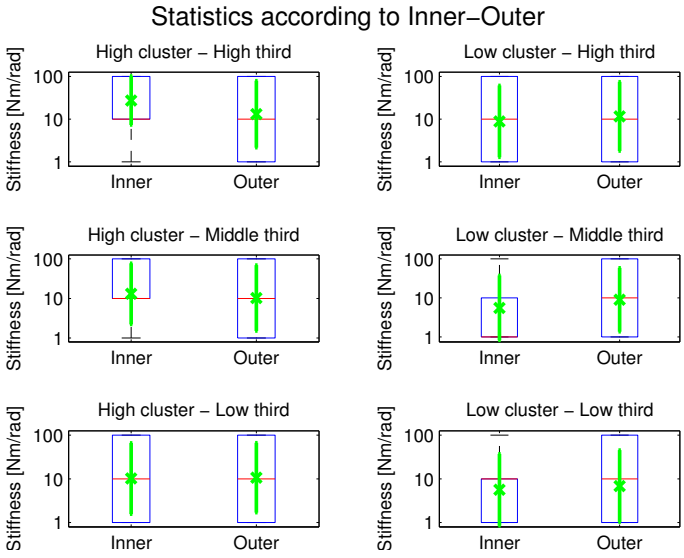


Figure 15: Dataset 1: Statistics of inner and outer compliant elements for the six subclusters.

In order to test more deeply the hypotheses, a different clustering can be used. As described before, this clustering is done by comparing the average stiffness of the inner elements against the average of the outer elements, excluding the front leg. Fig. 16 shows the fitness values for both clusters. We can see that most of the peaks are composed of blue configurations. As blue represents configurations where the inner elements are stiffer than the outer elements, this follows the hypotheses.

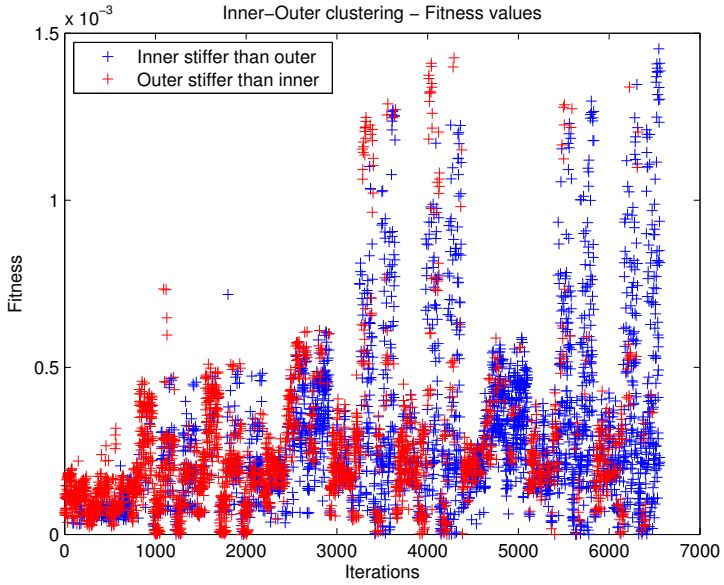


Figure 16: Dataset 1: Clustering by a Inner-Outer average comparison.

In order to verify this intuition, we need to check if the two clusters are significantly different. As the ANOVA assumptions are verified (see residuals histograms in Appendix B.1), we can run an ANOVA test on the cluster.
The ANOVA results can be seen in fig. 17 along with the box plots of both clusters in fig. 18. The value returned by the F-test in the ANOVA is 4.09324×10^{-57}. Since this value is smaller than the significance level, the two clusters are significantly different. This result corroborates the hypotheses.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Groups	0.00001	1	1.39461e-05	259.89	4.09324e-57
Error	0.00028	5290	5.36623e-08		
Total	0.0003	5291			

Figure 17: Dataset 1: ANOVA test for inner-outer clusters.

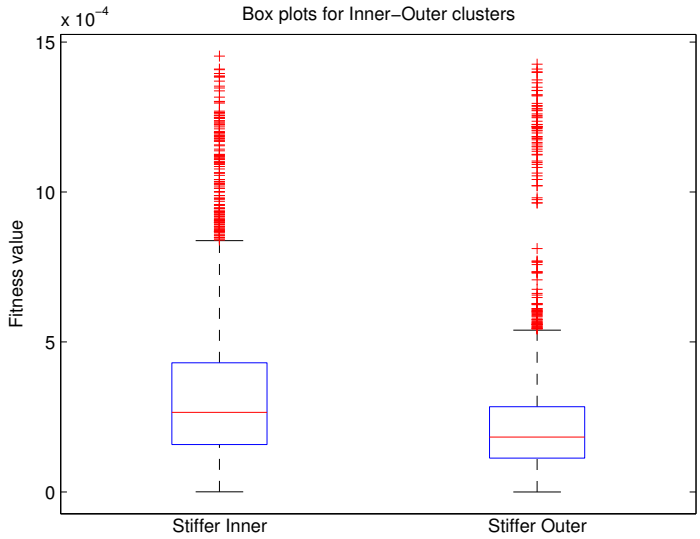


Figure 18: Dataset 1: Box plots for inner-outer clusters.

To summarize, this systematic search reinforces Hypothesis 1 that inner compliant elements should be stiff. Concerning Hypothesis 2, the outer elements can be softer, though it is not definitively clear. The results also show that too soft configurations fail to give satisfying gaits.

8.2 Dataset 2: Systematic search, 5% Roughness, 1-3.162-10 Nm/rad

Since 100 Nm/rad is a relatively high stiffness value, I wanted to see what happens when the entire structure is relatively soft. For this end I did a second systematic search, this time with possible stiffness values of 1, 3.162 and 10 Nm/rad, again on a rough terrain of 5% roughness. Note that in this dataset, the difference between configurations is smaller than in the previous dataset. Drawing definitive conclusions might be more difficult in this case.
The fitness value per configuration is shown in Fig. 19. This time, we get an even bigger fraction of low fitness configurations. However, there are still several peaks that reinforce the belief that compliance have a big impact on the gait.

The median of the fitness values is 0.163×10^{-3} and the clusters obtained by threshold clustering are shown in fig. 20.

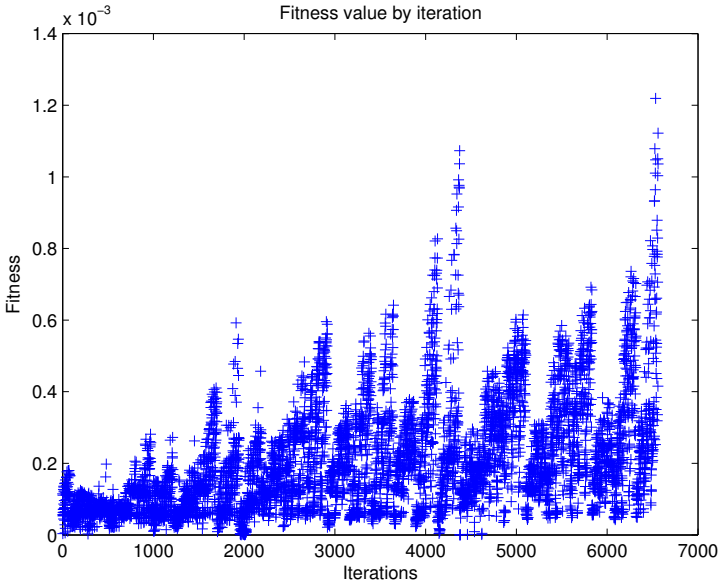


Figure 19: Dataset 2: Fitness values by iteration.

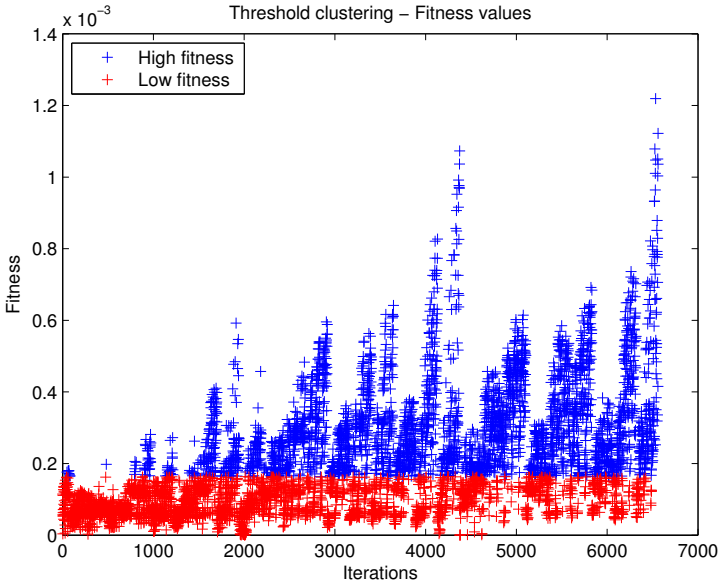


Figure 20: Dataset 2: Clustering by fixed threshold.

The best five configurations are: (A table containing all the 30 configurations can be found in Appendix B.2)

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	10.0	10.0	10.0	3.2	10.0	10.0	10.0
2	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
3	10.0	10.0	10.0	10.0	3.2	3.2	10.0	10.0
4	3.2	10.0	10.0	10.0	10.0	10.0	10.0	10.0
5	10.0	10.0	10.0	10.0	10.0	10.0	3.2	3.2

Looking at this table we see that, except for a single one, all the inner compliant elements have taken the maximum stiffness value possible (10 Nm/rad). The outer elements also seem to tend for high stiffness.
This can be also seen in fig. 21, where the statistics of the stiffness of each leg for the best and worst subclusters are shown. As before, with the exception of the front leg, all the inner elements are stiff in the best subcluster. Even more so, when looking at fig. 22 we see that for the best two subclusters, the inner compliant elements are stiff, with the best subcluster being even stiffer than the second best. (Note that the Inner median is $10Nm/rad$ for the best subcluster and $3.16Nm/rad$ for the second best). Meanwhile, the worst subcluster includes configurations with soft inner compliant elements. Again, this is a reinforcement for Hypothesis 1.
However, in this dataset we don't see a clear tendency for soft compliant elements in the outside of the structure. It seems that at these low stiffness values, the gait suffers a bit from compliant elements in all positions.

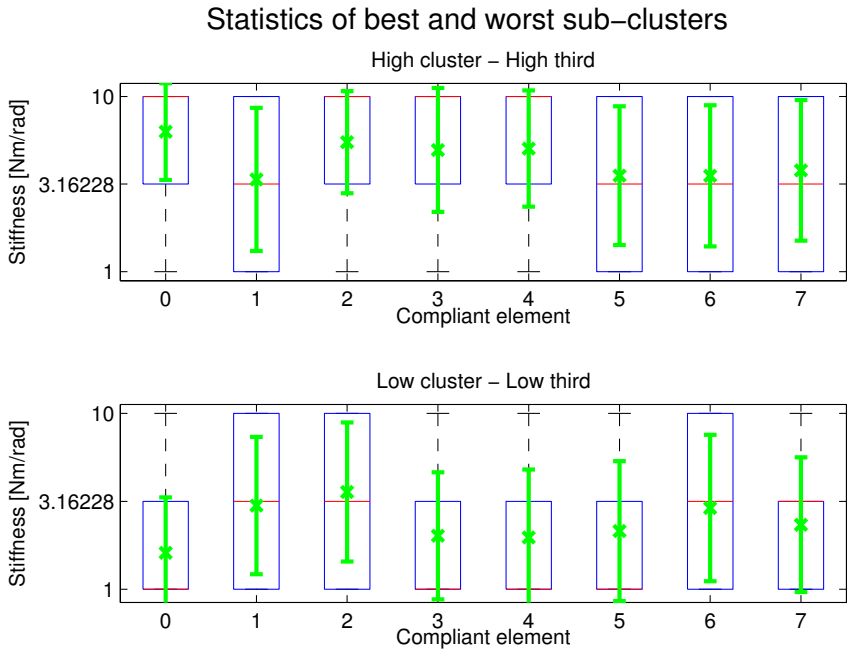


Figure 21: Dataset 2: Statistics of compliant elements for the best and worst subclusters.

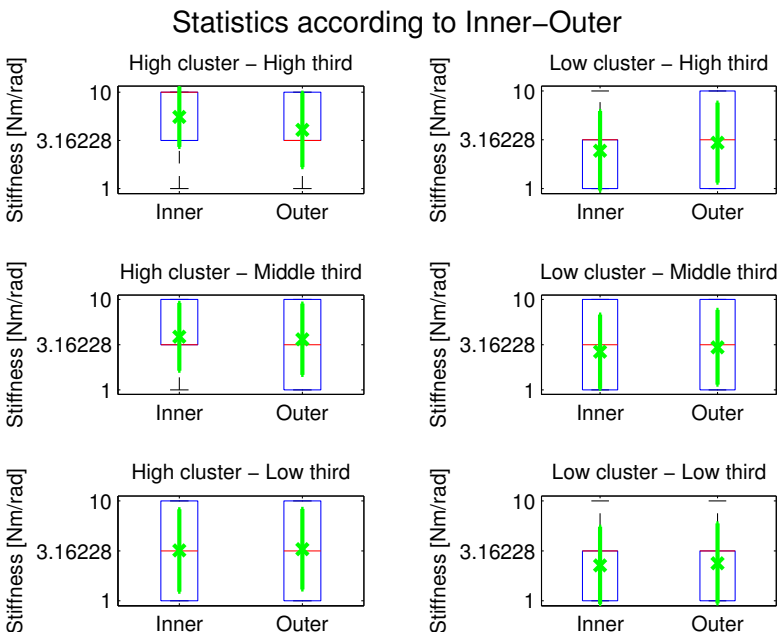


Figure 22: Dataset 2: Statistics of inner and outer compliant elements for the six subclusters.

Again, to further investigate the data, the dataset is re-clustered as described in the beginning of section 8. The new clusters can be seen in fig. 23. In this dataset, the separation between the clusters is less clear. We can still see that the majority of peaks are blue, meaning stiff inner elements.

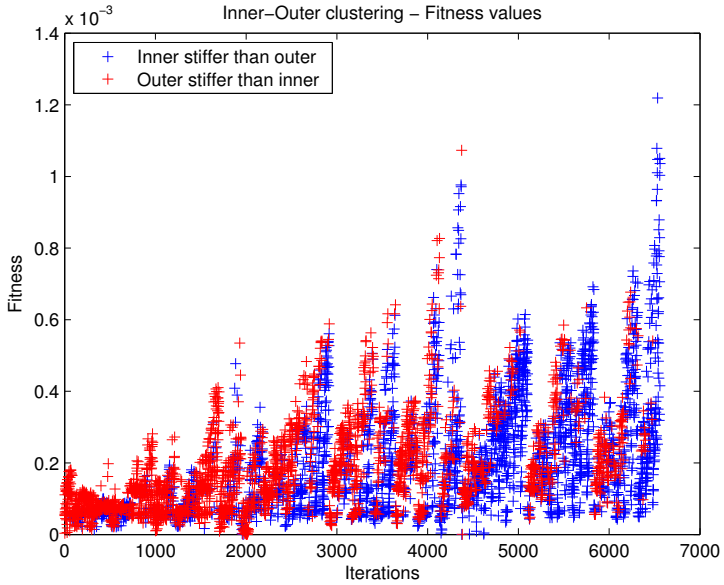


Figure 23: Dataset 2: Clustering by a Inner-Outer average comparison.

As the ANOVA assumptions are verifies (Residuals histograms in Appendix B.2), the ANOVA test

can be run on the clusters.

The ANOVA results can be seen in fig. 24 and the box plots of both clusters in fig. 25. The value returned by the F-test in the ANOVA is 3.12941×10^{-49}, which means the clusters are significantly different. This result supports the hypotheses.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Groups	0	1	4.98102e-06	222.08	3.12941e-49
Error	0.00012	5290	2.24285e-08		
Total	0.00012	5291			

Figure 24: Dataset 2: ANOVA test for inner-outer clusters.

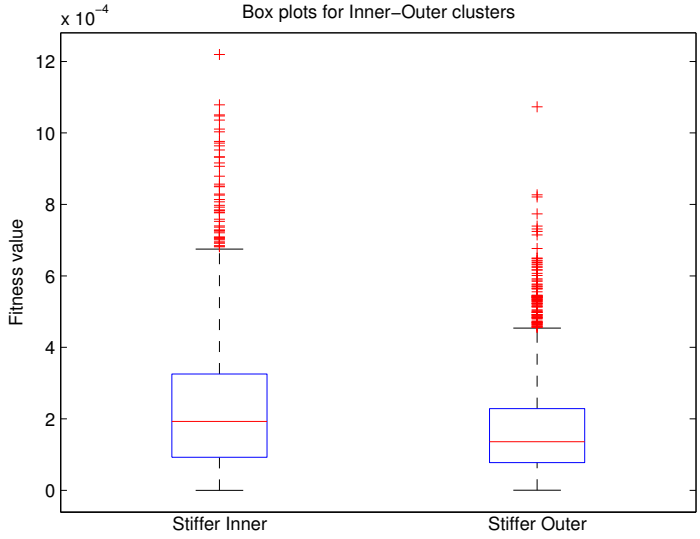


Figure 25: Dataset 2: Box plots for inner-outer clusters.

Looking at the data given by this systematic search we can reevaluate the hypothesis. This systematic search has studied robot's structures with only soft compliant elements, since high values such as 100 Nm/rad are excluded. Still, we can see some of the phenomena seen before, even though sometimes less clearly.
Hypothesis 1 is greatly supported by this systematic search. All the analysis techniques used have shown structures favoring stiff inner compliant elements.
Hypothesis 2 on the other hand, has not been largely confirmed. Even though good configurations do seem to take softer outer elements than inner elements, it is not always the case. More important, it seems that when using those possible stiffness values, soft outer compliant elements are not beneficial for the gait. This may come from the fact that all the stiffness values used now are low, so low stiffness in this dataset means very low stiffness. (In other words, if in the first dataset "low stiffness" was under 10 Nm/rad, this time it is under 3 Nm/rad).

8.3 Dataset 3: PSO, 5% Roughness

In order to validate the analysis, we have decided to run a PSO using the stiffness values of the compliant elements as variables. Such algorithm will result in the best configuration of compliant elements and could be a final test for the hypotheses. Since the PSO converges to a solution, there is no point in clustering the data. Instead we can look at the best solutions the algorithm has found. These are reported in the next table.
In the PSO, the compliant elements can take any stiffness value between 1 to 50 Nm/rad. (Blue - Stiffness lower than $15Nm/rad$, Red - Higher than $35Nm/rad$ and Cyan in the middle).

Configuration \ Element	0	1	2	3	4	5	6	7
1	47.9	44.1	44.1	35.9	49.4	31.7	11.5	19.4
2	47.4	44.0	39.7	33.6	49.6	28.0	10.7	25.1
3	47.8	44.0	38.9	32.8	48.7	28.3	10.8	24.2
4	48.2	44.2	40.9	33.3	46.0	29.2	10.9	24.9
5	49.3	44.4	45.8	34.3	43.2	29.8	9.6	25.1

As can be seen in the table, in the best solutions, all the inner compliant elements are above $30Nm/rad$ and most of them are even above $40Nm/rad$. This conforms well with Hypothesis 1.
For the outer compliant elements, we can observe a behavior that partially conforms to Hypothesis 2. As we can see, elements 5, 6 and 7 all take values under $32Nm/rad$ and even much lower. This conforms well with hypothesis 2. The only exception is element 4 whose values are very high in all the best solutions. (Note that element 0, which is on the same leg is the stiffest in the inner modules). This suggest that there might be another phenomenon taking place.
We can also note that not a single solution includes an element with stiffness below $9Nm/rad$. This indicates that using too soft modules is interfering with the locomotion.
To check the compatibility between the systematic search and the PSO, we can compare the fitness values of the best configurations. In dataset 1, the two best fitness values were 1.453×10^{-3} and 1.426×10^{-3}. In the PSO, the two best values are 1.483×10^{-3} and 1.482×10^{-3}. Thus, the best configurations in both tests give similar performances. The PSO is a bit better, which is normal since the compliant values can be fine tuned.

8.4 Dataset 4: Systematic search, 5% Roughness, 1-2.154-4.642-10 Nm/rad

All the results for this dataset can be found in Appendix B.3. As can be seen, we get similar results as in the previous datasets, mainly concerning the inner compliant elements. The best configurations are again composed of relatively stiff elements in the inner modules. This reinforces Hypothesis 1. However, Hypothesis 2 is not as well supported by the results.

8.5 Terrain with 10% Roughness

I ran the same systematic searches and PSO for this terrain. The results are reported in Appendices B.4 and B.4. Here I have only reported the statistics for grouped compliant elements with threshold clustering. This is shown in fig. 26 and 27 for the two datasets (Notice the different possible stiffness values).
The results for this terrain are a bit less clear than for the lower roughness terrain. We can still observe the same patterns, especially for the Hypothesis 1. For both datasets, the best results are again obtained by setting high stiffness in the inner modules. Hypothesis 2 however is not conclusive.

Though we don't observe a tendency for soft outer elements, they do not tend to high stiffness either.

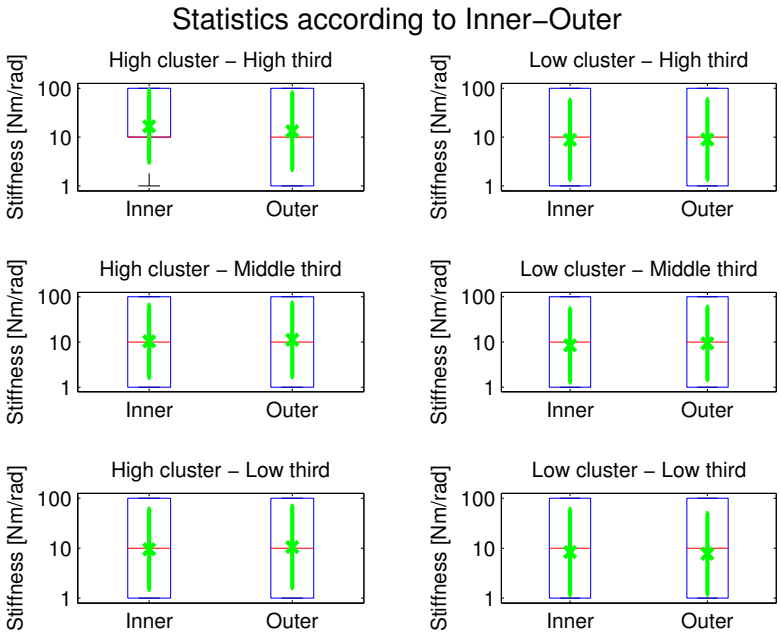


Figure 26: Dataset 10% and 1-10-100Nm/rad: Statistics of inner and outer compliant elements for the six subclusters.

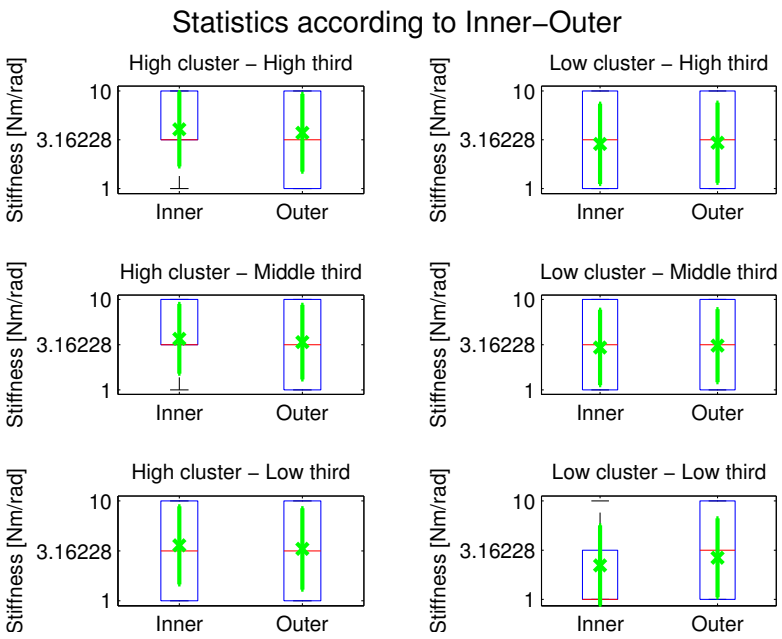


Figure 27: Dataset 10% and 1-3.16-10Nm/rad: Statistics of inner and outer compliant elements for the six subclusters.

The PSO results for this terrain are reported in Appendix B.6. We can see a pattern similar to the PSO on 5% roughness. Elements 0, 1, 2 and 4 are stiff, followed by elements 5 and 3 and element 6 is the softest. Element 7 is different from the last PSO. Thus, this PSO shows a reassembling pattern, but much less conclusive.

8.6 Qualitative Analysis

In addition to the quantitative analysis, I have also tried to assess the effect of compliance qualitatively by looking at the gait of the best and worst cases of each database. (Videos can be found in the Project CD).
The main phenomenon affecting the gait is the robot's traction. In all the worst cases, the quadruped is not able to grip well the terrain, which results in slipping. As the robot moves, the legs get a hold on to the roughness of the terrain. However, if the legs aren't firm enough, they bend upwards and slip across the terrain. This can mainly happen in too soft structures. The softness of the modules causes the structure to lose its grip and negatively impacts the gait.

8.7 Discussion

As shown during the results chapter, I have tried to analyze the data obtained from the systematic searches and the PSO in many directions. During the analysis, certain patterns have been noticed. Based on these patterns, I have formulated two hypotheses concerning the use of compliance in a quadruped structure. I have then tried to study more the data in order to reinforce or reject the hypotheses.
The analysis of the data allows me to draw some conclusions about those hypotheses.

Hypothesis 1: The inner compliant elements of the structure should have higher stiffness.
This hypothesis has been confirmed by the data. Practically all the tests done have pointed to the advantage of using high stiffness in the inner part of robot's structure. All the systematic searches have shown this behavior. When analyzing the threshold clusters, we have seen that the best results are given by configurations where the inner compliant elements are stiff. Meanwhile, the worst configurations are those where the inner elements are soft. The PSO has also corroborated this hypothesis. We have again seen that the particles converge to solutions with high stiffness in the inner modules. I can conclude that this hypothesis is very likely confirmed.

Hypothesis 2: The outer compliant elements of the structure should have lower stiffness.
This hypothesis has not been clearly decided as the first one. In most cases, I have observed lower stiffness in the outer part of the robot's structure. For the 5% roughness terrain, both searches and the PSO have shown that for the best configurations, the stiffness of the outer compliant elements is generally lower than the stiffness of the inner elements. However, this pattern is not as clear as the stiff inner elements. It seems that, while soft outer modules can improve the gait, reducing the stiffness too much has a negative effect. This would explain well the observations. While the outer elements do have lower stiffness, it is not as low as it can get. In other words, while the optimal stiffness is lower than that of the inner modules, it is not the lowest value possible. This could be explained by the fact that during the entire study the lowest stiffness value was set to $1Nm/rad$, which is a very low stiffness value. It is possible that this value is just too soft to enable efficient gait and thus successful configurations do not include such low values.

All those phenomena can be explained by thinking about the effect of compliance on the structure. The main use of compliance is to improve the robot's structure's flexibility. These characteristics are very useful when moving on rough terrain since they allow the legs to adjust their positions in order to get a better hold on the ground. The compliant elements can also help the structure to absorb collisions with obstacles and the roughness of the terrain. Intuitively, both those advantages take place mainly during interactions with the terrain. Since these interactions only happen in the outer modules of the structure, which are touching the ground, we would expect that the outer modules would benefit more from compliance.
On the other hand, the inner modules do not interact with the terrain. Their role is to transfer movement energy to the outer modules so the robot can move efficiently. Adding compliance to those parts may cause dissipation of this kinetic energy and inefficient energy transfer. This can explain well hypothesis 1 and why we see high stiffness in the inner modules.
In addition, using too soft compliant elements can cause the structure to lose its form. Such soft structure will be affected by every small change in the terrain and would not be able to efficiently transform the movement of its motors into a movement of its body.
One point that was a bit neglected during the study and the discussion is the front leg. This is because, while the other legs showed certain patterns, the front leg seemed to act much differently. This might be explained by looking and the current structure and gait. In fact, the robot moves with on leg directly to the front. Thus, this leg has the harshest interaction with the terrain since it needs to absorb all the collisions. Having stiff elements in this leg would transfer all the collisions energy into the robot structure and will harm the gait. Thus, this leg doesn't fall into any of the behaviors observed in the other legs.

9 Conclusion and Future work

As shown in this report, the use of compliance in robotic structures is a vast and interesting research field. This field has drawn the interest of the scientific community in the last few years and for a good reason. Many previous studies have shown that compliant elements can improve some aspects of robotic locomotion.
In this project, I have studied the effect of compliance on the locomotion of a particular robot. For this end, I have chosen a robotic structure I find interesting and promising - the quadruped. Being also a modular robot, this robot can be useful for many applications, for example spatial exploration, due to its adaptability.
After optimizing the gait parameters to get an optimal locomotion, the effect of the compliant elements in the structure was tested. Patterns in the data have led me to formulate two hypotheses. Hypothesis 1 claims that the inner modules of the structure should be stiff. Hypothesis 2 claims that the outer modules of the structure should be softer.
As discussed before, I have shown that Hypothesis 1 is confirmed by all the tests. This can be explained by the need to efficiently transfer the motors' energy from the inner modules to the outer ones. By setting high stiffness in the inner structure, this energy transfer is facilitated and the gait is improved.
Hypothesis 2 is concluded to be partially confirmed. A large number of tests have shown that using lower stiffness in the outer modules is beneficial for the gait. However, this pattern was not observed in all the datasets and the decrease in stiffness in these modules is not pronounced enough to be conclusive. Intuitively, the outer modules could benefit from compliance, since their role is to absorb the collisions' energy from interactions with the rough ground.
I have also shown the existence of a lower limit for the stiffness of the structure. Using very soft compliant elements have been shown to be harmful to the gait and the locomotion.
I have excluded the front leg from a big part of the analysis since it doesn't seem to fall to any of the observed patterns. This peculiar behavior might be due to its special role. Using the current structure and gait, the front leg is responsible for providing traction to move the robot and for absorbing shocks from collisions with the ground. These two roles are contradictory in the sense of compliance use. This might be the reason for the lack of definitive results on the optimal compliance to use.

During the study, I have encountered certain obstacles that had to be overcome. I sometimes had to change my approach and my methodology. The main issue I have encountered is the bug in Webots that prevented me from using a serial model. Thus, since I have used a modular model, I was not able to easily change the rotation of the quadruped in the environment. The resulting gait had seen the robot move with one leg to the front and one to the back.
The main issue with this type of gait is its lack of symmetry. When planning the gait optimization and the CPG network, I have assumed symmetry between legs. With this assumption, I supposed the existence of two front legs and two back legs. However, when using the actual gait, these symmetries are not confirmed. As could be seen in videos showing the final gait used, the two "front" legs are the right and front legs. The "hind" legs are the left and back ones. Because of the symmetry used in the CPG network, each pair is using the same gait parameters. This resulted in a gait where two legs move a lot and two hardly move.
This might have also influenced the compliance analysis, since no two legs are now symmetric. As seen in the analysis, there seems to be a connection between the placement of the leg and its role to the optimal compliance. This is also why I have avoided looking for symmetries between legs and have concentrated on the inner-outer pattern.
I believe that using a symmetric gait might also result in a second pattern of symmetry in the compliant elements. It is possible, that when using a more natural gait, where the effort is better distributed between the legs, we would observe a less inconclusive effect of the compliance on the locomotion.

Another direction to continue this work is the spatial application. Due to the complexity of the Mars terrain, designing and testing robots for this environment is not an easy task. As seen in this work, I was finally not able to get very close to a real terrain. A first interesting step would be to adjust the simulation for a better correspondence to Mars in terms of friction, softness, roughness, etc. Taking into account the dust in the environment is the next step, tough it would be extremely difficult and might require dustproof hardware.

To conclude, as seen in previous studies and during this work, compliant elements definitely have an impact on locomotion. I have shown that correctly choosing the compliance of the modules have a live-or-die effect on the gait. Even tough the current results are not conclusive enough to understand all the effects of compliance, we are able to see certain patterns. I believe that future studies could give very interesting results in this field and could improve the locomotion of many future robots.

References

[1] Alexander Sprowitz, Soha Pouya, Stéphane Bonardi, Jesse Van den Kieboom, Rico Möckel, Aude Billard, Pierre Dillenbourg, and Auke Jan Ijspeert. *"Roombots: reconfigurable robots for adaptive furniture."*, Computational Intelligence Magazine, IEEE 5, no. 3 (2010): 20-32.

[2] Auke Jan Ijspeert. *"Central pattern generators for locomotion control in animals and robots: a review."*, Neural Networks 21, no. 4 (2008): 642-653.

[3] Hossein Ahmadzadeh, Ellips Masehian, and Masoud Asadpour. *"Modular Robotic Systems: Characteristics and Applications"*, Journal of Intelligent & Robotic Systems (2015): 1-41.

[4] Hossein Ahmadzadeh, and Ellips Masehian. *"Modular robotic systems: Methods and algorithms for abstraction, planning, control, and synchronization."*, Artificial Intelligence 223 (2015): 27-64.

[5] Matthew D. Hancher, and Gregory S. Hornby. *"A modular robotic system with applications to space exploration"*, Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on. IEEE, 2006.

[6] Massimo Vespignani, Emmanuel Senft, Stéphane Bonardi, Rico Moeckel, and Auke J. Ijspeert. *"An experimental study on the role of compliant elements on the locomotion of the self-reconfigurable modular robots Roombots."*, In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pp. 4308-4313. Ieee, 2013.

[7] Massimo Vespignani, Kamilo Melo, Mehmet Mutlu, and Auke Ijspeert. *"Compliant snake robot locomotion on horizontal pipes."*

[8] Massimo Vespignani, Kamilo Melo, Stéphane Bonardi, and Auke J. Ijspeert. *"Snake robot locomotion with compliant elements."*

[9] Massimo Vespignani, Kamilo Melo, Stéphane Bonardi, and Auke J. Ijspeert. *"Role of Compliance on the Locomotion of a Reconfigurable Modular Snake Robot."*

[10] Riccardo Poli, James Kennedy, and Tim Blackwell. "Particle swarm optimization: An overview." Swarm intelligence 1, no. 1 (2007): 33-57.

[11] Stéphane Bonardi, Massimo Vespignani, Rico Moeckel, Jesse Van den Kieboom, Soha Pouya, Alexander Sproewitz, and Auke Ijspeert. *"Automatic generation of reduced CPG control networks for locomotion of arbitrary modular robot structures."*, In Proceedings of Robotics: Science and Systems, no. EPFL-CONF-200520. 2014.

[12] Serge Kernbach, Eugen Meister, Florian Schlachter, Kristof Jebens, Marc Szymanski, Jens Liedke, Davide Laneri et al. *"Symbiotic robot organisms: REPLICATOR and SYMBRION projects."* In Proceedings of the 8th workshop on performance metrics for intelligent systems, pp. 62-69. ACM, 2008.

[13] *"Codyn - Open Source Software Framework for Modeling and Simulating Coupled Dynamical Systems."*, http://codyn.net/.

[14] *M. UPenn. SMORES (Self-assembling MOdular Robot for Extreme Shapeshifting)*, http://modlabupenn.org/smores/.

[15] *"Roombots - BioRob."*, EPFL website, http://biorob.epfl.ch/roombots.

[16] *"Webots: Robot Simulator."*, https://www.cyberbotics.com/.

A Files Guide

Files marked in bold are the most recent ones. Many files are copies made to avoid clashes with the cluster.

A.1 Controllers

3 controllers: codyn-supervisor, robot, middle_cube

supervisor_cluster_noYaw: noYaw network, R without yaw

supervisor_forces: = supervisor_cluster_noYaw. For forces test. R without yaw.

supervisor_compliant = supervisor_cluster_noYaw but commented. Network_compliant parameters

supervisor_compliant_pso = like supervisor_cluster_noYaw but stiffness = job parameter (and not log value). Network_compliant parameters

robot_cluster = **robot**

middle_cube_cluster = **middle_cube**

subsubsectionPhysics Plugins ApplyForces

A.2 Webots world files

quad: Given quad model

quad_serial: Serial model

quad_modules: Modular model. supervisor, robot, middle_cube

quad_modules_rough1: 5% rough slope. supervisor, robot, middle_cube

quad_modules_rough1_cluster: same as rough1. supervisor_cluster, robot_cluster, middle_cube_cluster

quad_modules_rough1_cluster_noYaw: as rough1 but R without yaw (different supervisor). supervisor_cluster_noYaw, robot, middle_cube

quad_modules_rough2: 10% rough slope. supervisor, robot, middle_cube

quad_modules_forces: flat ground + physics plugin. Using supervisor_forces, robot_cluster, middle_cube_cluster, ApplyForces plugin

quad_modules_slope5: 5 degrees smooth slope. supervisor, robot, middle_cube

quad_modules_slope13: 13 degrees smooth slope. supervisor, robot, middle_cube

quad_modules_slope20: 20 degrees smooth slope. supervisor, robot, middle_cube

quad_modules_slope25: 25 degrees smooth slope. supervisor, robot, middle_cube

quad_modules_compliant: as noYaw, 5% rough slope. supervisor_compliant, robot_cluster, middle_cube_cluster

quad_modules_compliant10: as noYaw, 10% rough slope. supervisor_compliant, robot_cluster, middle_cube_cluster

quad_modules_compliant5_pso: as compliant, but with supervisor_compliant_pso (, robot_cluster, middle_cube_cluster)

quad_modules_compliant10_pso: as compliant10, but with supervisor_compliant_pso (, robot_cluster, middle_cube_cluster)

A.3 Job Files

Parameters: local/cluster, world, controller(s), network file

quad: local, slope13, codyn_supervisor

quad_cluster: cluster, slope13, codyn_supervisor

quad_rough_cluster: cluster, rough1_cluster, codyn_supervisor_cluster

quad_rough_cluster_noYaw: cluster, rough1_cluster_noYaw, codyn_supervisor_cluster_noYaw, network.cdn

quad_forces_cluster: cluster, quad_modules_forces, supervisor_forces

quad_forces_noR_cluster: cluster, quad_modules_forces,

quad_compliant5: cluster, quad_modules_compliant, supervisor_compliant, robot_cluster, middle_cube_cluster

quad_compliant10: cluster, quad_modules_compliant10, supervisor_compliant, robot_cluster, middle_cube_cluster

quad_compliant5_4values: Same as quad_compliant5 but with 4 steps.

quad_compliant5_pso: cluster, quad_modules_compliant5_pso, supervisor_compliant_pso, robot_cluster, middle_cube_cluster

quad_compliant10_pso: cluster, quad_modules_compliant10_pso, supervisor_compliant_pso, robot_cluster, middle_cube_cluster

A.4 Databases

quad_cluster: PSO - Smooth slope 13

quad_rough_cluster: PSO - 5% rough slope with yaw

quad_rough_cluster_noYaw: PSO - 5% rough slope without yaw

quad_rough_cluster_noR: PSO - 5% rough slope without R

quad_forces_cluster1: Flat ground Forces repetition 3N

quad_forces_cluster2: Flat ground Forces repetition 4N

quad_forces_cluster3: Flat ground Forces repetition 5N

quad_forces_cluster4: Flat ground Forces repetition 10N

quad_forces_noR_cluster1: Flat ground Forces repetition 5N (Without R in Fitness)

quad_forces_noR_cluster2: Flat ground Forces repetition 4N (Without R in Fitness)

quad_forces_noR_cluster3: Flat ground Forces repetition 10N (Without R in Fitness)

quad_forces_noR_cluster4: Flat ground Forces repetition 3N (Without R in Fitness)

quad_compliant5_1: Systematic search - 5% rough slope, values: 10, 100, 1000

quad_compliant5_2: Systematic search - 5% rough slope, values: 1, 10, 100

quad_compliant5_3: Systematic search - 5% rough slope, values: 1, 3.162, 10

quad_compliant5_4values: Systematic search - 5% rough slope, values: 1, 2.154, 4.642, 10

quad_compliant5_pso: PSO for compliance - 5% rough slope, values: 1 - 50

quad_compliant10_1: Systematic search - 10% rough slope, values: 10, 100, 1000

quad_compliant10_2: Systematic search - 10% rough slope, values: 1, 10, 100

quad_compliant10_3: Systematic search - 10% rough slope, values: 1, 3.162, 10

quad_compliant10_pso: PSO for compliance - 10% rough slope, values: 1 - 50

A.5 Network Files

network.cdn: Initial network. Used for gait optimization

network_compliant.cdn: Used for compliance study. Same gait parameters as network_rough_noYaw, with stiffnnes in addition.

network_rough_noYaw.cdn: Gait parameters resulting from PSO on 5% roughness and final fitness.

network_rough.cdn: Gait parameters resulting from PSO on 5% roughness.

network_rough_noR.cdn: Gait parameters resulting from PSO on 5% roughness and fitness with no stability.

network_slope.cdn: Gait parameters resulting from PSO on smooth slope.

B Compliance data

This appendix includes all the compliance analysis data that was not presented in the main results chapter.

B.1 Dataset 1: Systematic search, 5% Roughness, 1-10-100 Nm/rad

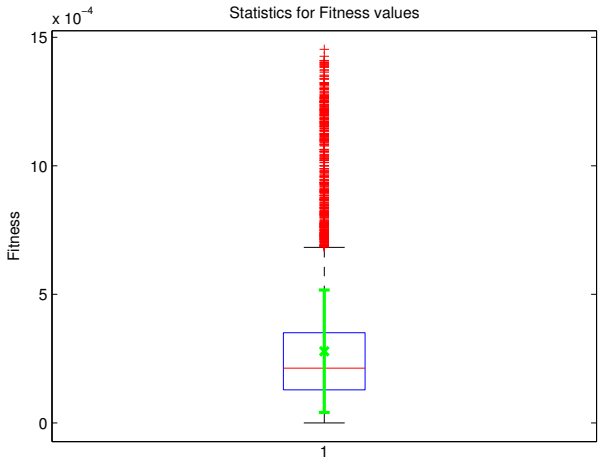


Figure 28: Dataset 1: Statistics for fitness values.

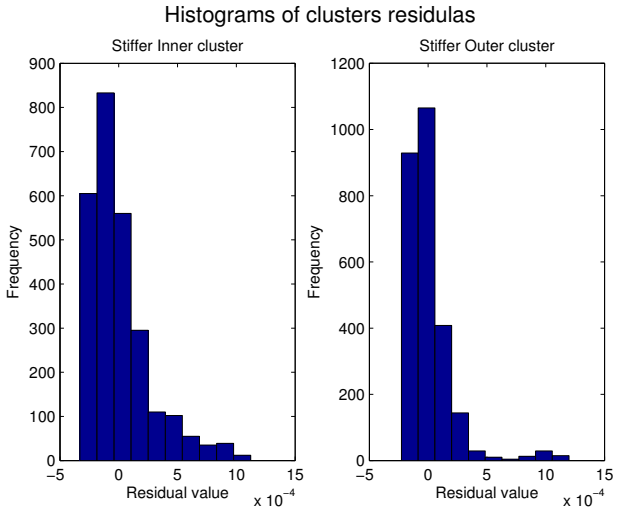


Figure 29: Dataset 1: Histograms of residulas.

First sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	100.0	100.0	100.0	100.0	100.0	10.0	10.0	10.0
2	10.0	100.0	100.0	10.0	100.0	100.0	10.0	100.0
3	10.0	100.0	10.0	10.0	100.0	10.0	100.0	10.0
4	100.0	100.0	100.0	100.0	100.0	100.0	10.0	100.0
5	10.0	100.0	100.0	10.0	10.0	10.0	10.0	100.0

Second sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	1.0	100.0	10.0	1.0	10.0	100.0	100.0
2	1.0	10.0	1.0	100.0	1.0	10.0	100.0	10.0
3	1.0	10.0	1.0	10.0	1.0	10.0	100.0	10.0
4	100.0	1.0	100.0	10.0	1.0	100.0	1.0	1.0
5	10.0	1.0	100.0	100.0	100.0	10.0	100.0	1.0

Third sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	1.0	100.0	10.0	1.0	10.0	100.0	1.0
2	10.0	100.0	10.0	10.0	100.0	100.0	10.0	1.0
3	100.0	100.0	100.0	1.0	10.0	100.0	1.0	10.0
4	10.0	10.0	1.0	10.0	10.0	100.0	1.0	10.0
5	100.0	1.0	100.0	1.0	10.0	1.0	100.0	1.0

Fourth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	100.0	100.0	100.0	1.0	100.0	10.0	1.0	1.0
2	10.0	1.0	100.0	1.0	10.0	1.0	10.0	1.0
3	10.0	10.0	100.0	100.0	1.0	10.0	10.0	100.0
4	100.0	1.0	1.0	10.0	1.0	100.0	100.0	10.0
5	100.0	100.0	1.0	1.0	10.0	1.0	10.0	1.0

Fifth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	10.0	1.0	1.0	10.0	10.0	100.0	1.0
2	1.0	10.0	100.0	100.0	10.0	1.0	1.0	1.0
3	10.0	100.0	1.0	1.0	10.0	100.0	100.0	10.0
4	1.0	1.0	100.0	100.0	10.0	1.0	1.0	100.0
5	1.0	1.0	1.0	100.0	100.0	1.0	1.0	100.0

Sixth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	100.0	100.0	100.0	1.0	10.0	100.0	10.0	100.0
2	10.0	100.0	100.0	1.0	100.0	100.0	100.0	10.0
3	100.0	10.0	10.0	1.0	10.0	100.0	100.0	10.0
4	100.0	1.0	1.0	1.0	1.0	1.0	10.0	10.0
5	1.0	100.0	100.0	1.0	1.0	1.0	100.0	10.0

B.2 Dataset 2: Systematic search, 5% Roughness, 1-3.162-10 Nm/rad

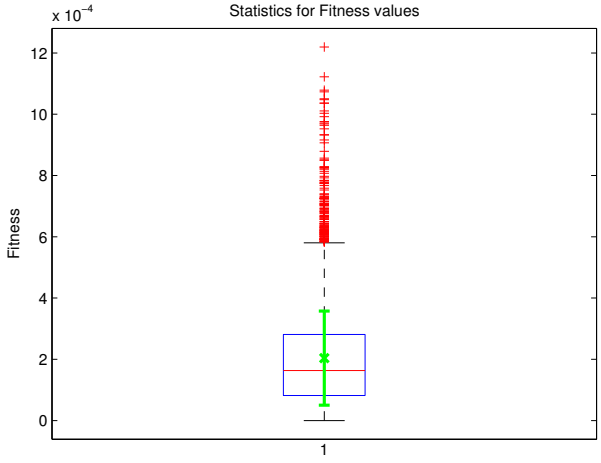


Figure 30: Dataset 2: Statistics for fitness values.

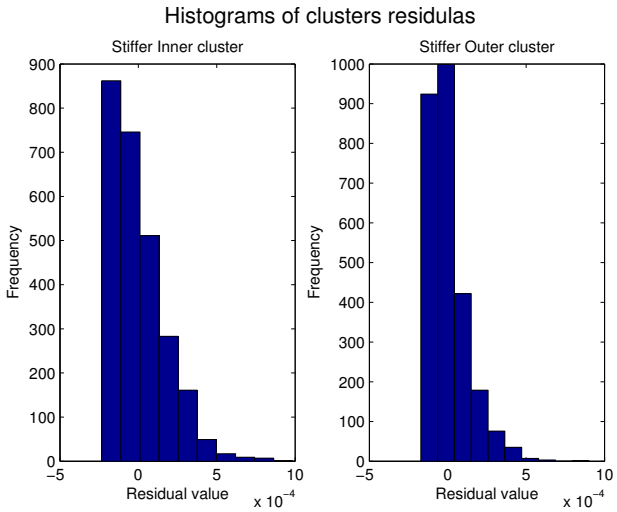


Figure 31: Dataset 2: Histograms of residulas.

First sixth:

Element / Configuration	0	1	2	3	4	5	6	7
1	10.0	10.0	10.0	10.0	3.2	10.0	10.0	10.0
2	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
3	10.0	10.0	10.0	10.0	3.2	3.2	10.0	10.0
4	3.2	10.0	10.0	10.0	10.0	10.0	10.0	10.0
5	10.0	10.0	10.0	10.0	10.0	10.0	3.2	3.2

Second sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	3.2	3.2	1.0	3.2	10.0	1.0	1.0
2	3.2	1.0	3.2	3.2	10.0	1.0	1.0	10.0
3	3.2	10.0	1.0	3.2	10.0	10.0	1.0	10.0
4	3.2	3.2	10.0	3.2	3.2	3.2	1.0	3.2
5	10.0	3.2	1.0	3.2	3.2	1.0	3.2	10.0

Third sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	3.2	1.0	1.0	3.2	3.2	10.0	3.2	1.0
2	10.0	10.0	1.0	10.0	10.0	3.2	3.2	10.0
3	3.2	1.0	10.0	10.0	1.0	1.0	3.2	10.0
4	3.2	10.0	1.0	3.2	1.0	3.2	10.0	3.2
5	10.0	10.0	1.0	3.2	10.0	10.0	3.2	1.0

Fourth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	3.2	10.0	10.0	3.2	3.2	1.0	10.0	3.2
2	1.0	10.0	10.0	1.0	10.0	3.2	1.0	10.0
3	10.0	1.0	10.0	1.0	1.0	3.2	1.0	10.0
4	3.2	10.0	10.0	3.2	3.2	1.0	3.2	10.0
5	10.0	3.2	1.0	10.0	1.0	3.2	3.2	1.0

Fifth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	3.2	3.2	1.0	3.2	3.2	10.0	10.0
2	1.0	1.0	3.2	3.2	10.0	3.2	3.2	10.0
3	3.2	1.0	3.2	10.0	1.0	3.2	3.2	10.0
4	1.0	3.2	10.0	3.2	10.0	1.0	10.0	3.2
5	3.2	10.0	10.0	10.0	1.0	1.0	3.2	3.2

Sixth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	1.0	1.0	1.0	1.0	1.0	10.0	3.2
2	1.0	10.0	10.0	1.0	10.0	1.0	10.0	1.0
3	10.0	1.0	1.0	1.0	1.0	1.0	3.2	3.2
4	10.0	1.0	1.0	1.0	1.0	1.0	1.0	3.2
5	1.0	10.0	10.0	1.0	1.0	10.0	3.2	1.0

B.3 Dataset 4: Systematic search, 5% Roughness, 1-2.154-4.642-10 Nm/rad

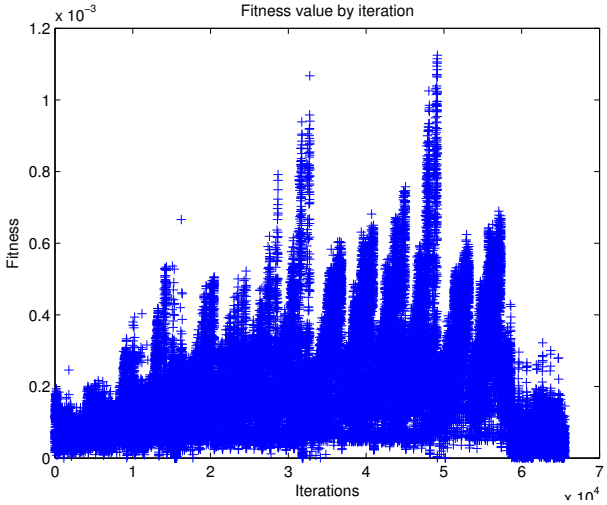


Figure 32: Dataset 4: Fitness values by iteration.

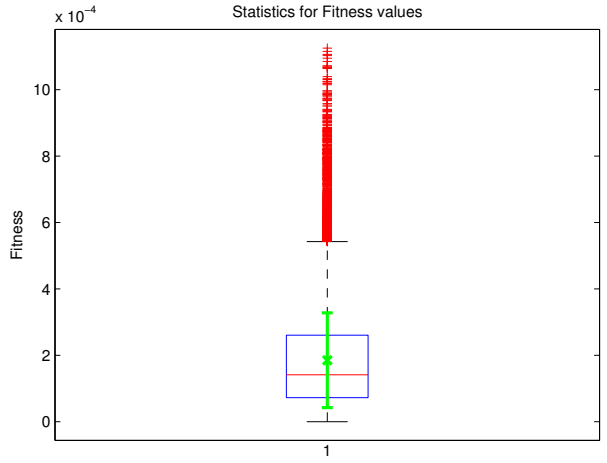


Figure 33: Dataset 4: Statistics for fitness values.

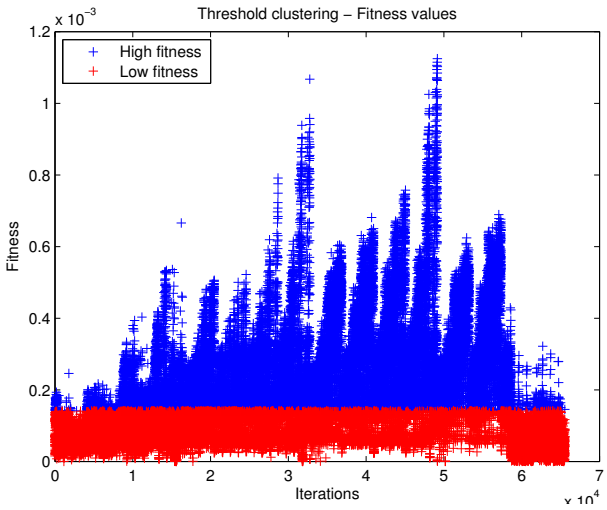


Figure 34: Dataset 4: Clustering by fixed threshold.

First sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	4.6	10.0	10.0	10.0	10.0	10.0	10.0	10.0
2	4.6	10.0	10.0	10.0	10.0	4.6	10.0	10.0
3	4.6	10.0	10.0	10.0	4.6	10.0	10.0	10.0
4	4.6	10.0	10.0	10.0	10.0	10.0	10.0	4.6
5	4.6	10.0	10.0	10.0	4.6	10.0	10.0	4.6

Second sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	4.6	1.0	10.0	4.6	2.2	4.6	10.0
2	10.0	2.2	2.2	2.2	4.6	4.6	2.2	4.6
3	4.6	1.0	10.0	1.0	2.2	2.2	2.2	4.6
4	2.2	4.6	2.2	4.6	2.2	4.6	2.2	4.6
5	10.0	1.0	2.2	2.2	10.0	1.0	2.2	4.6

Third sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	2.2	1.0	10.0	1.0	2.2	4.6	10.0	10.0
2	2.2	2.2	10.0	4.6	2.2	1.0	2.2	1.0
3	1.0	10.0	2.2	1.0	2.2	4.6	1.0	10.0
4	1.0	10.0	1.0	2.2	1.0	4.6	4.6	2.2
5	1.0	4.6	2.2	1.0	10.0	4.6	2.2	2.2

Fourth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	1.0	10.0	1.0	1.0	4.6	2.2	1.0
2	10.0	1.0	4.6	1.0	1.0	4.6	4.6	10.0
3	2.2	2.2	2.2	10.0	4.6	1.0	4.6	1.0
4	2.2	1.0	1.0	10.0	1.0	1.0	2.2	10.0
5	2.2	10.0	1.0	2.2	10.0	1.0	4.6	4.6

Fifth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	1.0	10.0	2.2	10.0	1.0	10.0	2.2
2	1.0	1.0	10.0	1.0	1.0	4.6	4.6	4.6
3	1.0	1.0	4.6	10.0	2.2	4.6	2.2	4.6
4	1.0	1.0	2.2	10.0	4.6	10.0	1.0	2.2
5	10.0	1.0	4.6	1.0	1.0	4.6	4.6	2.2

Sixth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	10.0	2.2	4.6	10.0	2.2	1.0	2.2
2	10.0	4.6	10.0	1.0	2.2	1.0	10.0	1.0
3	10.0	10.0	2.2	4.6	10.0	2.2	2.2	1.0
4	1.0	10.0	10.0	1.0	10.0	4.6	4.6	4.6
5	10.0	10.0	10.0	10.0	2.2	1.0	4.6	1.0

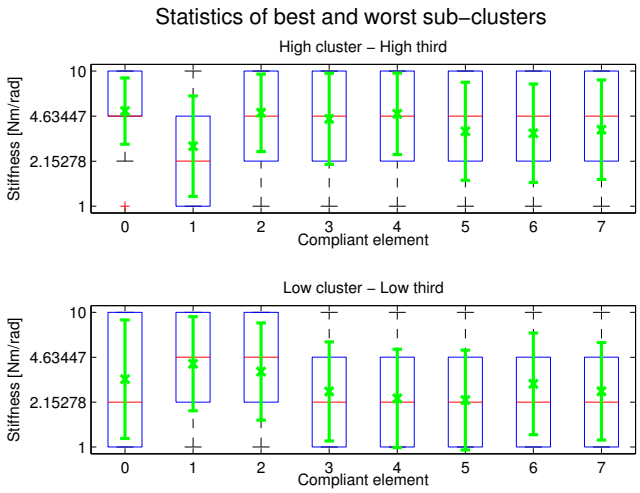


Figure 35: Dataset 4: Statistics of compliant elements for the six subclusters.

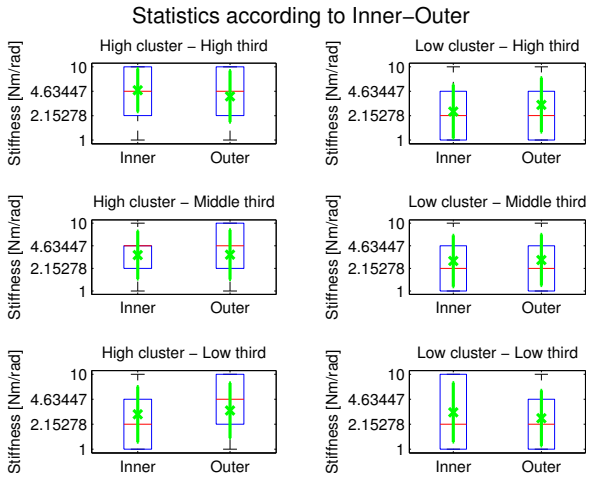


Figure 36: Dataset 4: Statistics of inner and outer compliant elements for the six subclusters.

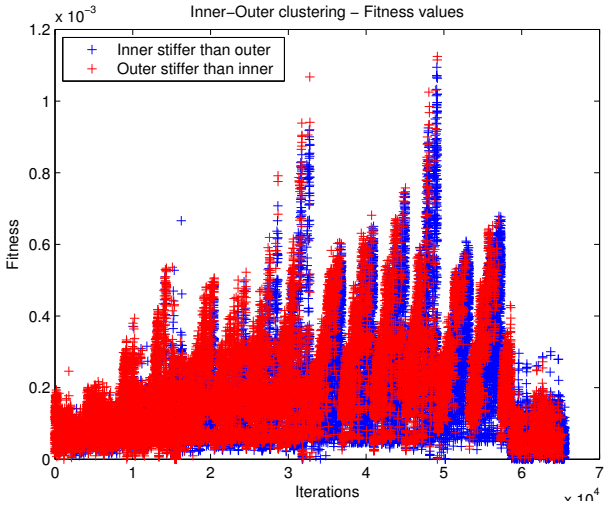


Figure 37: Dataset 4: Clustering by a Inner-Outer average comparison.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Groups	0.00001	1	7.45969e-06	367.56	1.15002e-81
Error	0.00116	56926	2.0295e-08		
Total	0.00116	56927			

Figure 38: Dataset 4: ANOVA test for inner-outer clusters.

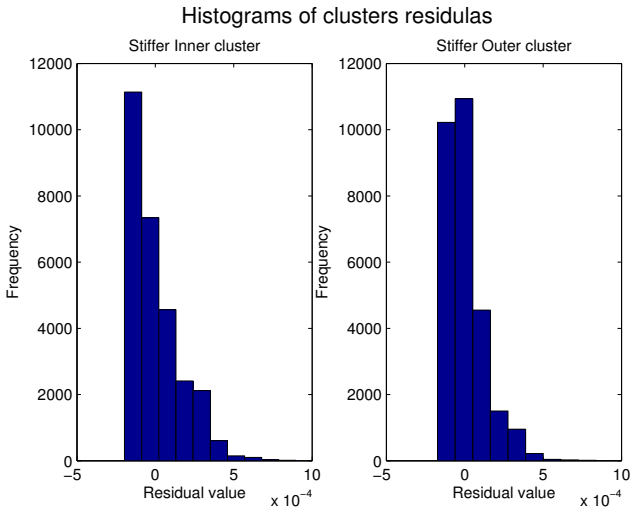


Figure 39: Dataset 4: Histograms of residulas.

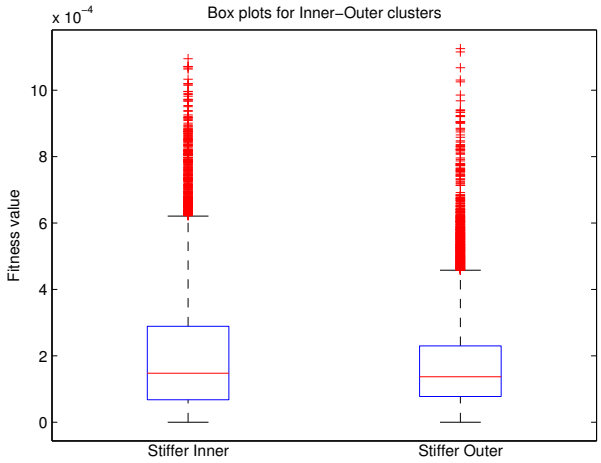


Figure 40: Dataset 4: Box plots for inner-outer clusters.

B.4 Dataset 5: Systematic search, 10% Roughness, 1-10-100 Nm/rad

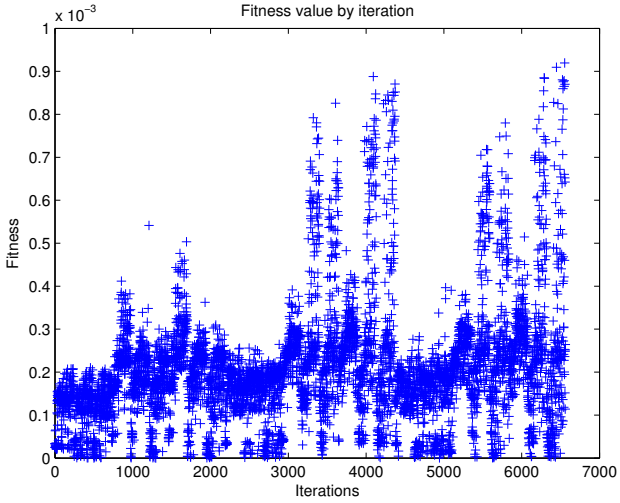


Figure 41: Dataset 5: Fitness values by iteration.

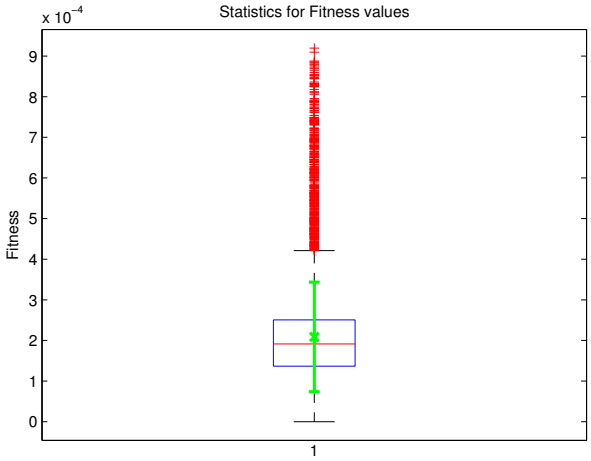


Figure 42: Dataset 5: Statistics for fitness values.

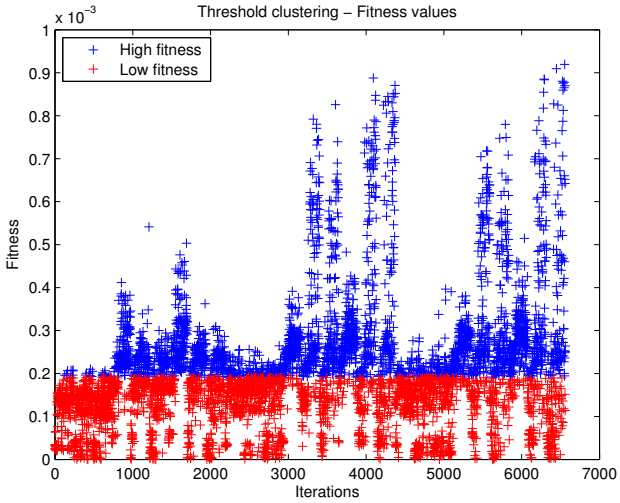


Figure 43: Dataset 5: Clustering by fixed threshold.

First sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	100.0	100.0	100.0	100.0	100.0	10.0	100.0	100.0
2	100.0	100.0	100.0	10.0	10.0	100.0	1.0	100.0
3	10.0	100.0	10.0	100.0	10.0	10.0	10.0	100.0
4	100.0	100.0	10.0	100.0	10.0	100.0	10.0	100.0
5	100.0	100.0	10.0	100.0	10.0	100.0	100.0	100.0

Second sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	10.0	10.0	100.0	1.0	10.0	100.0	1.0
2	10.0	100.0	1.0	10.0	10.0	10.0	10.0	100.0
3	1.0	10.0	1.0	10.0	100.0	10.0	100.0	100.0
4	1.0	100.0	100.0	100.0	100.0	1.0	100.0	100.0
5	100.0	10.0	1.0	1.0	10.0	10.0	100.0	100.0

Third sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	1.0	10.0	100.0	1.0	10.0	100.0	10.0
2	10.0	1.0	100.0	100.0	1.0	1.0	10.0	100.0
3	10.0	1.0	1.0	1.0	100.0	100.0	1.0	100.0
4	1.0	1.0	1.0	100.0	10.0	10.0	1.0	10.0
5	10.0	1.0	10.0	10.0	100.0	1.0	1.0	1.0

Fourth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	1.0	100.0	10.0	100.0	1.0	100.0	100.0
2	10.0	1.0	10.0	100.0	100.0	100.0	10.0	100.0
3	1.0	1.0	100.0	10.0	10.0	10.0	10.0	10.0
4	10.0	1.0	10.0	100.0	10.0	100.0	10.0	10.0
5	1.0	100.0	10.0	10.0	100.0	100.0	1.0	1.0

Fifth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	100.0	100.0	100.0	1.0	1.0	1.0	1.0
2	10.0	100.0	100.0	1.0	100.0	100.0	1.0	100.0
3	1.0	1.0	1.0	10.0	10.0	100.0	100.0	10.0
4	1.0	1.0	10.0	100.0	100.0	10.0	100.0	1.0
5	1.0	1.0	100.0	1.0	100.0	10.0	10.0	10.0

Sixth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	100.0	100.0	1.0	1.0	1.0	100.0	100.0
2	1.0	1.0	10.0	100.0	1.0	100.0	1.0	1.0
3	100.0	1.0	100.0	10.0	1.0	1.0	1.0	1.0
4	1.0	10.0	100.0	1.0	100.0	100.0	100.0	100.0
5	1.0	1.0	100.0	1.0	1.0	10.0	100.0	1.0

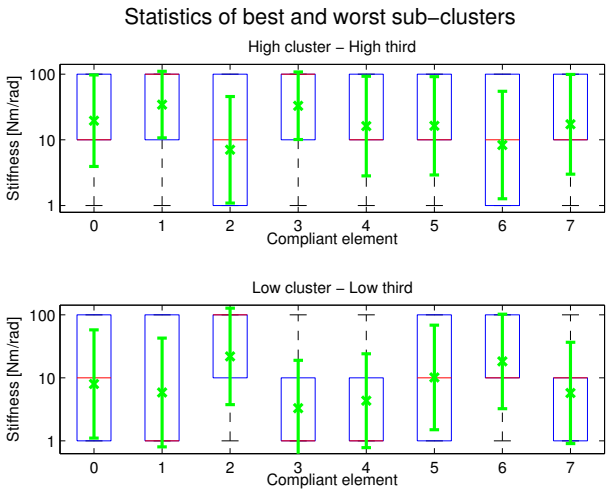


Figure 44: Dataset 5: Statistics of compliant elements for the six subclusters.

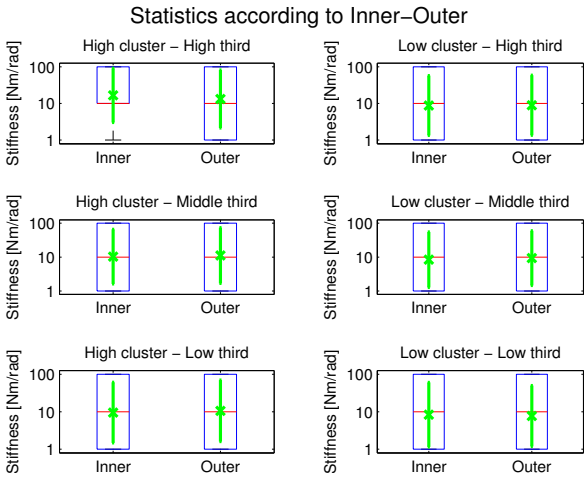


Figure 45: Dataset 5: Statistics of inner and outer compliant elements for the six subclusters.

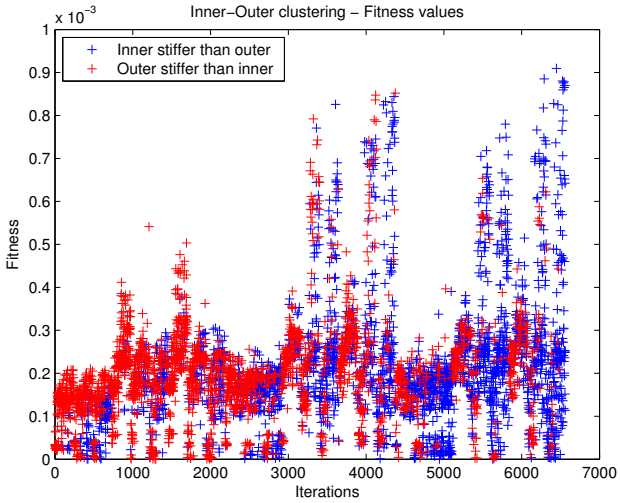


Figure 46: Dataset 5: Clustering by a Inner-Outer average comparison.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Groups	8.53962e-07	1	8.53962e-07	48.33	4.0415e-12
Error	9.34745e-05	5290	1.767e-08		
Total	9.43285e-05	5291			

Figure 47: Dataset 5: ANOVA test for inner-outer clusters.

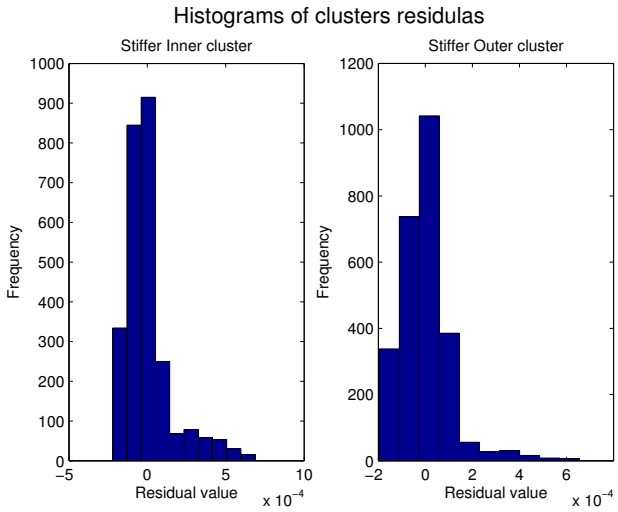


Figure 48: Dataset 5: Histograms of residulas.

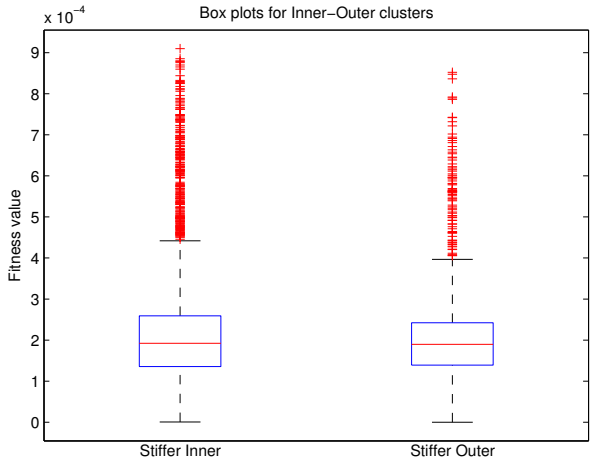


Figure 49: Dataset 5: Box plots for inner-outer clusters.

B.5 Dataset 6: Systematic search, 10% Roughness, 1-3.16-10 Nm/rad

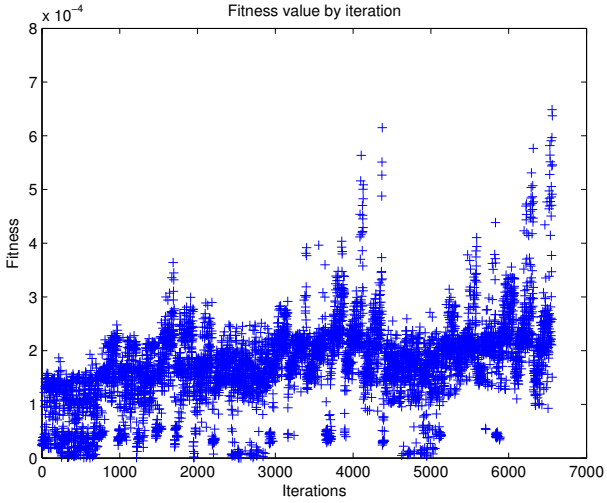


Figure 50: Dataset 6: Fitness values by iteration.

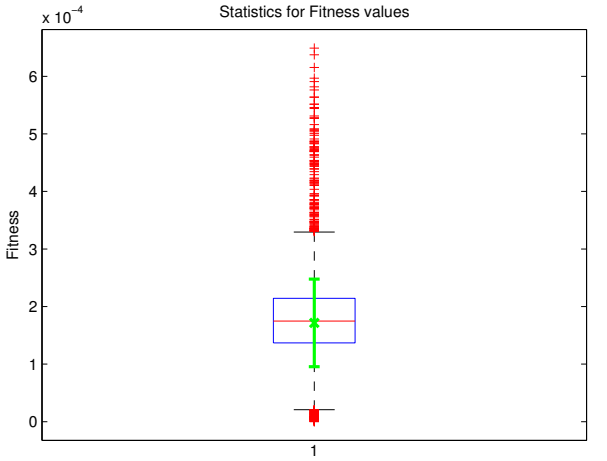


Figure 51: Dataset 6: Statistics for fitness values.

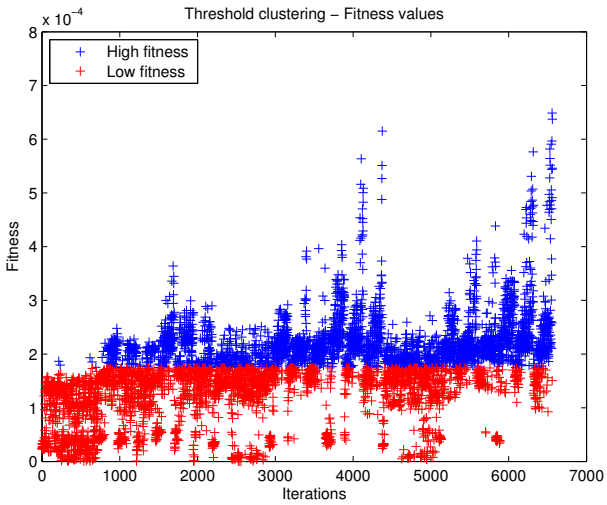


Figure 52: Dataset 6: Clustering by fixed threshold.

First sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	10.0	10.0	10.0	10.0	10.0	3.2	10.0
2	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
3	3.2	10.0	10.0	10.0	10.0	10.0	10.0	10.0
4	10.0	10.0	10.0	10.0	10.0	10.0	1.0	10.0
5	10.0	10.0	10.0	10.0	10.0	3.2	1.0	10.0

Second sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	3.2	3.2	3.2	1.0	3.2	3.2	3.2
2	10.0	10.0	3.2	10.0	1.0	1.0	10.0	10.0
3	10.0	3.2	3.2	1.0	10.0	3.2	3.2	10.0
4	10.0	1.0	10.0	1.0	10.0	1.0	1.0	10.0
5	10.0	3.2	10.0	10.0	3.2	10.0	3.2	10.0

Third sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	10.0	1.0	1.0	10.0	3.2	3.2	10.0	3.2
2	3.2	1.0	1.0	3.2	10.0	1.0	1.0	1.0
3	1.0	10.0	1.0	1.0	3.2	10.0	1.0	10.0
4	3.2	1.0	1.0	3.2	10.0	10.0	3.2	1.0
5	1.0	10.0	10.0	3.2	10.0	10.0	10.0	1.0

Fourth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	10.0	10.0	1.0	1.0	10.0	1.0	10.0
2	10.0	3.2	3.2	1.0	3.2	1.0	3.2	3.2
3	10.0	1.0	10.0	3.2	10.0	3.2	10.0	10.0
4	10.0	1.0	3.2	10.0	1.0	3.2	10.0	3.2
5	3.2	10.0	3.2	1.0	3.2	10.0	10.0	10.0

Fifth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	1.0	1.0	10.0	10.0	3.2	10.0	3.2
2	3.2	1.0	1.0	10.0	1.0	10.0	1.0	1.0
3	1.0	10.0	3.2	3.2	3.2	1.0	1.0	10.0
4	3.2	1.0	10.0	3.2	3.2	1.0	1.0	10.0
5	3.2	10.0	10.0	3.2	1.0	1.0	10.0	10.0

Sixth sixth:

Configuration \ Element	0	1	2	3	4	5	6	7
1	1.0	1.0	10.0	1.0	3.2	10.0	10.0	3.2
2	3.2	1.0	3.2	3.2	1.0	10.0	3.2	3.2
3	3.2	1.0	3.2	3.2	1.0	3.2	3.2	3.2
4	1.0	1.0	10.0	3.2	3.2	10.0	3.2	3.2
5	3.2	1.0	3.2	3.2	3.2	10.0	1.0	3.2

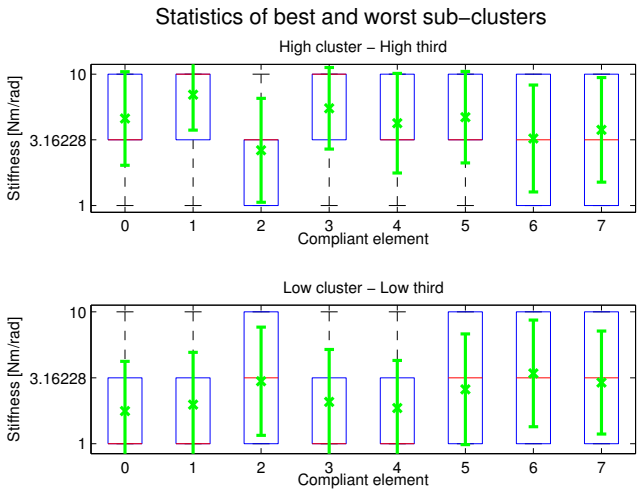


Figure 53: Dataset 6: Statistics of compliant elements for the six subclusters.

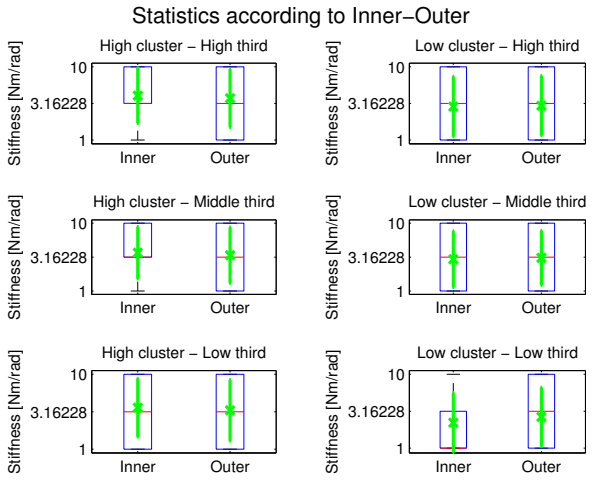


Figure 54: Dataset 6: Statistics of inner and outer compliant elements for the six subclusters.

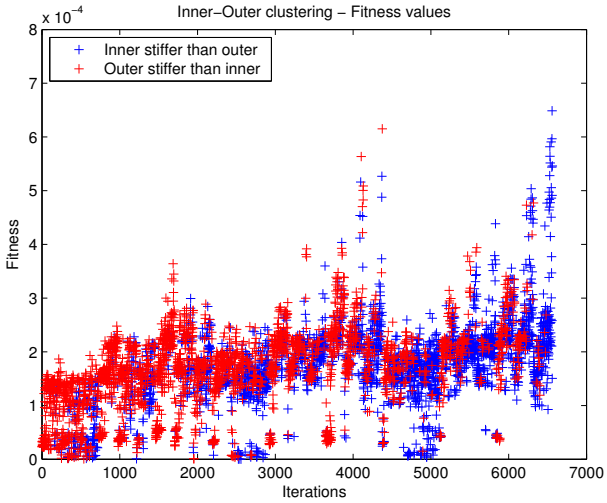


Figure 55: Dataset 6: Clustering by a Inner-Outer average comparison.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Groups	3.74064e-07	1	3.74064e-07	66.81	3.70153e-16
Error	2.96166e-05	5290	5.5986e-09		
Total	2.99907e-05	5291			

Figure 56: Dataset 6: ANOVA test for inner-outer clusters.

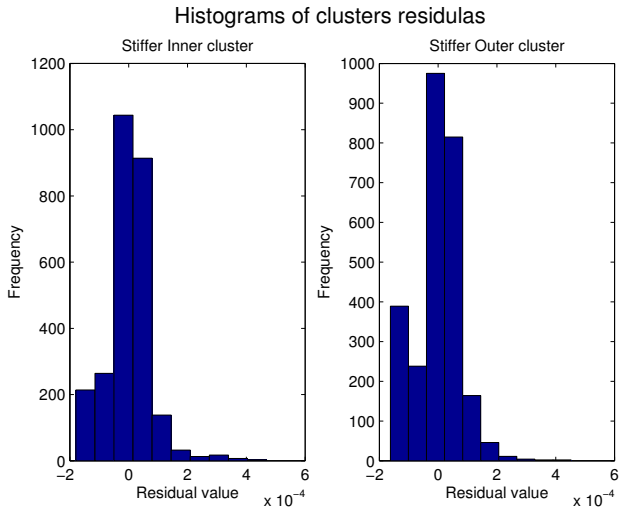


Figure 57: Dataset 6: Histograms of residulas.

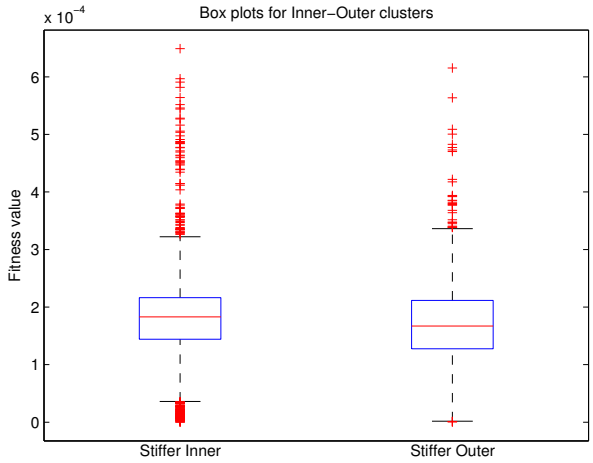


Figure 58: Dataset 6: Box plots for inner-outer clusters.

B.6 Dataset 7: PSO, 10% Roughness

Configuration \ Element	0	1	2	3	4	5	6	7
1	47.1	46.2	38.2	19.8	44.1	24.7	6.7	46.3
2	47.1	46.2	38.2	19.8	44.1	24.7	6.7	46.3
3	47.1	45.9	38.0	19.8	44.1	24.7	6.7	46.3
4	47.1	46.2	37.0	20.1	42.3	24.8	6.6	46.7
5	49.5	44.7	39.9	19.8	39.4	20.9	2.3	46.5