University of Southern Denmark

Bacholor Thesis

Robot Systems 6th semester - Spring 2016

Embodied control of a dung beetle inspired hexapod

Theis Strøm-Hansen
270792
thstr13@student.sdu.dk

Mathias Thor
210393
mthor13@student.sdu.dk

Supervisors: Leon Bonde Larsen & Poramate Manoonpong

Project period: February 1, 2016 - June 1, 2016

Abstract

This study presents a different approach to modelling biologically-inspired hexapods. Many projects concerning biologically-inspired hexapods are greatly influenced by approximations, which is commonly shown in oversimplified leg kinematics. The focus of these projects is usually on the locomotive aspect of the model and thus many of these hexapods are unable to perform manipulative related tasks.

This study will try to accurately model the dung beetle and its complex kinematics, based on actual measurements of the dung beetle species *Geotrupes stercorarius*. The intention is to exploit its locomotive and manipulative behavior by doing as few approximations as possible.

A position based version of the WALKNET controller is implemented on the dung beetle model, in order to test the performance and stability of its locomotive behavior. The manipulative behavior of the model is however tested through static positions that the dung beetle is commonly seen in, where the actual controller implementation of this behavior is left for future study.

Preface

This study is drawn up during the bachelor thesis in robot systems at University of Southern Denmark in the spring term of 2016, by a team consisting of two students. The period of the study was from February 1th to June 1th. In the future an extension to the bachelor thesis will be made. The primary focus of the continued study will be the implementation of manipulative behavior in the developed dung beetle model.

The primary areas of responsibility in the study are listed below. The process has however been hugely influenced by collaboration, meaning that both students have been involved in almost every part of the thesis.

Mathias Thor:	Main focus was the controller and the modeling
Theis Strøm-Hansen:	Main focus was the biological investigation, lpzrobots and tests

Reading guide

In section 2, the observations and measurements done on specimens of the dung beetle species *Geotrupes stercorarius* will be reported. In section 3 the results from section 2 are used to create a 3D model of the dung beetle. In section 4 the locomotion controller, called WALKNET , which enables the dung beetle model to walk, will be implemented. In section 5, the most important results from locomotive experiments and tests of the dung beetle model in common dung beetle positions will be reviewed. The discussion and conclusion will then take place throughout section 6 and 7.

Furthermore the `C++` project and the MATLAB scripts used during this study can be seen in the GITHUB repositories listed below.

`https://github.com/CurrentH/gorobots_edu`

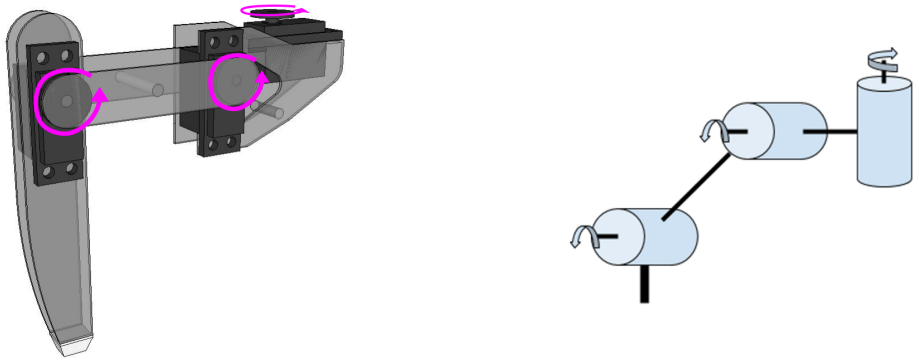
`https://github.com/CurrentH/lpzrobots`

The MATLAB scripts and the most used `C++` files are located in the `gorobots_edu` repository in the following file path: `gorobots_edu/practices/dungbot`.

Contents

1 Introduction

Biologically-inspired robotics is a growing field, aimed at imitating natural features of walking animals with artificial legged locomotion systems [1]. However, many of the projects concerning biologically-inspired robotics use animal models that are heavily influenced by approximations and as a result the kinematics differ from the real ones. Examples of these are the HECTOR robot based on the studies of a stick insect [2] and the AMOS robot inspired by a cockroach [3]. Oversimplified leg kinematics is one of the most common approximations and it is also the reason why most of today's models have leg kinematics similar to the one in figure 1, where each joint controls only one degree of freedom for the endpoint.



(a) Example of the simple hexapod leg. Image from [4]

(b) The kinematics of the simple hexapod leg

Figure 1: Robot leg based on approximations of an insect leg

These robots have been designed entirely for locomotion purposes, thus additional mechanics and actuators need to be installed in order to perform manipulation related tasks [1]. A more preferable solution is to use the same legs for both locomotion and manipulation, which is exactly what the dung beetle achieves. The dung beetle is a six legged insect, that can both walk and manipulate spherical objects. It uses these skills for transporting a ball of dung away from enemies into soft ground where the beetle can bury the dung [5]. Seen in a robotic context, the dung beetle shows fascinating locomotion abilities due to the fact that they use their legs for both locomotive and manipulative related tasks [1].

This study will investigate the anatomy and kinematics of a real life dung beetle, which will hopefully result in a more accurate biologically-inspired model that when used in a simulation is able to elicit a gait behavior comparable to that of the real dung beetle. The primary focus will be the locomotive behavior, but the dung beetle model should, in theory, also be able to manipulate a dung ball. This behavior will however not be tested

in this study and is instead left for future work.

There exist various ways of controlling the locomotion of hexapods and they can roughly be separated into three categories including; central pattern generation approaches, finite state approaches, and coordination-based approaches [6]. In the central pattern generation approach, a gait is selected by the designer and a central pattern generator is used to provide each leg with a trajectory that they need to follow [6]. Many approaches to hexapod locomotion follows this paradigm, as represented by the work of Lee and Lee [7], Zielinska et al. [8], and others. The finite state approach incorporates a set of conditions that place the hexapod into one of several states, as determined by a predetermined rule set for various types of environmental interaction (i.e., climbing stairs, walking over flat terrain, etc.) [6]. Examples of this work include that of Tanaka and Matoba [9], Saranli et al. [10], and others.

Both the central pattern generation and the finite state approaches depend on choices made by a designer, which results in static gaits and non-dynamic behavior. The coordination based approach is one in which the gait is not statically chosen, instead it is an emergent behavior resulting from some sort of coordination system [6]. This dynamic approach is preferable when dealing with biologically-inspired robotics, where the gait is to emerge from both the kinematics of the model and the environment around it. Cruse et al. utilized the coordination-based approach and in essence "reverse-engineered" the neural circuitry of a stick insect. Based on their investigations, they proposed a system of interconnected neural networks (collectively termed WALKNET) that emulates the circuitry that coordinates locomotion in the insect [6].

The overall objective is to make an accurate model of a dung beetle and equip it with the WALKNET controller. The hypothesis is that the stick insect inspired WALKNET controller will work on the accurate dung beetle model, where the expected result from using WALKNET is the elicitation of a gait behavior comparable to that of the real dung beetle.

2 Biological investigation

To develop a biologically plausible model of a dung beetle it is necessary to know its kinematics and anatomy. The initial research for this study was therefore to find literature describing this. However it turned out to be a difficult task as the dung beetle anatomy with regards to robotics is not well described, and thus it became apparent to start with researching the kinematics and anatomy of the beetle.

At a seminar by Prof. Dr. Stanislav N. Gorb from the Zoological Institute of Kiel University a discussion on the dung beetle was brought up. The Zoological Institute in Kiel is currently working on dung beetles which led to an arrangement where it was possible to travel to Kiel and study their specimens of the species *Geotrupes stercorarius*, which became the reference beetle for this study (seen on figure 2).

Figure 2: Picture of the a *Geotrupes stercorarius*. Image by [11]

The reference species *Geotrupes stercorarius* lives in Northern Europe where the ground is soft [12]. Its incentive for rolling dung balls is therefore non-existing, since it can bury the dung right away. However it is important to note that even though the dung beetle species vary in shape and size, they all presumably have the same basic kinematics (*Personal communication with Prof. Dr. Stanislav N. Gorb at Kiel University*), which are the most important part of the model, as they are key to the locomotive and manipulative behavior.

During a personal conversation with Prof. Emily Baird from Lund university it was found that the dung beetle walks with a tripod gait, which is also shown in a video of a dung beetle walking from KekPafrany [13]. An example of the tripod gait can be seen on figure 3. During the tripod gait the robot will only have three legs on the ground at the time, except in the transition where the legs are switching from being on the ground to lifted, and vice versa.

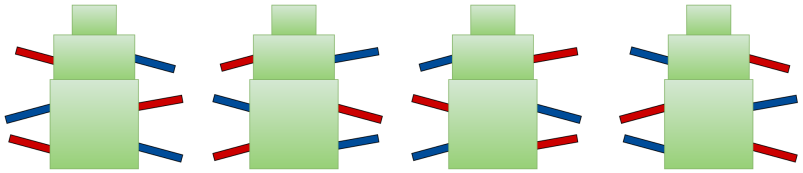


Figure 3: Illustration of a tripod gait. The legs in blue are lifted, while the red colored legs have ground contact

The main body of the dung beetle consist of three parts called abdomen, thorax and head. These parts are connected through two active joints, one between the thorax and the abdomen (TA-joint) and another between the head and the thorax (HT-joint). The legs consist of six segments, called coxa, trochanter, femur, tibia, tarsus, and pretarsus. Each of these segments are connected by an active joint; one that connects the body and coxa (BC-joint), coxa and trochanter (CT-joint), trochanter and femur (TF-joint), and femur and tibia (FT-joint). The TF-joint was however found by Canio et al. [1] to have almost no movement and the trochanter is therefore merged into the femur segment, which yields a coxa and femur joint (CF-joint) instead. Tarsus is formed by a series of five parts, where it was chosen to merge pretarsus together with the last segment [1]. The examined dung beetle body parts and joints are shown in figure 4.

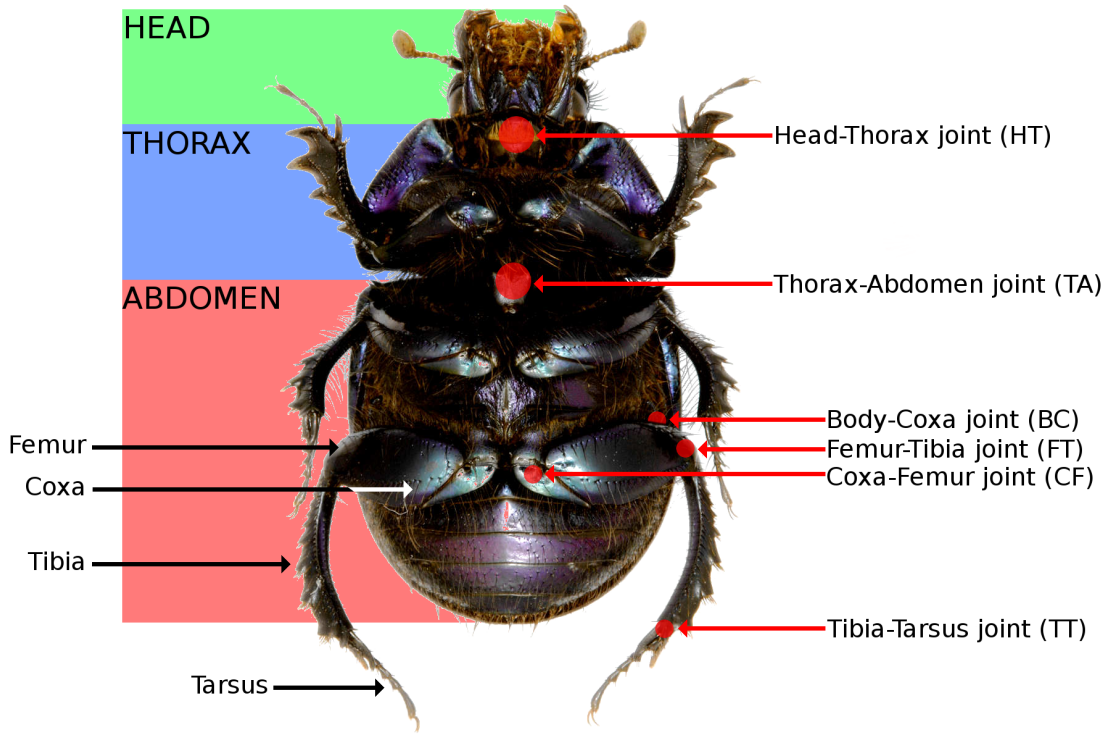


Figure 4: Body part and joint names of the dung beetle. Coxa placement is shown with a white arrow indicating that it is located behind the femur. Joints are indicated with red dots. Dung beetle image from [14]

In order to model the dung beetle, a specimen from the Zoological Institute of Kiel University was thoroughly measured, which included weights, and dimensions of all the individual parts of the dung beetle. Different setups were made to find the rotational range of motion for all the joints, as well as the position of the coxa with respect to the body.

The different equipment used for measuring the dung beetle is described in the list below:

- A microscope for visual inspection of the body parts and joint movements.
- A measuring microscope which is a device that measures 3D Cartesian positions given the center and focus of the microscope. This was used for measuring different points on the dung beetle.
- A micro-scale with a high precision down to 1 microgram.
- Bee wax used to fixate the joints in different configurations, during measurements.

The weights and dimensions of the different body parts was found by detaching and measuring each individual part (shown on figure 5), with the micro-scale and the measuring microscope. The results from these measurement are shown in appendix B table 3, where a variation was found in the dimensions of the legs. The table shows a variation where the hind legs are the longest, then the middle legs and finally the front legs as the shortest.

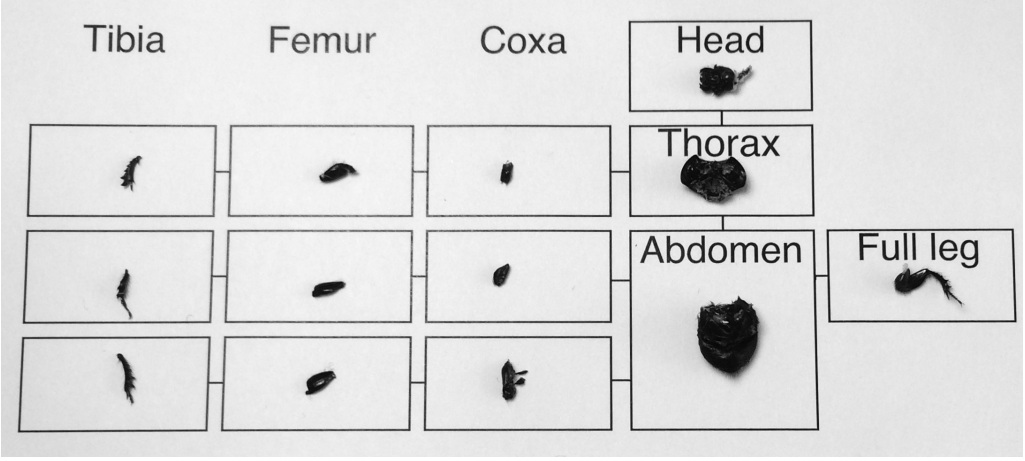
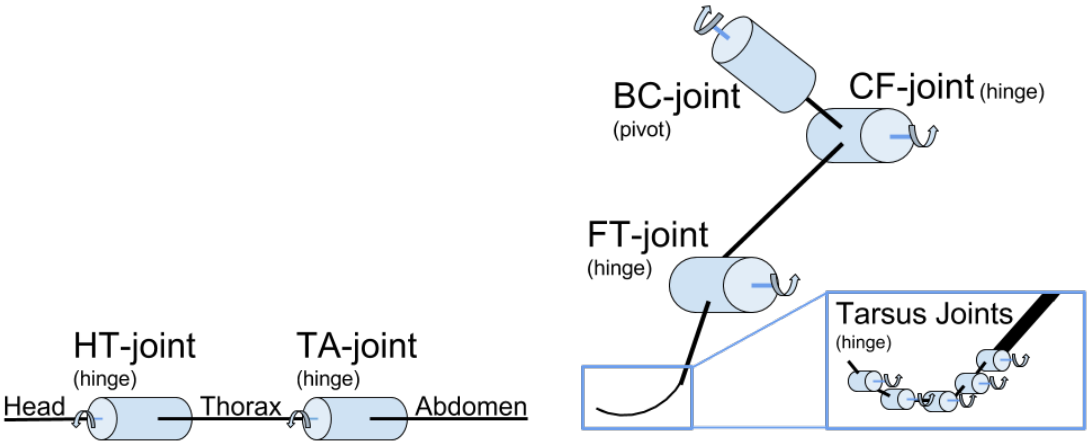


Figure 5: Detached dung beetle limps, ready for weighing and dimension measurements. A map was made to keep the body parts separated.

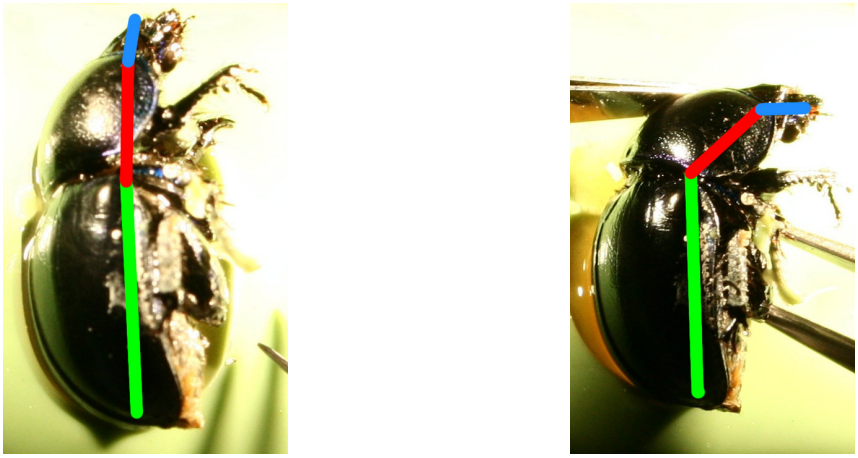


(a) Kinematics for the body

(b) Kinematics for the leg

Figure 6: Kinematics for the dung beetle

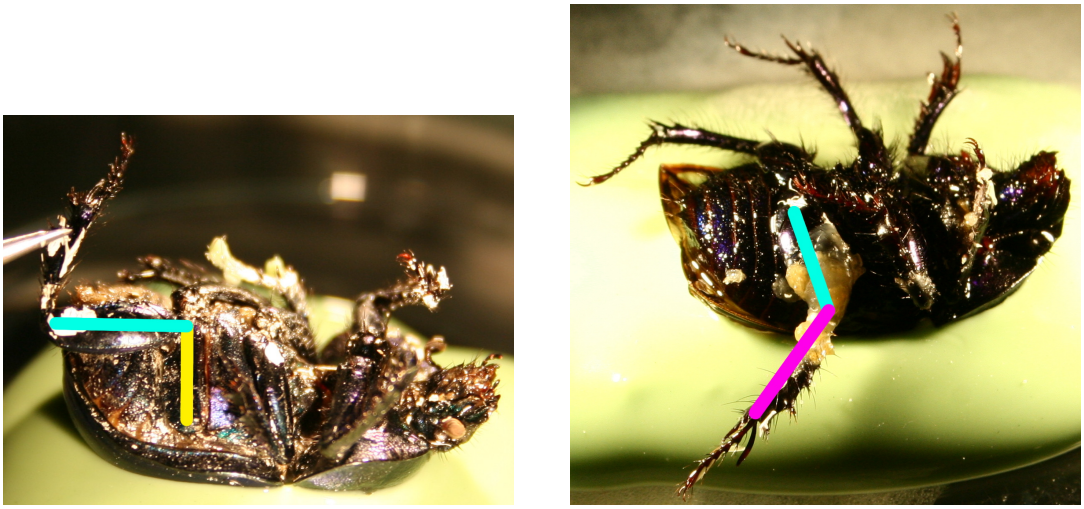
The kinematics derived in Kiel provides both the range of motion and axes for the different joints. Figure 6a shows the derived kinematics of the body, where two hinge joint are present. These joints are able to bend the beetle downwards, each by 45° resulting in a 90° bend derived from visual inspection of figure 7a and 7b.



(a) Configuration of the dung beetle where the HT- and TA-joint are not bended

(b) Configuration of the dung beetle where the HT- and TA-joints are bended

Figure 7: Illustration of the dung beetle in a straight and bended position. Green indicated the abdomen, red indicated the thorax and blue indicated the head



(a) The CF-joint in its maximum position (b) The FT-joints in its maximum position

Figure 8: The maximum position of the legs where the joints have been fixated with bee wax

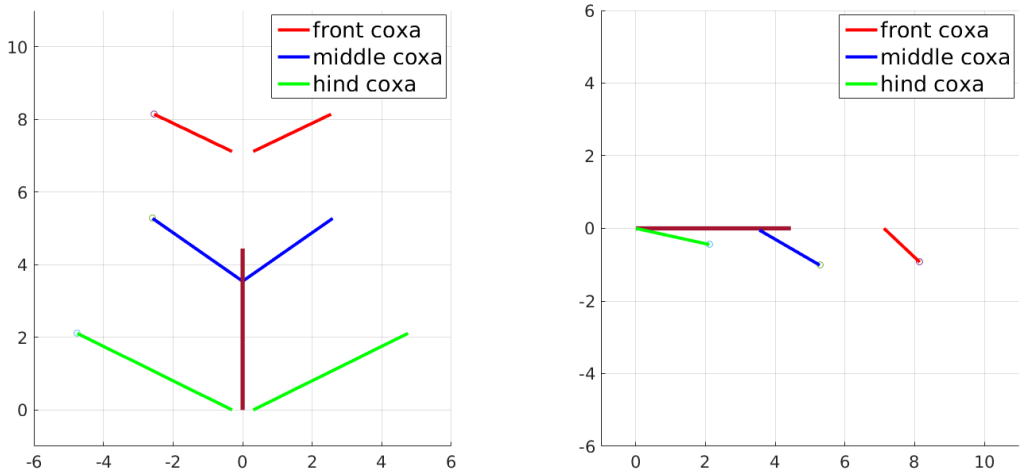
Figure 6b shows the derived kinematics of a dung beetle leg, where a total of eight joints are present. Each leg has the same joint configuration, but with different rotational ranges of motion in the BC-joints. Looking at figure 4 on page 4 it can be seen that the dung beetle has bilateral symmetry, where the two hind legs, the two middle legs and the two front legs are mirrored [15]. The BC-joints was found to be pivot joints, where both the CF- and FT-joints are hinge joints.

It is possible for the dung beetle to retract its legs completely so that they are hidden under the beetle preventing predators from eating them (*Personal communication with PH.D Alexander Kovalev at Kiel University*). This fact can be used to find the rotational range for the CF- and FT-joints, as it is the angle between the retracted position and the stretched position seen in the figure 8. Examination through visual inspection showed that the rotational range of the CF-joints are 90°, and the FT-joints are 170°.

The bilateral symmetry made it possible to find the rotational range of motion for the BC-joints by measuring the positions of the BC-, CF- and FT-joints in a minimum and maximum position on only one side of the dung beetle. At both the minimum and maximum position two vectors representing the coxa, and femur were created. This yielded four vectors, two for each position. A plane could then be created for each configuration utilizing the vector pairs, where the angles between the planes are representing the rotational range of motion for the BC-joints. This experiment showed a variation in the rotational ranges for the different BC-joints. The hind leg pair can rotate in a range of approximately 160°, the middle leg pair approximately 116° and the front leg pair approximately 95°. This

variation is however expected since the hind legs of the dung beetle play an important role in both locomotive and manipulative related tasks, where the front legs are mainly used for digging purposes (*Personal communication Prof. Dr. Gorb*).

In order to find the placement of the coxa three vectors are made by measuring the position of the BC- and CF-joints in 3D space for each of the coxae, again only on side of the dung beetle. These vectors represents both the placement and length of the coxa in a 3D space, where a plot of these vectors can be seen in figure 9 on page 8. An example of a 3D vector is shown by a green arrow on figure 10. This vector can be described by the two angles, α and β, which can be found by projecting the 3D vector onto both the xy- and xz-plane, resulting in the two dashed arrows. This approach is used on each of the three coxa vectors in order to get their placements via the two angles. The method for finding these angles is shown in equation 1 to 3.



(a) The coxa 3D vectors seen in the xy-plane

(b) The coxa 3D vectors seen in the yz-plane

Figure 9: The 3D vectors representing the coxae placements and lengths. The units on the figures are in millimeters. Note that only the coxa placement for the right side was measured and then mirrored due to the bilateral symmetry

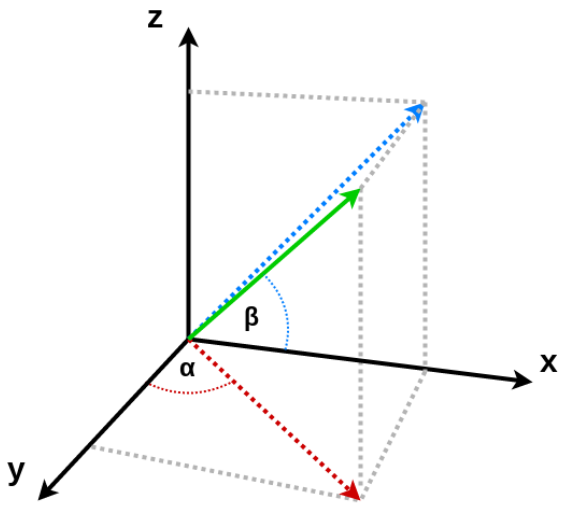


Figure 10: The green arrow the the 3D vector, and the dashed arrows is its projections onto both the xy- and xz-plane

The following is the 3D vector found by measuring the position of the BC- and CF-joints:

$$\vec{v} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1}$$

This vector is projected onto the xy and xz-plane, resulting in the following two vectors:

$$\begin{aligned} \vec{v}_{xy} &= \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \\ \vec{v}_{xz} &= \begin{bmatrix} x \\ 0 \\ z \end{bmatrix} \end{aligned} \tag{2}$$

It is now possible to calculate the angles from $\vec{v}_{xy}$ and $\vec{v}_{xz}$ to the unit vectors for the two axes x ($\vec{u}_x$) and y ($\vec{u}_y$), by using the equations in 3. The results are the two angles α and β, which are capable of describing the placement of the 3D vector, and thus the coxa.

$$\begin{aligned} \beta &= arccos\left(\frac{\vec{u_x} \cdot \vec{v}_{xy}}{\vec{u}_x \times \vec{v}_{xy}}\right) \\ \alpha &= arccos\left(\frac{\vec{u_y} \cdot \vec{v}_{xz}}{\vec{u}_y \times \vec{v}_{xz}}\right) \end{aligned} \tag{3}$$

The 3D vector length and thereby the length of the coxa is simply calculated by using equation 4.

$$|\vec{v}| = \sqrt{x^2 + y^2 + z^2} \tag{4}$$

The results from the coxa placement and length calculations are located in appendix B table 5. Notice that the remaining leg parts are attached to the coxa and thus only the position of the coxa is needed in order to place the legs.

Finally the extent of the measurement errors can be discussed. All of the measurements were carried out on two dead dung beetles, conserved in ethanol. This meant that the joints of the specimens were stiffer than normal and their internals were drier. This was especially apparent during the detachment of body parts, where the samples almost crumbled and withered. It was required to manually move the joints of the specimens, since it was dead during examination. This introduced yet another source of error, as the manual movement needed to be done with caution, in order not to overextend the joints to unnatural positions. The measurements were only done with a single specimen, without any comparisons to other dung beetles of the same species. This might give a slightly wrong result, as the anatomy of one dung beetle alone is expected to have some variance from the overall anatomy of the species. The measurements should however be fine for the purpose of this study, which mainly focuses on the kinematics.

All of the data gathered at the Zoological Institute of Kiel University is listed in appendix D and is ready to be used for modeling a dung beetle, which is done in the following section.

3 Modeling the dung beetle

In order to create a model of the dung beetle that can be simulated in software, a program called lpzrobots (LPZ) [16], based on the Open Dynamic Engine (ODE) [17], is used. The usage of LPZ is described in depth in appendix A, thus only the actual modeling is discussed in this section. The model is based on the data gathered in section 2 and will be composed of primitives, joints, actuators and sensors. These four essential parts are used for; representing the dung beetle visually, connecting separate body parts, moving the body, and sensing the environment.

The primitives are the core part of the model, as they act as the building blocks of the model and interacts with the world inside the simulation. Every primitive is instantiated with a dimension, mass and placement all determined from the real dung beetle, where the textures and colors are arbitrary chosen. LPZ uses its own units (LPZ-UNITS), that has an unknown scaling factor to the SI units. In order to convert LPZ-UNITS to SI units, the length and mass of the dung beetle model are scaled down, so that the model is one LPZ-UNITS long and one LPZ-UNITS of mass when simulated. This means that the simulation has one LPZ-UNITS of 106.4 mg when describing masses and one LPZ-UNITS of 18.6 mm when describing distance.

LPZ , unfortunately, has a very limited amount of primitives, resulting in involuntary large approximations. It is however important to note that this is mostly a visual approximation, that presumably does not effect the behavior and kinematics of the model. The most approximated body parts are the head, thorax and abdomen. The real dung beetle has an ellipsoid shape, which is not an option in LPZ and thus a box primitive is used. A slightly better approximation is made for the coxa, femur and tibia, which are all represented by cylindrical primitives. The real femur is somewhat flat resulting in a large radius of the femur primitive used in the simulation. In order to counter this the radius was scaled down to a visually pleasing size. The last part of the leg is the tarsus which, as described in section 2, consists of five individual parts. These parts are modelled as five consecutive cylindrical primitives. An illustration of the primitives used to model the dung beetle is shown in figure 11 and 12.

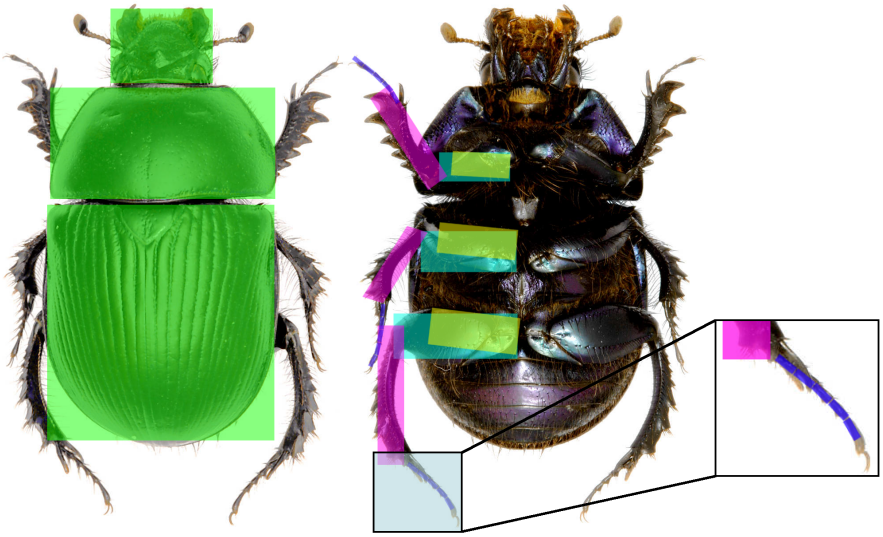


Figure 11: Top and bottom view of the dung beetle. The top view shows three light-green box primitives used to model the head, thorax and abdomen. The bottom view shows all of the leg parts. The five blue cylindrical primitives at the end of each leg are used to model tarsus parts. The pink cylindrical primitives are used to model the tibiae. The turquoise cylindrical primitives are used to model the femurs and the yellow cylindrical primitives, placed beneath the femur primitives, are used to model the coxae. Dung beetle picture from [14]

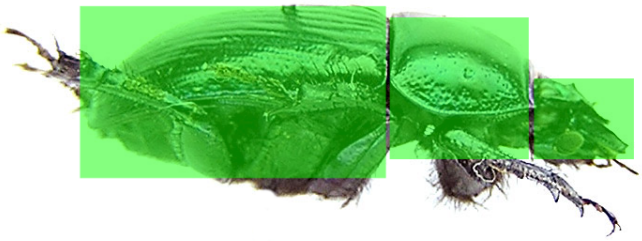
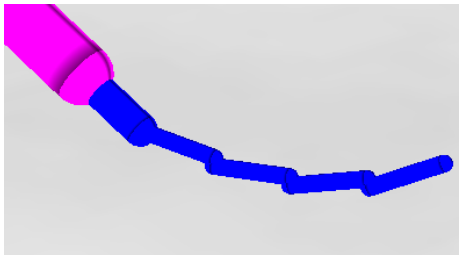


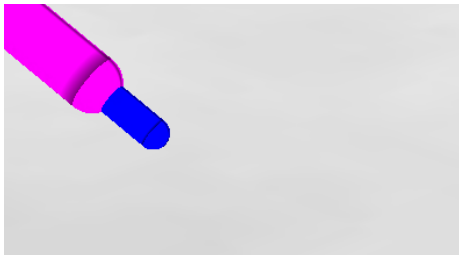
Figure 12: Side view of the dung beetle. The side view shows three light-green box primitives used to model the head, thorax and abdomen. Dung beetle picture from [18]

The dung beetle has a total of 50 joints, three on each leg including five on each tarsus and two on the main body. All 50 joints are necessary for the model, as they besides working as normal joints also links the different body parts, and thereby the primitives, together. Each joint is instantiated with parameters specifying their position, type and rotational range of movement. These parameters are chosen based on the measurements obtained in the previous section. The fact that each tarsus consists of five joints and five primitives requires a lot of processing power while running the simulation. Thus the tarsus is made optional in order to avoid heavy CPU load. Moreover this can be justified

since the tarsus is most actively used for manipulation related tasks, where the adhesion generated between the dung ball and the tarsus is very important [19]. The tarsus is more passive in locomotive tasks, where it is simply laying flat on the walking surface [19]. When tarsus is deselected, only the first part of the tarsus, which is in extension of the tibia, is instantiated. The model with and without tarsi is shown in figure 13.



(a) With the tarsus. All five primitives of the tarsus are created

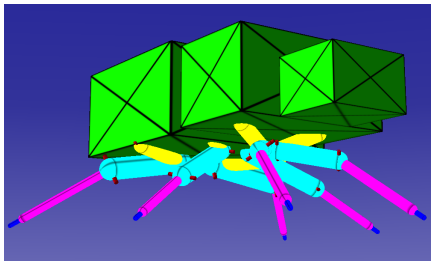


(b) Without the tarsus. Only the first primitive of the tarsus is created

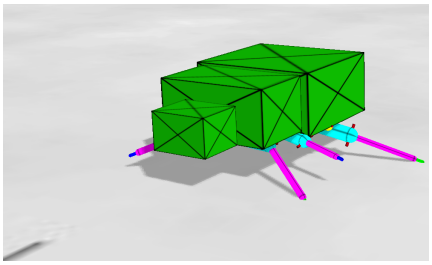
Figure 13: Leg with and without tarsus. The purple primitive is the tibia, where the blue primitives are the different tarsus parts

Finally the actuators and sensors, which makes it possible to control the different joints, are instantiated. Each joint is assigned both an actuator for controlling it and a sensor for reading its position. Additionally ground contact sensors are placed on each tarsus primitive, indicating whether a leg has ground contact or not. If the tarsus is deselected, the ground contact sensor will only be placed on first tarsus part.

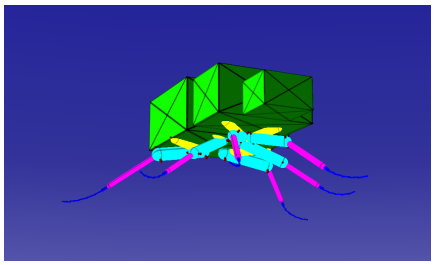
Figure 14 shows the final model created in LPZ. Additional picutres of the model can be found in appendix C. Note that the coxa, the yellow leg part, extends into the body of the beetle. This is due to the shape of the body, which on the real dung beetle is curved so the coxa is aligned with the body.



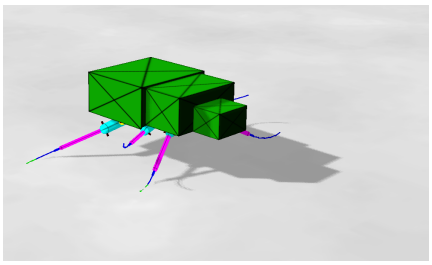
(a) Without tarsi, fixed in the air



(b) Without tarsi, walking on the ground



(c) With tarsi, fixed in the air



(d) With tarsi, walking on the ground

Figure 14: Illustration of the dung beetle model in various positions. Following colors are used for the different parts of the beetle: Green = Head, Thorax and Abdomen. Yellow = Coxae. Turquoise = Femurs. Pink = Tibiae. Blue = Tarsi parts. Red = joints. A light green color of the tarsi is an indication of ground contact

Besides extending into the body of the model, the coxa is also the part that causes the biggest kinematic difference from the commonly used hexapod leg explained in the introduction. On the common hexapod leg the BC-joints are limited to horizontal movement of the leg endpoint. The same applies to the femur and tibia (CF- and FT-joint), which on the common hexapod leg is limited to vertical movement of the leg endpoint. This is however not the case for the dung beetle model, where the BC-joint moves the leg endpoint both vertically and horizontally. When the BC-joint moves from one extreme position to its other extreme position (shown in figure 15), the rest of the leg is moved in a parabolic trajectory.

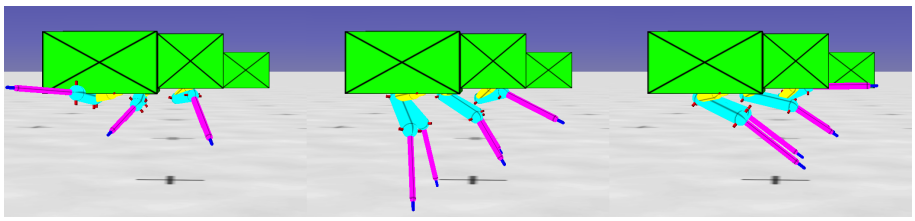


Figure 15: The BC-joints moves between its extreme positions, while the CF- and FT-joints are kept fixed

During this section it has been shown that it is possible to make a model for a dung beetle without using excessive approximations. The following section will furthermore show how to equip this model with a controller, utilizing the complex kinematics of the model.

4 Locomotion control

Locomotion can seem rather trivial to us humans, as no one really has to think consciously about how to move each leg joint when walking [20]. At times it may appear like the movement is handled by each individual leg, so that we can focus on other things while walking. This thought is the reality of the dung beetles, as their small brains prevent them from having a centralized locomotion control, which requires simultaneous control of up to at least 18 joints (*Personal communication with Dr. Emily Baird from Lund University*).

Embodied artificial intelligence (embodied AI) and Good Old-Fashioned Artificial Intelligence (GOFAI) are the two main approaches for solving the locomotion control task. It is preferable that the control and locomotion of the model is biologically-inspired, so it fits accordingly to the biologically-inspired model. Having this in mind utilizing GOFAI as a control method is counter intuitive, since its design philosophy is based on the robot knowing the configuration of the world around it. Another disadvantage of GOFAI is the calculations needed for the precise movement of the robot [21]. The calculations required to control a robot with 18 degrees of freedom like in the locomotion of a hexapod would demand a lot of processing power, and like the small dung beetle brain, most processors would not be able to handle this.

The control architecture of embodied AI is based on the robot knowing only what it can sense [22]. Embodied AI is decentralized and it is therefore possible to layer and split the controller into several smaller controllers, also known as modules. These small controllers are supposed to work on a set simple rules, which requires less processing power. So instead of defining the complex behavior directly, like in GOFAI, the behavior is expected to emerge from the different controllers all following some simple rules. By that the robot needs to have a physical presence in the environment, as the world is said to be its own best model [22]. The result is that instead of making imperfect models where information is missing, use sensors to measure the environment of the robot. With all this in mind, embodied AI seems to be the best solution for simulating dung beetle behavior.

4.1 Walknet

In 1998 a paper by Dr. Holk Cruse [23] was released which introduced a control system called WALKNET. The control system is inspired by experiments with stick insects and is able to control the locomotion of a hexapod [24]. WALKNET is based on the principles of embodied AI and it uses a decentralized modular architecture, which shows from the fact that each leg has its own controller (module) working on sensory inputs. An advantage of

WALKNET is that it does not use predefined walking gaits as many other control systems. Instead the gait emerges from the fact that each leg-controller has the capacity to decide the state it should be in, by following six basic coordination rules [6]. Although these rules are originally derived from a stick insect, they will be used on the model of the dung beetle. The hypothesis is that the behavior of the dung beetle will emerge from the same rules, due to the dung beetle kinematics of the model.

The WALKNET controller is mainly used for locomotion, which call for an extension in order to gain control of a dung ball. A way to achieve this extension is by adding additional behavioral layers to the WALKNET controller. This is possible because the WALKNET controller is build on the principles of the embodied AI approach called subsumption architecture. The idea behind subsumption architecture is to use simple rules divided into different layers together with signals from the robot's sensors. From these layers of rules, all grounded in its physical interactions, complex behaviors are expected to emerge, where the complex behavior in this study is the behavior of the dung beetle [21, 22]. The additional layers used for manipulating dung balls are however not investigated in this study and are left for future study. Instead different positions of the dung beetle model are tested in section 5, to show if the previously rather impossible positions of the approximated insect model, is now possible with the accurate dung beetle model.

The leg-controllers mentioned above can be in one of two states, representing the two micro behaviors swing and stance phase [24]:

- The leg is protracting (*swing phase*): The leg is lifted off the ground in its posterior extreme position (PEP). It then *swings* to its anterior extreme position (AEP), where it yet again obtains ground contact (figure 16).
- The leg is retracting (*stance phase*): After obtaining ground contact at the anterior extreme position the leg has to move back to its posterior extreme position, while maintaining ground contact. In this way the body is being propelled to the front, resulting in the animal moving forward (figure 16).

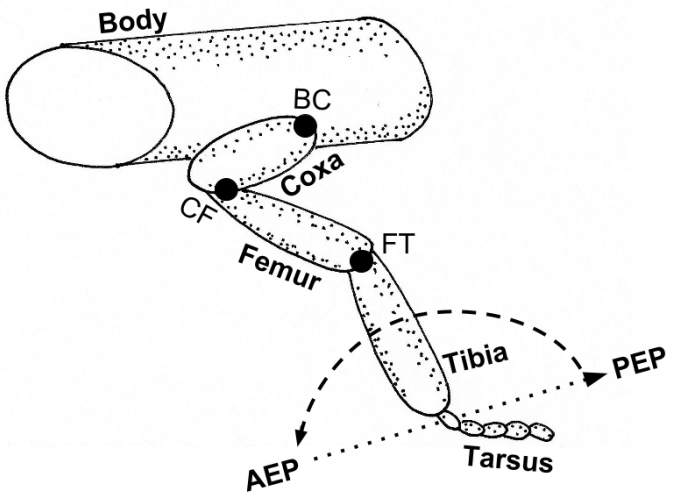


Figure 16: Illustration of the posterior extreme position (PEP) and anterior extreme position (AEP). The dotted line indicates the stance path, where the dashed line indicates the swing path. See figure 6 on page 6 for orientations of the three joints.

The control of the legs during either swing or stance phase is achieved through two low-level control networks called stance net and swing net. The purpose of these networks are to generate trajectories for the legs to follow. The networks are often implemented as neural networks, generating velocity commands for the joint actuators, as represented by the work of V.Dürr et al. [20], H.Cruse et al. [23], P.Arena et al. [25] and others. However a specific implementation is not required as there are numerous ways of generating leg trajectories.

It is not possible for a leg to be in both swing and stance phase at the same time, so a mechanism must be in place to decide which of the two networks should control the actuators of the different legs. This mechanism is implemented in a network called the selector net, which essentially chooses if either the swing net or stance net should have actuator control. The selection is based on ground contact sensors, position sensors that observe whether the leg has reached its posterior extreme position or not, and the six coordination rules derived from experiments with the stick insect [23]. The six rules, which are the corner stone of the WALKNET behavior, act as a coupling between the legs [20]. Each rule can be viewed as an information channel, by which a leg signals its current state to its neighbors. This means that the front and hind legs pass information to two neighboring legs, while the middle legs send to three neighboring legs [20].

There are many interpretations of the six coordination rules and in order to explain the

rules in the most general and intuitive way, descriptions from different articles ([20], [23] and [24]) are merged and rewritten. The resulting explanations of the six coordination rules are listed below.

Suppress lift-off rule (rule 1)**:**

A leg in swing phase, i.e when it is lifted off the ground, suppresses lift-off, and thereby the swing phase, for the neighboring leg in front. This rule avoids potentially harmful situations of static instability for a hexapod that must not fall over.

Facilitate early swing phase rule (rule 2)**:**

A leg experiencing a touch-down, i.e end of an swing phase, facilitates lift-off/swing phase for the neighboring legs in front and besides the leg. This favors a temporal cohesion.

Enforce late swing phase rule (rule 3)**:**

A leg in late stance, i.e approaching its posterior extreme position, facilitates lift-off/swing phase for the neighboring legs behind and besides the leg. This is causing the two neighboring legs to 'catch up' in order to maintain the temporal cohesion.

AEP determination rule (rule 4)**:**

A leg's anterior extreme position, i.e the position where its swing phase ends, is determined by the current position of the neighboring leg in front. This causes a leg to exploit the foothold close to where the neighboring leg in front is already standing.

Force distribution rule (rule 5)**:**

A leg with increased load causes neighboring legs to prolong their stance phase. This effectively shares the load between neighboring legs.

Avoid treading-on-tarsus rule (rule 6)**:**

A leg that is trod-on by the neighboring leg behind itself, sends it a signal telling it to step off. This solves situations where a leg steps on another leg's tarsus.

Rules 1-3 all affect the timing of the transitions between the stance and swing phase [20]. This is done by manipulating the posterior extreme position input-signal to the selector net, which thereby either shortens or prolongs the stance phase. Rule 1 manipulates the posterior extreme position signal by relocating the posterior extreme position further away, thus prolonging the stance phase, where rule 2 and 3 does the exact opposite [24]. Figure 17 shows how the three rules act on the different legs. Unlike rule 1-3, rule 4-6 handles specific situations that does not occur periodically. These three rules are mainly used in non-ideal situations where external influence such as rough terrain are present.

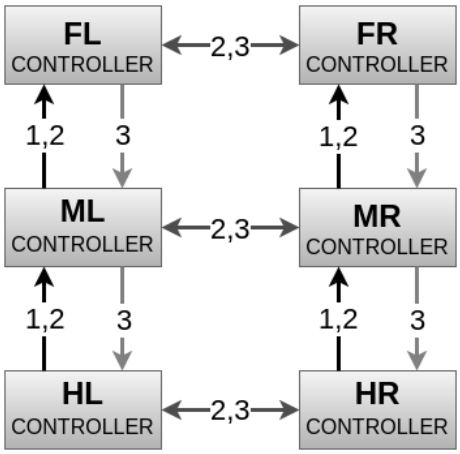


Figure 17: Rule 1, 2 and 3 acting between the different leg controllers. Legs are abbreviated by F(*front*), M(*middle*), and H(*hind*) as well as L(*left*) and R(*right*)

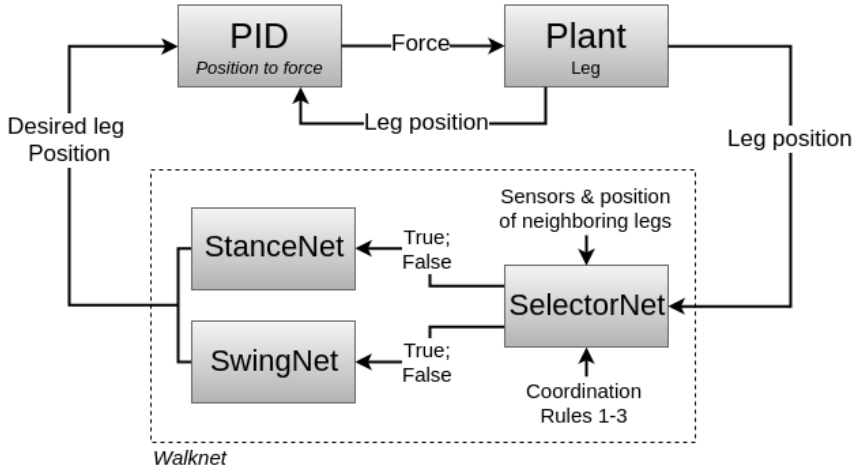


Figure 18: Block diagram of the implemented control system. The plant consists of the actuators located in each leg, with a PID assigned for each actuator

4.2 Implementation of Walknet

Figure 18 shows a block diagram of the WALKNET implementation, used in this study. As seen on the figure, the output from both the stance and swing net is leg positions, each consisting of three joint angles for the BC-, CF-, and FT-joint. The reason for this choice is that it is easier and more intuitive to work with positions rather than velocities. This is especially reflected in the stance and swing net, where the neural networks from Cruse's original system are replaced with simple state machines. Another reason is the low coupling in LPZ, which makes the simulation time difficult to access from the controller class. The PID controller used for position control requires the time, which is why it cannot be implemented in the controller class. It is therefore only possible to use the actuators as either position controlled or velocity controlled, due to the fact that the PID needs to

be implemented in the actuator class where the time variable is known. This problem is elaborated further in appendix A.

Figure 18 also shows that only coordination rules 1 to 3 are implemented. Rule 4 is omitted as the simulated dung beetle model is expected to have foothold at all times, where rule 5 is left out due to the assumption of a constant load on all of the six legs. Rule 5 is most commonly used in situations where the hexapod is expected to either lose a leg or to have a leg fixated in a constant position. Moreover, since it is possible to ensure that the tarsus of the different legs will not overlap by setting a constant posterior extreme position and anterior extreme position, rule 6 is also omitted. A last modification is the implementation of rule 1 between the two hind legs. This is required because nothing is stopping the hind legs from automatically lifting off the ground, when they arrive at their posterior extreme positions, before the other hind leg has ended its swing phase. Thus without rule 1 acting between them, situations where both hind legs are lifted simultaneously may occur.

The rest of the implementation, besides the modification described above, is largely based on Cruse's system [23]. This system consists of independent networks (stance and swing net) that generate the swing and stance trajectories and a third network (stance net) that chooses which of the two trajectories to use. Like in Cruse's system this implementation also utilizes boolean signals from the ground contact sensor, the rules, and the posterior extreme position sensor (figure 19).

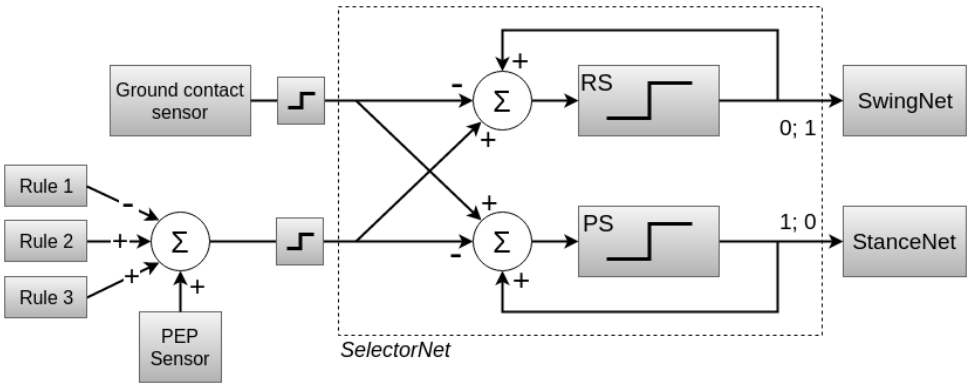


Figure 19: Block diagram of implemented selector network. Notice that the network only uses boolean values in order to choose between the stance- and swingNet. RS is short for Return Stroke aka swing phase and PS is short for Power Stroke aka stance phase

The next three sections describe how each individual network is implemented. The stance and swing networks have multiple implementations, because it is desired to find the best solution for trajectory generation based on position control.

4.2.1 The selector network

The selector network (shown on figure 19) consist of a swing phase/return stroke (RS) unit and stance phase/power stroke (PS) unit, both having binary output of either 0 or 1 (false or true). The units are enforced with positive feedback that stabilizes the current state of the network. The inputs to the network are the ground contact sensor, signaling if a leg has ground contact or not, and the posterior extreme position sensor, signaling if a leg is in its posterior extreme position or not. Remember that the posterior extreme position signal might be prolonged or shortened by the three coordination rules. The output from the selector net dictates whether the stance or swing network should drive the leg.

4.2.2 The stance network

As mentioned in section 3, the biggest difference in having an anatomically correct modeled dung beetle leg, is that the movement of the BC-joint results in both a horizontal and vertical movement of the leg endpoint, instead of pure horizontal movement as seen in the typically used hexapod leg. This could have a rather large impact on the stability of the dung beetle models body, as the vertical displacement may cause it to 'bounce' while walking.

Two versions of the stance network have been made, both implemented as state machines with outputs in form of a leg position (three joint positions). State machines for both versions can be seen in figure 20 and 21, where the generated trajectory for the leg endpoint is shown in figure 22.

Version 1 :

This version is very simple and it consists of only two states. In the first state, called `TO_PEP_STANCE`, the leg is moved from its current position to its posterior extreme position. The state changes to `IDLE`, when the leg it located in its posterior extreme position and stays here until the selector net selects the stance net yet again.

Version 2 :

This version is an expansion of version 1 and it consists of three states. The extra state, called `TO_MID_STACE`, aims at fixing the vertical displacement problem. The new state is placed between the states `TO_PEP_STANCE` and `IDLE`, which is analogous to adding an extra point between anterior extreme position and posterior extreme position. It is possible to eliminate the vertical displacement problem by assigning the 'midpoint' a vertical position equal to the opposite vertical displacement of the leg during a normal stance phase with version 1, which is seen on figure 22b.

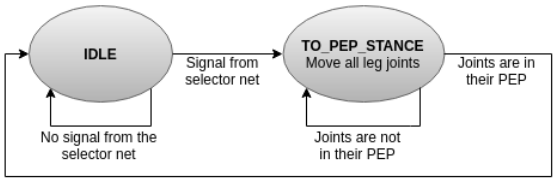


Figure 20: Stance net version 1

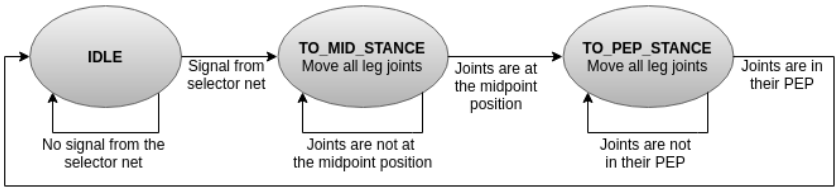


Figure 21: Stance net version 2

(a) Trajectory generated by stance net version 1, for the leg endpoint

(b) Trajectory generated by stance net version 2, for the leg endpoint

Figure 22: Illustration of the trajectory for the leg endpoints. The green circle is the posterior extreme position and red the anterior extreme position. The gray circle indicates both a position on the trajectory and a change of state

4.2.3 The swing network

The vertical displacement problem also affects the implementation of the swing net. The impact is however minimal, as it only causes the leg to swing lower as it approached the midpoint between the posterior extreme position and the anterior extreme position.

Two versions of the swing network have been made, both implemented as state machines with outputs in form of a leg position (three joint positions). State machines for both versions can be seen in figure 23 and 24, where the trajectory generated for the leg endpoint is shown in figure 25.

Version 1 :

This version consists of four states. In the first state, called `LIFT`, the leg is lifted of the ground to a predefined height, thus only the CF- and FT-joint is used. The next state, called `TO_AEP_SWING`, then swings the leg from the current position to a position above the anterior extreme position, by using only the BC-joint. The penultimate state, called `LOWER`, then lowers the leg until it senses ground contact

and ends the swing phase. The last phase is called `IDLE`, where it stays until the selector net selects the swing net yet again.

Version 2 :

This version consists of three states. The idea behind this method is to generate a trajectory by using different velocities in the three leg joints. In the first state, called `TO_AEP_SWING`, the BC-joint moves towards the anterior extreme position, and the CF- and FT-joint moves up with a positive velocity. When the BC-joint passes the midpoint between the posterior extreme position and anterior extreme position the velocity of the CF- and FT-joint changes sign and the leg is then lowered. The state changes when the BC-joint arrives at the anterior extreme position. The new state, called `LOWER`, then lowers the leg until it senses ground contact and ends the swing phase by changing the state to `IDLE`.

In this way it is then possible to create a parabolic swing trajectory by letting the CF- and FT-joint move faster than the BC-joint. One thing to mention is that the output from this network is still positions, thus the velocity is generated by incrementing the target position by some constant depending on the desired velocity.

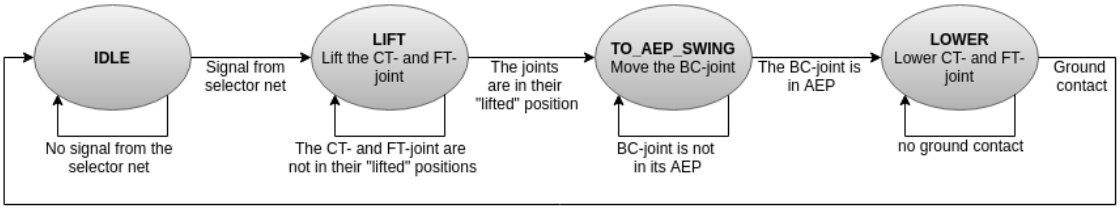


Figure 23: Swing net version 1

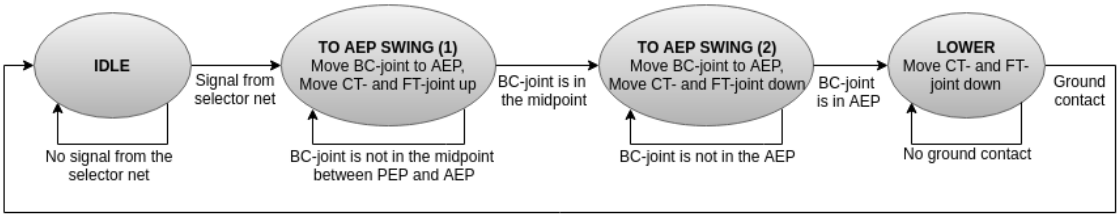
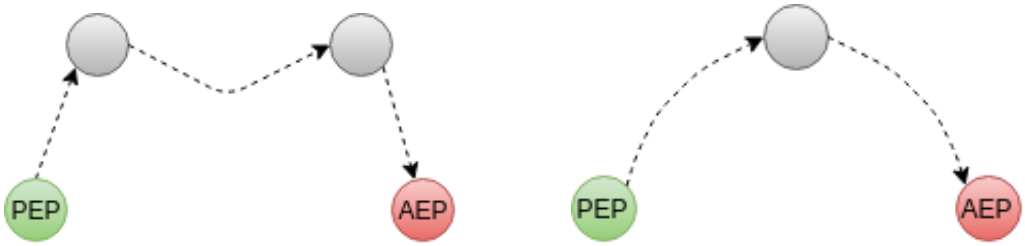


Figure 24: Swing net version 2



(a) Trajectory by swing net version 1 (b) Trajectory by swing net version 2

Figure 25: Illustration of the trajectory for the leg endpoint. The green circle is the posterior extreme position and red the anterior extreme position. The gray circles indicates both a position on the trajectory and a change of state

All of the versions described above will be tested and discussed in the following sections.

5 Experiments and results

It was hypothesised that the WALKNET controller would make the dung beetle model walk with a gait behavior comparable to that of the real dung beetle, even though it is developed based on a stick insect. Several implementations of the swing and stance net were proposed in the previous section. It is therefore necessary to test all of the different combinations to see if a gait emerges and which combination is the fastest, most stable, and most accurate walking implementation of the WALKNET controller. The abbreviation for the swing and stance net combinations are shown in the following list.

11: Swing net 1 and stance net 1

12: Swing net 1 and stance net 2

21: Swing net 2 and stance net 1

22: Swing net 2 and stance net 2

The different combinations will be tested in both a height test for stability, and a walking test for speed and walking accuracy. These tests will run for 300 seconds each, in which the dung beetle model enters two states. At the beginning of the test the dung beetle model will be in a transient state, until it later enters the steady state both shown in figure 26. During the transient state the dung beetle model is trying to find a proper gait, causing swerving and unpredictable movements. When a stable gait is established it enters the steady state. These tests will not be concerning the transient state and thus the first five seconds of sensory data received from the dung beetle model is neglected. The sensory data includes 3D Cartesian positions for each part of the dung beetle model, as well as boolean signals from the ground contact sensor. All the data is processed using MATLAB.

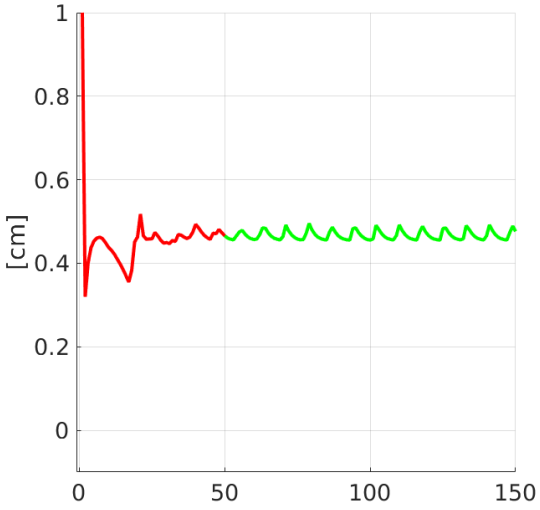


Figure 26: Plot of the height vs samples. Here it can be seen that it takes roughly 50 data samples before the dung beetle model is past the transient state (red) and is in a steady state (green)

An example of the data received, when the dung beetle model is simulated for 300 seconds and the transient state is removed, is shown in figure 27a. In order to easily compare the different tests, the data is translated so that the start position of the thorax is at $x = 0$, $y = 0$ and is rotated down on the x-axis so that the last data point for the thorax has $y = 0$.

The translation is done by the use of standard vector calculations where the new position of the data points are given by going through all the data with equation 5. The vector $\vec{v}_{new}$ holds the new data points after the translation. The data points subscripted *old* are the coordinates of the data points before the translation, where the coordinate subscripted *transient* is the last coordinate of the data points in the transient state. By doing this subtraction all the data points are moved back so that the steady state starts at $x = 0$ and $y = 0$.

$$\vec{v}_{new} = \begin{bmatrix} x_{old} \\ y_{old} \\ z_{old} \end{bmatrix} - \begin{bmatrix} x_{transient} \\ y_{transient} \\ 0 \end{bmatrix} \tag{5}$$

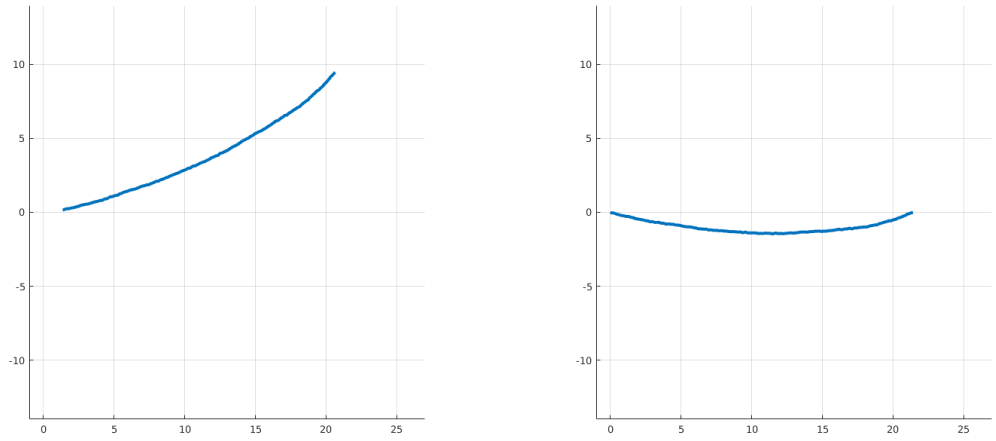
The rotation of the last point is done by creating a vector $\vec{v}$ from origin to the last data point. The angle between $\vec{v}$ and the x-axis is then found with equation 6, where $\vec{u}_x$ is the unit vector for the x-axis. The amount of rotation on the last data point should then be

applied to the rest of the data set. For this purpose each data point is multiplied with a rotational matrix as shown in equation 7.

$$r = -arccos\left(\frac{\vec{u}_x \cdot \vec{v}}{\vec{u}_x \times \vec{v}}\right) \tag{6}$$

$$\begin{bmatrix} x_{new} \\ y_{new} \\ z_{new} \end{bmatrix}^T = \begin{bmatrix} x_{old} \\ y_{old} \\ z_{old} \end{bmatrix}^T \cdot \begin{bmatrix} cos(r) & -sin(r) & 0 \\ sin(r) & cos(r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

An example the translated and rotated data can be seen in figure 27b.



(a) Walking test plotted before rotation and translation of the data

(b) Walking test plotted after rotation and translation of the data

Figure 27: Plot of the data received from the dung beetle model when walking, before and after the data have been processed

The data is now ready to be used in the following tests.

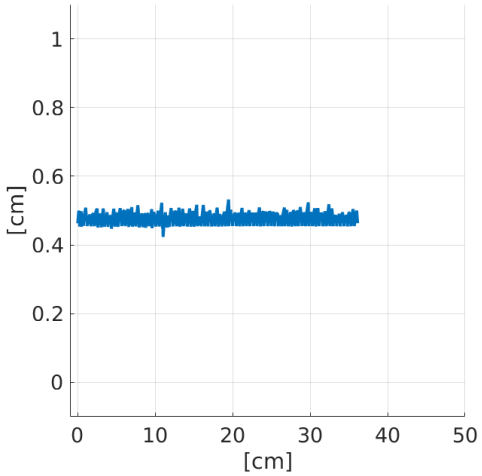
5.1 Test - Stability of height

As written in section 2 the dung beetle walks with a tripod gait, which can be seen in a video [13] by *kekPafrany*. This video also shows that the dung beetle has a small variance in its height while walking. The purpose of this test is therefore to see how much the dung beetle models height deviates while walking and if it is acceptable or not.

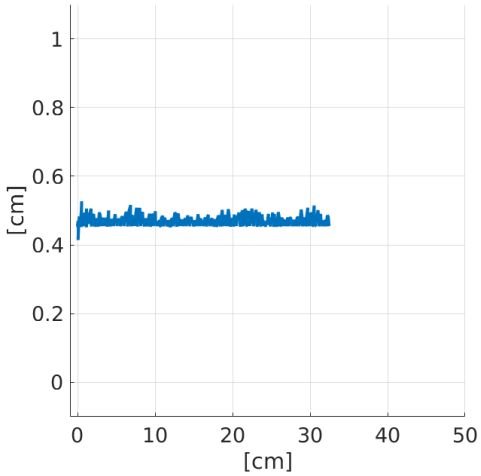
The test is repeated five times for each combination of the swing and stance net, from where both a mean height and variance are determined. The standard and max deviation

is also noted to check for outliers.

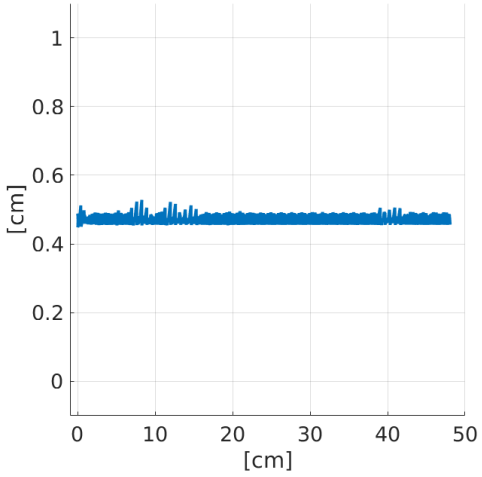
The test results are shown in figure 28. These only include one trial for each combination, where the figures for the rest of the trials can be found in the appendix D.1.



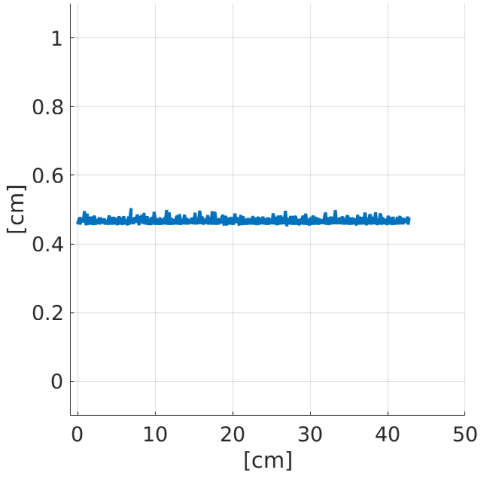
(a) Height test - swing net 1 and stance net 1



(b) Height test - swing net 1 and stance net 2



(c) Height test - swing net 2 and stance net 1



(d) Height test - swing net 2 and stance net 2

Figure 28: Plot of the four combination tested. The plot includes only one trial for each combination

The mean, variance, standard deviation and max deviation for the trials in the figure, are shown in table 1. The tables for the rest of the trials also found in appendix D.1.

Combination	**11**	**12**	**21**	**22**
mean [cm]	0.4656	0.4631	0.4678	0.4651
variance	0.0002	0.0001	0.0001	0.0001
st.d [cm]	0.0145	0.0108	0.0119	0.0086
max deviation [cm]	0.5321	0.5160	0.5217	0.5120

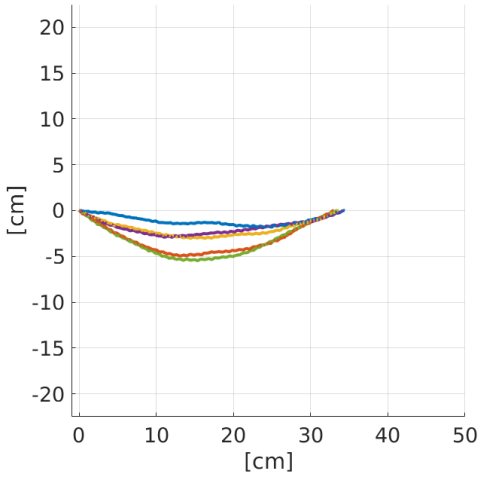
Table 1

5.2 Test - Walking capabilities

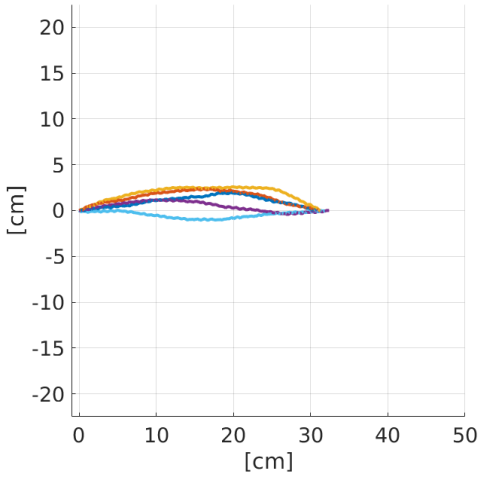
The implemented WALKNET controller does not compensate for swerving of the dung beetle model while walking. It is however expected that the model is able to keep a fairly straight walking direction, which is therefore tested here. This test is also repeated five times, once for each combination of the swing and stance net, in order to find the combination that walks the straightest.

A mean value is calculated around the x-axis for each test. The smaller the mean is, the closer the path of the model is to the x-axis. A low variance also indicates that the dung beetle model is primarily walking in a straight line and is not swerving from side to side around the x-axises, where the max deviation shows how far the dung beetle model has moved to one of the sides during the test. The walking distance of the dung beetle model is found from origin to the x-value of the last data point.

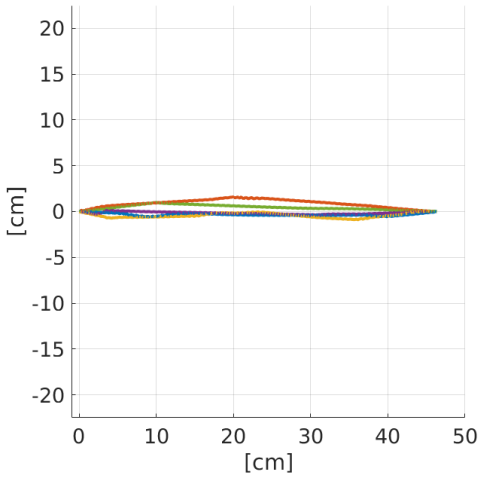
The test results are shown in figure 29. These figures include all five trials for each swing and stance net combination.



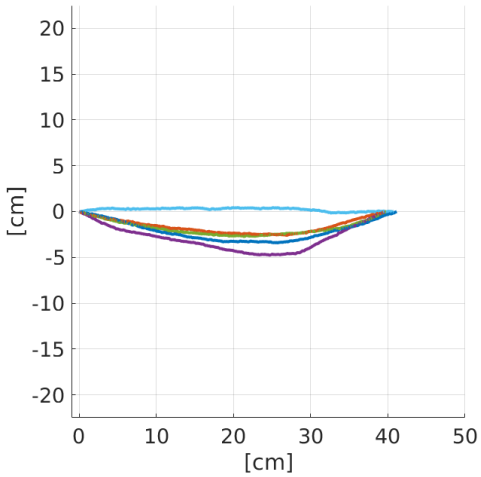
(a) Walking test - swing net 1 and stance net 1



(b) Walking test - swing net 1 and stance net 2



(c) Walking test - swing net 2 and stance net 1



(d) Walking test - swing net 2 and stance net 2

Figure 29: Plot of the four combination tested. The plot includes all five trials for each combination

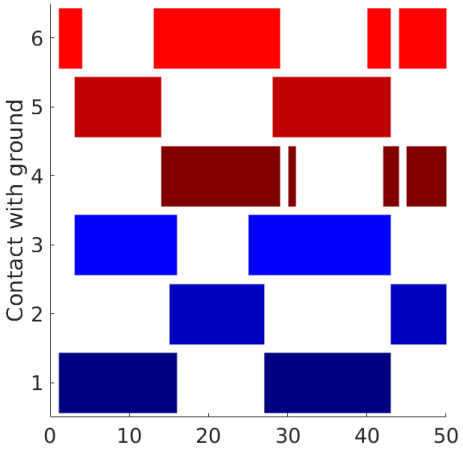
The mean, variance, standard deviation, max deviation and walking distance for one trial of each combination, are shown in table 2. The tables for the rest of the trials are found in appendix D.2.

Combination	**11**	**12**	**21**	**22**
mean [cm]	-1.0855	1.0114	-0.3300	-2.1015
variance	0.2796	0.4983	0.0180	1.1539
st.d [cm]	0.5288	0.7059	0.1340	1.0742
max deviation [cm]	1.8218	2.3688	0.5822	3.4037
distance [cm]	34.4144	31.3660	46.3417	41.1332

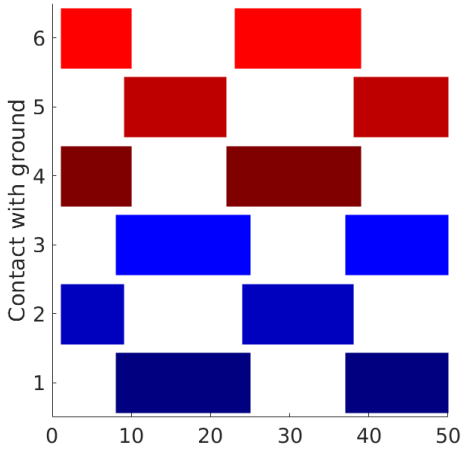
Table 2

5.3 Test - Stability of gait map

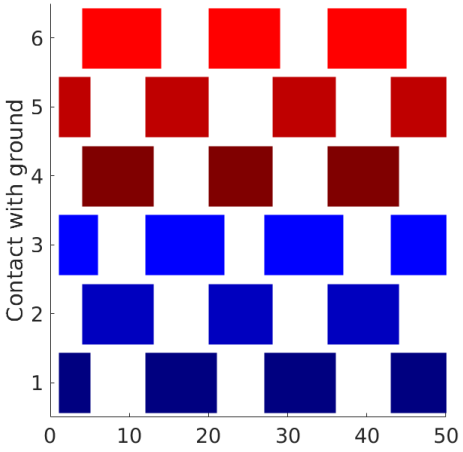
The dung beetle uses as mentioned a tripod gait. The dung beetle model therefore needs to follow a tripod gait after the transient state, if the WALKNET controller is to be biologically plausible. The dung beetle model is equipped with boolean contact sensors at the tarsus, which is necessary when generating gait maps, that can visually show which gait is used. The plots in figure 30 shows the gait map for each combination of the stance and swing net, which all shows that the dung beetle model follows the tripod gait. This test is only repeated once for each combination of the swing and stance net, as the same gait map always emerges when the dung beetle model is in its stable state.



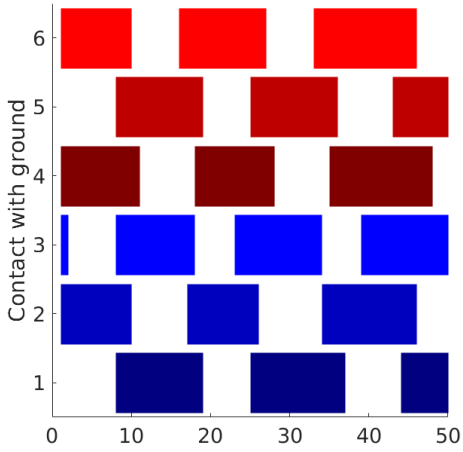
(a) Gait map - swing net 1 and stance net 1



(b) Gait map - swing net 1 and stance net 2



(c) Gait map - swing net 2 and stance net 1



(d) Gait map - swing net 2 and stance net 2

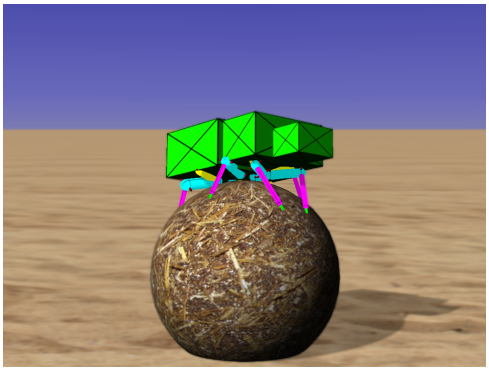
Figure 30: Plot of the gait map for the four combination. The red colors represents ground contact sensors on the left side, and blue on the right side. Each color is sorted so that the top is the front leg, then the middle leg and finally the hind leg

5.4 Test - Poses

Although the additional layers used for manipulating dung balls are not investigated in this study, the possibility of adding them needs to be tested. Two of the common dung beetle tasks are tested, one where the dung beetle is on top of a dung ball (figure 31a) and another where the dung beetle rolls a dung ball (figure 31c).

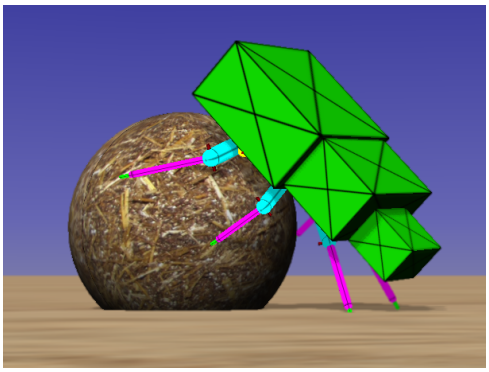
In order to replicate these tasks with the dung beetle model, both the position of the model and the configuration of the legs need to be determined. This is done by manually setting the position of all the joints and by placing the dung beetle model in a suitable position. The resulting replications of the dung beetle tasks are shown in figure 31b and 31d, where it can be seen that the more accurate dung beetle kinematics makes it possible for the model to manipulate dung balls.

(a) A real dung beetle on top of a dung ball



(b) The dung beetle model on top of a dung ball

(c) A real dung beetle ready to roll a dung ball



(d) The dung beetle model ready to roll a dung ball

Figure 31: A real dung beetle and the dung beetle model performing the same tasks

6 Discussion

The model proposed in this study is based on measurements of the dung beetle species *Geotrupes stercorarius*, which does not roll dung balls, but buries them right away. It can be discussed whether it is adequate to use a species with a limited amount of manipulation related tasks. The kinematics should however be more or less the same for every dung beetle species and only the dimensions might differ as suggested by Prof. Dr. Stanislav N. Gorb. One could imagine that the hind legs of a species that roll dung balls might be longer, which is a hypothesis based partially on pictures of the African dung beetle that rolls and crawls on top of dung balls.

The dung beetle model relies on measurements of a single dead dung beetle conserved in ethanol at the Zoological Institute of Kiel University. This meant that the joints of the specimens were stiffer than normal and the internals were drier. This was especially apparent during the detachment of body parts, where the samples almost crumbled and withered. This forced the measurements to rely on videos of living dung beetles as well as manual movement of the dead dung beetles joints, in order to see how the different joints and parts behaved. A source of error in this approach is the fact that the manual movement needed to be done with caution, in order not to overextend the joints to unnatural positions. Another problem is the fact that the measurements rely on a single specimen, without any comparisons to other dung beetles of the same species. This might give a slightly wrong result, as the anatomy of one dung beetle alone is expected to have some variance from the overall anatomy of the species. The measurements should however be sufficient for the purpose of this study, as the main focus is the kinematics of the beetle, which, as said earlier, are expected to be approximately the same for all dung beetles species. However measurements including living dung beetles might give an even more accurate model, that could also result in new discoveries.

There are various differences when dealing with an accurate dung beetle model that includes as few approximations as possible in contrast to many of today's biologically-inspired hexapods. The largest difference is found in the leg configuration, where both the anatomy and kinematics are considerably more complex than most approximated ones. All the legs are no longer exact copies of each other and are instead split into three identical pairs, the hind, middle and front legs. Another difference as a result from the added complexity is the parabolic trajectory generated at the leg endpoint, when moving the BC-joint, which on many approximated models only causes horizontal movement of the endpoint. This difference was expected to have a negative impact on the stability and velocity of the

model, as it was hypothesised to cause instability when a leg is both moving its BC-joint and has ground contact (e.g. in the stance phase). This was surprisingly not the case, as tests revealed that it in fact resulted in a straighter and longer walk, with minimal change in the stability of the model. Even though the reason for this improvement might be unclear, due to missing observations of a living dung beetle, it is important to remember that nature has worked on perfecting itself for thousands of years and that this might be one of those perfections. Thus the real dung beetle has evolved to get the best locomotion, which includes the parabolic movement of the leg endpoints.

Tarsus is more or less neglected in this study, due to its minimal usage in locomotion. It could be argued if this is a fair approximation, when even the smallest thing can have a big influence on the end result, as seen with the parabolic trajectory of the leg endpoint. The need of the tarsus is described by Gorb et al. [19], who emphasizes that the main goal of the tarsus is to generate as much adhesion as possible. This is not needed in the simulation as the walking surface is ideal for walking and because the dung beetle model is only tested in locomotive tasks. The tarsus is however indispensable if the additional behavioral layers for manipulating dung balls is to be implemented. In manipulative tasks the tarsus swings around objects to get a higher adhesion and better control of the object (e.g a dung ball) [19].

The dung beetle model is implemented with the WALKNET controller, due to the fact that it is build on the principles of embodied AI, with a subsumption architecture. This architecture enables the controller to be expanded so that it in the future can handle manipulative tasks as well. WALKNET is also a somewhat complex controller, due to its decentralized and modular architecture where the locomotive behavior is expected to emerge from the kinematics of the model and the simple rules. This, coupled with the complex kinematics of the dung beetle, makes it a considerably difficult controller to implement. Therefore to ease the implementation, it was chosen to implement WALKNET as a series of networks working on positions, instead of velocities which is more commonly used. The velocity based swing and stance networks are often implemented as neural networks. It is therefore easier to implement the position based networks, where it is possible to use less complex state machines.

However there exist drawbacks when using a position based WALKNET controller, since it is difficult to do any conclusions and tests involving the velocity. By altering the velocity of the dung beetle model, different gaits are expected to emerge [26]. The tetrapod gait should presumably emerge at low velocities and the tripod gait at high velocities. This

test is difficult to perform when the actuators are using PID controllers on the position and not the velocity. The position based WALKNET controller and the fact that all legs are no longer equal in size also calls for additional manual fine adjustment of the trajectories used in the swing and stance networks. This is because the duration of the swing phase is expected to be equal for all of the legs, which is hard to fulfill with a position based WALKNET controller, where the velocity depends on the parameters of the PID controller. This task may be easier with a velocity based WALKNET controller, where it is possible to alter the velocity directly to adjust the duration of the swing phases so that they are equal for all the legs.

An adjustment to the WALKNET controller is the implementation of rule 1 between the hind legs. This might also be a consequence of the position based WALKNET controller and the anatomy of the legs. If the stance phase is shorter than the swing phase, the hind legs will lift before the swing phase of the opposite leg has finished, as no rule 1 is acting on the hind leg. Again a velocity based WALKNET controller could be used to adjust the duration of the stance or swing phases, so that this would not happen.

7 Conclusion

This study focused on the effects of having an accurate kinematic model of a dung beetle. The kinematics, mass and dimensions for the model are derived from a dead dung beetle conserved in ethanol of the species *Geotrupes stercorarius*. The fact that *Geotrupes stercorarius* does not roll dung balls in nature is neglected, as the interspecies kinematics are presumably similar. The kinematics measured is found to be more complex than the kinematics commonly used in hexapods. The primary kinematic difference between the common hexapods and the dung beetle model is the fact that, the dung beetle model have pairwise identical legs due to bilateral symmetry, which means that all the legs are no longer of equal size. Another difference is the parabolic trajectory generated at the leg endpoint by the body-coxa joint. This uncommon movement was expected to make the model walk slower and with less stability. However, tests showed that this was not the case as it instead made the dung beetle model walk faster in a more straight line, with negligible change in its stability. The accurate kinematics also allows the possibility of manipulating dung balls. This was shown by placing the dung beetle model in different positions that the real dung beetle are commonly seen in when manipulating dung balls. The fact that the dung beetle model is able to adapt to these positions, indicates that it is able to somehow do the same tasks with a proper controller.

Therefore when making a biologically-inspired model that should solve certain tasks, it is suggested to do as little approximation as possible. An example of this is the hypothesised problem concerning the parabolic trajectories caused by the body-coxa joint. Solving this like an engineer proved to be worse than imitating the real dung beetle. Another suggested benefit is the ability to use the legs for both locomotion and manipulation like the real dung beetle, which most biologically-inspired hexapods with approximated leg kinematics are unable to do. Further study will show if it is possible to implement this manipulative behavior in the dung beetle model.

Different locomotive controller approaches have been examined. Utilizing the Good Old-Fashioned Artificial Intelligence approach would demand too many calculations, which both the dung beetle brain and many of today's processors are incapable of handling. Contrarily embodied AI is decentralized and it is therefore possible to layer and split the controller into several smaller controllers. A complex behavior is expected to emerge from the different controllers, all following simple rules which requires less processing power. For this reason the WALKNET controller, which is based on the principles of embodied AI, was implemented and tested on the dung beetle model. The tests showed that it is possible to

equip the dung beetle model with the WALKNET controller, even though it is inspired by the stick insect. The WALKNET controller implemented uses position based networks instead of the traditionally velocity based ones. This made it possible to implement the swing and stance networks as state machines instead of the more cumbersome neural networks. Additionally the position based controller helped in positioning the dung beetle model in common dung beetle poses, to demonstrate the manipulative abilities of the dung beetle model. Since the WALKNET controller is based on the principles of embodied AI and uses a subsumption architecture, it is possible to add additional behavioral layers, enabling the dung beetle model to actually perform these manipulative tasks.

All in all this study shows that it is possible to use the WALKNET controller on an accurate model of a dung beetle, based on measurements of a single dead dung beetle conserved in ethanol. It is also shown that the WALKNET controller is able to use position based networks, making it possible to use state machines instead of the velocity based neural networks. Finally it is revealed that the gait behavior emerging when using the WALKNET controller on the model is the tripod gait, which is comparable to that of the real dung beetle.

8 Further study

This study can be seen as a preliminary research into further study of the dung beetle with regard to biologically-inspired robotics. Some possible areas that can be investigated in the future are written below.

A source of error during this project can be contributed to the small amount of dung beetle specimens available at the Zoological Institute of Kiel University, as it was only possible to do each test once. Further study should include an increased sample size to remove unique errors that might have been with the specimen observed. It would be preferable to measure the difference in weights and kinematics between fresh dung beetles and dried to see if the preservation method can contribute to errors in e.g. rotational range of motion, or the weight. It would also be preferable to observe living specimens to analyse its locomotion and manipulation behaviors.

The implemented WALKNET controller is able to make the dung beetle model walk in a forward direction. Further work on the dung beetle model could include implementations of a turning and backward walking mechanism. It would be interesting to implement some additional locomotion behaviors that can enable the dung beetle model to find and roll away a dung ball.

It is possible in LPZ to make objects of different shape and size. So it would be interesting to test the robustness of the WALKNET controller in an environment with rough terrain, which would require the implementation of rule 4-6.

One the of primary themes for the study was that nature has been perfecting itself. The kinematics of the dung beetle leg should be a perfect machine for locomotion and spherical manipulation. This study focused on the locomotion part, so it would be interesting to look at the behavioral layer of manipulating objects. This would include standing on, and rolling a dung ball.

The WALKNET controller is implemented with position controlled actuators instead of the velocity based which is commonly seen. It would be interesting to implement the velocity based controller, to see the difference in performance compared to the position based one.

LPZ is developed so that it is possible to transfer the implemented controller to a real life robot. It could therefore be interesting to build a dung beetle model in real life and apply the WALKNET controller used in this study.

9 Acknowledgments

We express a deep sense of gratitude to PH.D stud. Leon Bonde Larsen, for encouraging us to do this study as well as giving us indispensable assistance and guidance during various stages of the project. The authors also thanks Assoc. Prof. Poramate Manoonpong, for his great interest in the project and for putting us in contact with Kiel University.

This work was therefore also supported by Kiel University, where PH.D Alexander Kovalev and Prof. Dr. Stanislav N. Gorb was of great help.

10 Bibliography

[1] Giuliano Di Canio, Stoyan Stoyanov, Christian Larsen, John Hallam, Thomas Kleinteich, Stanislav N Gorb, and Poramate Manoonpong. A Dung Beetle-like Leg and its Adaptive Neural Control, 2005.

[2] Axel Schneider, Jan Paskarbeit, Mattias Schaeffersmann, and Josef Schmitz. HECTOR, a new hexapod robot platform with increased mobility-control approach, design and communication. *Advances in Autonomous Mini Robots*, pages 249–264, 2012.

[3] Poramate Manoonpong, Ulrich Parlitz, and Florentin Wörgötter. Neural control and adaptive neural forward models for insect-like, energy-efficient, and adaptable locomotion of walking machines. *Frontiers in neural circuits*, 7(February):12, 2013.

[4] Robotshop.com. Picture of todays leg kinematics.

[5] Marcus Byrne. The dance of the dung beetle, https://www.ted.com/talks/marcus_byrne_the_dance_of_the_dung_beetle. 2012, Visited 10/2 2016.

[6] Keith W. Wait and Michael Goldfarb. A biologically inspired approach to the coordination of hexapedal gait. *Proceedings - IEEE International Conference on Robotics and Automation*, WeA9.4(April):275–280, 2007.

[7] B.H. Lee and I.K. Lee. The implementation of the gaits and body structure for hexapod robot. *IEEE International Symposium on Industrial Electronics, 2001. Proceedings. ISIE 2001*, 3(ii):1959–1964, 2001.

[8] T. Zielinska. Design of autonomus hexapod. *System*, pages 65–69, 1999.

[9] Yutaka Tanaka and Yasunori Matoba. Study of an intelligent hexapod walking robot, 1991.

[10] U. Saranli, M. Buehler, and D.E. Koditschek. Design, modeling and preliminary control of a compliant hexapod robot. *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 3(April):2589–2596, 2000.

[11] Bernie Kohl. Geotrupes stercorarius - A live dung beetle https://upload.wikimedia.org/wikipedia/commons/4/4d/AD2009Sep13_Geotrupes_stercorarius.jpg. Visited 26/4 2016.

[12] Yves Cambefort and Ikka Hanski. Dung Beetle Ecology. page 249, 1991.

[13] Youtube user: Edit Gosztola 'KekPafrany'. Geotrupes stercorarius movie, https://www.youtube.com/watch?v=eSc6okDcHHg, 2013, Visited 25/5 2016.

[14] Sarah A. Beynon. Geotrupes stercorarius top/bottom view picture, http://www.allaboutbeetles.co.uk/ show/english/hidden_pages/id_guide/geotrupidae.aspx . page 1, Visited 17/5 2016.

[15] Biokids. Coleoptera, http://www.biokids.umich.edu/critters/Coleoptera/. Visited 3/5 2016.

[16] Ralf Der, George Martius, Frank Hesse, Rene Lievscher, Marcel Kretschmann, Dominic Schneider, Claus Stadler, and Fran Güttler. LPZ robots documentation, http://robot.informatik.uni-leipzig.de /software/lpzrobots_doc-0.2/html/index.html, Visited 10/2 2016.

[17] Russell Smith. Open Dynamics Engine, http://www.ode.org/. Visited 10/5 2016.

[18] S.A. Neporotovskiy. Geotrupes stercorarius side view pictrue, https://www.zin.ru/animalia/coleoptera/eng/geostenp.htm. page 1, Visited 17/5 - 2016.

[19] Dmytro Gladun and Stanislav N. Gorb. Insect walking techniques on thin stems. *Arthropod-Plant Interactions*, 1(2):77–91, 2007.

[20] Volker Dürr, Josef Schmitz, and Holk Cruse. Behaviour-based modelling of hexapod locomotion: Linking biology and technical application. *Arthropod Structure and Development*, 33(3):237–250, 2004.

[21] Rodney Brooks. Elephants don't play chess. *Robotics and Autonomous Systems*, 6(1-2):3–15, jun 1990.

[22] Rodney A Brooks and Rodney A Brooks. Intelligence Without Reason. *Proceedings of the 12th International Joint Conference on Artificial Intelligence ({IJCAI}-91)*, 1293(Fall):569–595, 1991.

[23] Holk Cruse, Thomas Kindermann, Michael Schumm, Jeffrey Dean, and Josef Schmitz. Walknet - A biologically inspired network to control six-legged walking, 1998.

[24] Malte Schilling, Holk Cruse, and Paolo Arena. Hexapod Walking: An expansion to Walknet dealing with leg amputations and force oscillations. *Biological Cybernetics*, 96(3):323–340, 2007.

[25] Paolo Arena, Luca Patane, Malte Schilling, and Josef Schmitz. Walking capabilities of

Gregor controlled through Walknet. *Proceedings of SPIE*, 6592(0):65920K–65920K–9, 2007.

[26] Malte Schilling, Thierry Hoinville, Josef Schmitz, and Holk Cruse. Walknet, a bio-inspired controller for hexapod walking. *Biological Cybernetics*, 107(4):397–419, 2013.

[27] Danya Rao. GRASP Design Principles, https://www.cs.colorado.edu/ kena /classes/5448/f12/presentation-materials/rao.pdf. Visited 13/5 2016.

Appendices

A lpzrobots

In order to both model a dung beetle and simulate it in a realistic environment, an already existing framework called LPZROBOTS (LPZ) is used. LPZ is based on the Open Dynamic Engine [16] which is an open source library for simulating rigid body dynamics implemented in `C/C++` [17]. LPZ offers the possibility to create and view models in a 3D environment, as well as adding a controller to the model.

LPZ is a collection of multiple major and minor projects. Two of these projects, used in this study, are `ode_robots` and `selforg`. These are both mainly developed by *Georg Martius* and described below.

`ode_robots`: A project that includes the majority of utilities used when developing a model, which includes primitives, actuators and sensors. To visualize the model in a 3D environment the project `OpenSceneGraph` is used [16]. All the physics in the simulations are described by ordinary differential equation. `ode_robots` is used for solving these equations by numerical integration, where the method used is Runge-Kutta 4.

`selforg`: A project developed by the Robotic group of Leipzig University. This project is used for linking the model with a controller [16].

LPZ includes several important classes itself, which provides features that can be instantiated by the model. Examples of these are actuators and sensors. The most commonly used actuator and sensor types are already implemented in LPZ . It is however possible to modify them in order to implement unique types for a given study. An example of this is the position based actuator implemented during this study.

When developing a project in LPZ the three main classes listed below are mostly used.

`model`: Is a class in which the model is created and where all actuators and sensors are instantiated.

`empty_controller`: Is the primary controller class. All sensory data can be accessed from here, and it is also possible to send commands to the actuators.

`simulation`: This is the class where the model and the controllers are instantiated and aggregated to each other. The class also sets op the environment of the simulation, including simulation time, obstacles and gravity.

General Responsibility Assignment Software Patterns, abbreviated GRASP, is a term in software development for assigning responsibility to classes and objects in object oriented programming. The GRASP patterns are mostly used to solve some of the problems that commonly arise during a object oriented project.

Some of the common patterns in relation to LPZ are listed below.

Creator: Creation of objects is one of the most common activities in an object-oriented system. Which class is responsible for creating objects is a fundamental property of the relationship between objects of particular classes [27].

- The `model` class in LPZ is an excellent example of this pattern. Every body part, actuator and sensor are separate objects that are created and handled.

High cohesion & low coupling: These two patterns are integrally connected.

A project with high cohesion is typically created from many small classes that are specialized to a certain task. Low cohesion is on the other hand a sign of few classes that are not specialized to a certain task. These are also known as garbage classes, which can be hard to maintain.

A class with low coupling makes it reusable in other applications, or other places in the project, just like a LEGO brick. When a class is strongly dependent on other classes it is said to be high coupled and is therefore not easily used in other projects.

- An example of this is the relationship between the `empty_controller` and `actuator` class. There exists a very low coupling and a high cohesion between these classes. This is shown by the fact that the two classes only communicate through one single variable, which is a sign of low coupling. Also the `empty_controller` does not know the variable indicating the real time in the actuators which is a sign of high cohesion.

Polymorphism: Inheritance in a software project makes it easy to manage new variations of an already existing class.

- The `motor` and `sensor` classes in LPZ are prime examples of this. Actuators in LPZ utilizes methods that lies in the core of `ode_robots`. All actuators are required to use these virtual methods to work properly. The sensory data in LPZ works in a similar way. There are virtual methods that are required to be called for a sensor to work.

LPZ proved to be a helpful framework when working with a biologically inspired model. The only existing drawbacks are the low amount of primitives available and the fact that the `empty_controller` class is too isolated from the `actuator` class, due to low coupling

and high cohesion. These classes could have been more coupled so that it would be easier to create PID's and other time dependent methods in the `empty_controller` class.

B Results from biological investigation

Measuring of body parts

The following table shows the results of the measurements and weighing of the dung beetle body parts. A note here that the tarsus was not weighted due to crumbling during the detachment.

Body part		Length [mm]	Width [mm]	Heigth [mm]	Weight[mg]
Full body					106.4
Head		4.5	3.7	2.9	14.8
Thorax		5.1	9.1	4.3	23.8
Abdomen		9.0	10.3	4.0	30.4
Coxa	Front	2.4	1.6		1.2
	Middle	2.1	1.6		1.5
	Hind	4.0	1.6		3.0
Femur	Front	3.2	1.8		2.8
	Middle	4.2	2.0		2.2
	Hind	4.6	2.4		2.6
Tibia	Front	4.6	1.0		1.5
	Middle	3.7	0.9		1.3
	Hind	5.5	0.9		2.1
Tarsus	Front	3.4	0.2		
	Middle	3.0	0.2		
	Hind	3.9	0.2		

Table 3: This table shows the results from the weight- and dimension tests done in Kiel

Rotational range of motion

The following table shows the results of the rotational range of motion of the different dung beetle joint.

Joint placement		Rotational range of motion [degree]
Head-Thorax *(HT)*		45
Thorax-Abdomen *(TA)*		45
Thorax-Coxa *(BC)*	Front	95.7878
Abdomen-Coxa *(BC)*	Middle	116.1153
Abdomen-Coxa *(BC)*	Hind	160.8514
Coxa-Femur *(CF)*		90
Femur-Tibia *(FT)*		170

Table 4: This table shows the results from the tests finding the rotational range of the different joints. The HT-, TA-, CF- and FT-joints are all hinge joint, where the BC-joints are pivot joints

Coxa placement

The following table shows the calculated angles for placing the coxa anatomic correct in a 3D space.

Coxa	x rotation [degree]	z rotation [degree]
Front	65.3676	112.5464
Middle	56.2446	110.4237
Hind	64.6293	95.7143

Table 5: The calculated rotations of the coxae in a 3D space

C Dung beetle model figures

The following figures shows the dung beetle model from different angles and in different positions.

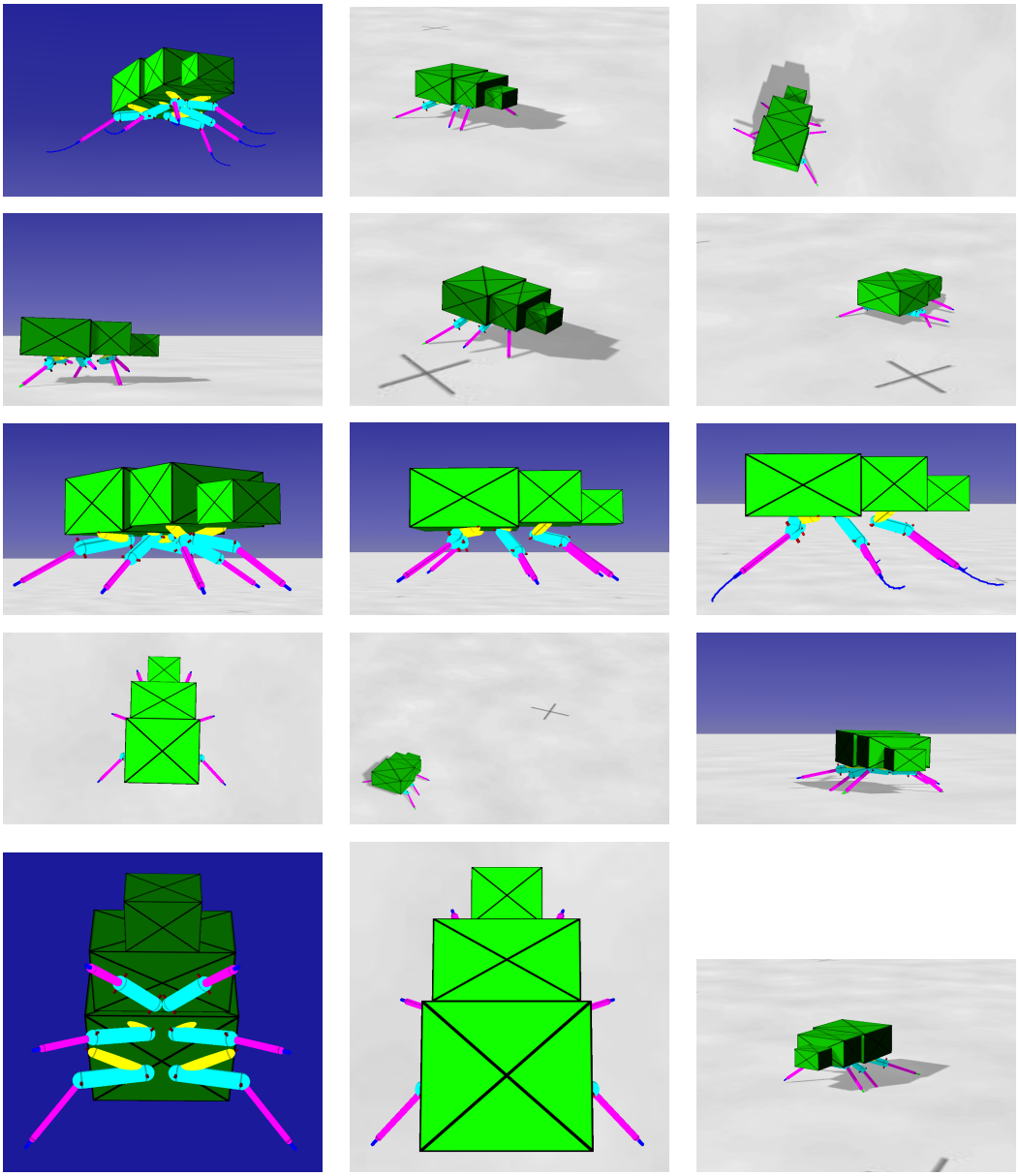
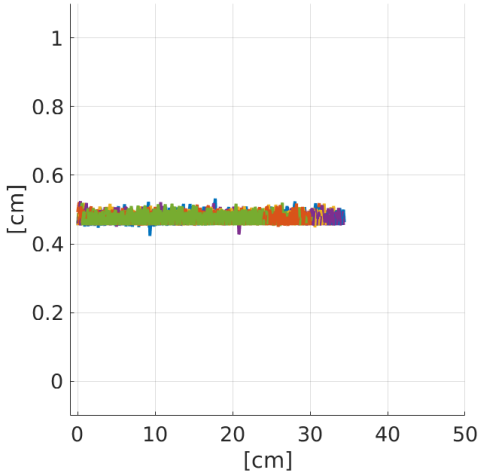


Figure 32: The dung beetle model

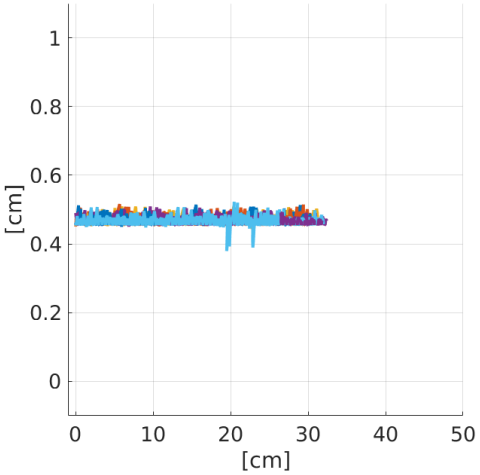
D Additional results from simulation

D.1 Height test results

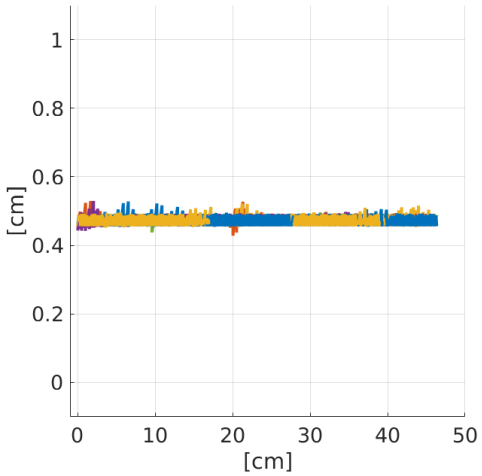
The following figures includes five trials for each combination of swing and stance net.



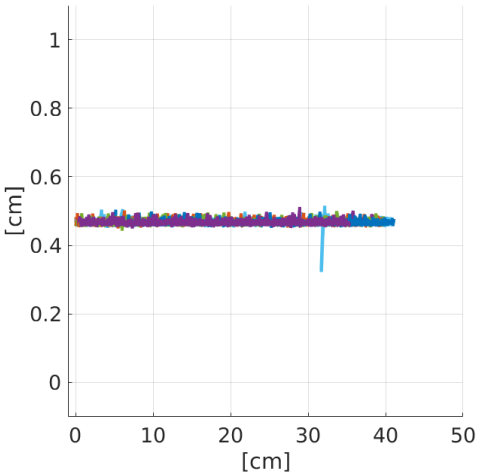
(a) Height test - swing net 1 and stance net 1



(b) Height test - swing net 1 and stance net 2



(c) Height test - swing net 2 and stance net 1



(d) Height test - swing net 2 and stance net 2

Figure 33: Plot of the four combination tested. The plot includes all the trials for each combination. Each color indicates a different trial

The following tables shows all five trials for each combination of swing and stance net.

Combination	trial	1	2	3	4	5
11	mean [cm]	0.4656	0.4656	0.4656	0.4655	0.4660
	variance	0.0002	0.0002	0.0002	0.0002	0.0002
	st.d [cm]	0.0145	0.0140	0.0140	0.0139	0.0141
	max deviation [cm]	0.5321	0.5217	0.5195	0.5246	0.5202

Table 6: Height of the robot while walking with combination **11**

Combination	trial	1	2	3	4	5
12	mean [cm]	0.4631	0.4631	0.4630	0.4631	0.4622
	variance	0.0001	0.0001	0.0001	0.0001	0.0001
	st.d [cm]	0.0105	0.0108	0.0107	0.0103	0.0121
	max deviation [cm]	0.5142	0.5160	0.5091	0.5093	0.5227

Table 7: Height of the robot while walking with combination **12**

Combination	trial	1	2	3	4	5
21	mean [cm]	0.4675	0.4677	0.4678	0.4673	0.4674
	variance	0.0001	0.0001	0.0001	0.0001	0.0001
	st.d [cm]	0.0118	0.0121	0.0119	0.0117	0.0115
	max deviation [cm]	0.5282	0.5296	0.5217	0.5296	0.4958

Table 8: Height of the robot while walking with combination **21**

Combination	trial	1	2	3	4	5
22	mean [cm]	0.4656	0.4650	0.4645	0.4651	0.4651
	variance	0.0001	0.0001	0.0002	0.0001	0.0001
	st.d [cm]	0.0083	0.0087	0.0126	0.0086	0.0085
	max deviation [cm]	0.5041	0.5018	0.5157	0.5120	0.4999

Table 9: Height of the robot while walking with combination **22**

D.2 Walking test results

The following tables shows all five trials for each combination of swing and stance net.

Combination	trial	1	2	3	4	5
11	mean [cm]	-1.0855	-3.1203	-1.9316	-1.7859	-3.3199
	variance	0.2796	2.2549	0.7535	0.6311	2.8907
	st.d [cm]	0.5288	1.5016	0.8681	0.7944	1.7002
	max deviation [cm]	1.8218	4.9759	3.0331	2.9242	5.4764
	distance [cm]	34.4144	32.9863	33.6299	34.3092	33.4206

Table 10: Table over how straight the robot is walking with combination **11**

Combination	trial	1	2	3	4	5
12	mean [cm]	1.0114	1.4341	1.8425	0.4405	-0.4478
	variance	0.3408	0.4983	0.5987	0.2523	0.1144
	st.d [cm]	0.5838	0.7059	0.7738	0.5023	0.3383
	max deviation [cm]	1.9580	2.3688	2.6038	1.1923	1.0589
	distance [cm]	31.2825	31.3660	31.3258	32.3834	31.9866

Table 11: Table over how straight the robot is walking with combination **12**

Combination	trial	1	2	3	4	5
21	mean [cm]	-0.3300	0.9035	-0.4266	-0.1548	0.4577
	variance	0.0180	0.1868	0.0612	0.0141	0.0674
	st.d [cm]	0.1340	0.4322	0.2473	0.1189	0.2596
	max deviation [cm]	0.5822	1.6032	0.8848	0.3867	0.9925
	distance [cm]	46.3417	45.8411	45.4148	45.0615	46.3460

Table 12: Table over how straight the robot is walking with combination **21**

Combination	trial	1	2	3	4	5
22	mean [cm]	-2.1015	-1.6130	0.2277	-2.8835	-1.7946
	variance	1.1539	0.5733	0.0280	1.9228	0.5568
	st.d [cm]	1.0742	0.7572	0.1673	1.3866	0.7462
	max deviation	3.4037	2.5925	0.4324	4.7578	2.7124
	distance [cm]	41.1332	39.7121	40.7407	39.5712	39.8121

Table 13: Table over how straight the robot is walking with combination **22**