Consider the Robot

—

Abstraction of Bioinspired Leg Coordination and its Application to a Hexapod Robot Under Consideration of Technical Constraints

DISSERTATION

submitted in partial fulfillment
of the requirements for the degree of
Doktor der Naturwissenschaften
(Dr. rer. nat.)

Jan Paskarbeit

Faculty of Technology
Bielefeld University
March 2017

You can't possibly be a scientist
if you mind people thinking that you're a fool.

Wonko the Sane
in So Long And Thanks For All The Fish
by Douglas Adams (1985)

Acknowledgments

The list of people to whom I owe gratitude is too long to be printed without giving the impression of me having done no work at all. Therefore, I will list only the most important supporters of this work by name and assure the unnamed majority of my enduring gratefulness. Thank you!

To begin with, I would like to thank my supervisor Prof. Dr. Axel Schneider for the opportunity and the challenge to create a robotic system from scratch, the willingness to discuss fallacious concepts, and for motivating me to continue.

With respect to the biological topics of this work, Prof. Dr. Josef Schmitz, Prof. Dr. Holk Cruse, and Dr. Malte Schilling have supported me tremendously. Had it not been for them, the robot would be much less bioinspired and functional.

I would especially like to thank my colleagues from the Biomechatronics Group, namely (and alphabetically) Dr. Salvatore Annunziata, Daniel Basa, Dr. Hanno Gerd Meyer and Mattias Schäffersmann for productive discussions, lots of practical help, and those countless cake breaks.

During the design phase, Prof. Anke Bernotat and Achim Seemayer have ensured a pleasing appearance of the robot, which I certainly would not have achieved myself. Undoubtedly a challenge on its own, Dr. Axel Spickenheuer, Kai Uhlig, and Martin Schulz have added a functional layer (consisting mostly of CFRP) to the design by realizing a sturdy, yet light body structure.

Talking about the construction of the robot, I would like to thank the members of the mechanical workshops of Biology-CITEC and Physics for their enthusiasm for the manufacturing of the mechanical parts.

The whole endeavor of writing this thesis would have come to an early, sudden, demotivated stop without the constant support of my wife Nicole Paskarbeit. Also, I would like to thank my family for never abandoning the hope of me finishing this thesis.

Finally, I would like to thank Prof. Dr.-Ing. Ralf Möller and Prof. Dr. Hartmut Witte who agreed to review this thesis.

This work has been supported by the DFG Center of Excellence ’Cognitive Interaction TEChnology’ (CITEC, EXC 277) within the EICCI-project and by the EU-FP7 grant EMICAB (ICT-2009.2.1 270182).

Summary

To emulate the movement agility and adaptiveness of stick insects in technical systems such as piezo actuators (Szufnarowski et al. 2014) or hexapod robots (Schneider, Cruse, et al. 2006), a direct adaptation of bioinspired walking controllers like WALKNET has often been suggested. However, stick insects have very specific features such as adhesive foot pads that allow them to cling to the ground. Typically, robots do not possess such features. Besides, robots tend to be bigger and heavier than their biological models, usually possessing a different mass distribution as well. This leads to different mechanical and functional properties that need to be addressed in control.

Based on the model of the stick insect *Carausius morosus*, the six-legged robot HECTOR was developed in this work to test and evaluate bioinspired controllers. The robot's geometrical layout corresponds to that of the stick insect, scaled up by a factor of 20. Moreover, like the stick insect, the robot features an inherent compliance in its joints. This compliance facilitates walking in uneven terrain since small irregularities can be compensated passively without controller intervention. However, the robot differs from the biological model, e.g., in terms of its size, mass, and mass distribution. Also, it does not possess any means to cling to the ground and therefore must maintain static stability to avoid tilting.

To evaluate the ability of stick insects to maintain static stability, experimental data (published by Theunissen et al. (2014)) was examined. It can be shown that stick insects do not maintain static stability at all times. Still, due to their adhesive foot pads, they do not tumble. Therefore, a direct replication of the biological walking controller would not be suitable for the control of HECTOR.

In a next step, the bioinspired walking controller WALKNET (Cruse, Kindermann, et al. 1998) was evaluated regarding its applicability for the control of HECTOR. For this purpose, different parametrizations of WALKNET were tested in a simulation environment. For forward walking, parameter sets were found that achieve a high, although not permanent stability. Thus, for the control of HECTOR, which requires continuous stability, a more abstract adaption of the bioinspired coordination had to be found.

Based on the original coordination concepts of WALKNET, new coordination mechanisms were developed that incorporate the technical requirements (static stability, angular joint limits, torque constraints, etc.). The ability of the resulting controller to generate insect-like gaits is demonstrated for different walking scenarios in simulation. Moreover, locomotion that is unlikely for insects such as backwards and sidewards walking is shown to be feasible using the novel control approach. At the end of this work the applicability of the approach for the control of the real robot is proved in experiments on visual collision avoidance and basic climbing ability.

Contents

1. Introduction

> My dear Miss Glory, Robots are not people.
> They are mechanically more perfect than we are,
> they have an astounding intellectual capacity,
> but they have no soul.
>
> *Domin*
> *in* R.U.R. (Rossum's Universal Robots)
> *by Karel Čapek (1921), translated by*
> *Claudia Novack (2004)*

Before the word *robot* was coined by Karel Čapek in his play *R.U.R. (Rossum's Universal Robots)* in 1921, the concept of lifeless, yet lifelike mechanisms was already quite old—until then they were named *automatons* or *androids* (Riskin 2003). Most of these automatons imitated biological systems. However, bound to the technological possibilities, their functionality was restricted to the repetition of particular, mechanically determined motion sequences. One example for such an automaton is an artificial swan from 1733 that was constructed by a mechanic named Maillard (Académie des sciences 1735). This swan contained a mechanism that moved its head from side to side and propelled the swan on water using a paddle-wheel (depicted in fig. 1.1 (a)). Since the automatons were used as showcases for the technological abilities of their creators, the mechanisms were improved piece by piece to imitate the biological models more accurately.

In the 18th century, automatons emerged that also simulated the inner functionality of animals and humans. For example, Vaucanson's flute player imitated the movement of a human including flexible fingers, lips, tongue, and the generation of various air pressures (Vaucanson 1738). Beside its primary use as a showcase, this automaton also allowed for a deeper understanding of the physics of sound production (Riskin 2003).

The invention of electrically driven actuators revolutionized the mechanics of the automatons, and the invention of logic devices widened the applicability of the machines (Devol 1961; Walter 1950). This allowed for the development of robots in their current form. Beside the purely work-related robots built for automation tasks, still, animal- and human-like robots are being developed. As for the development of the previously discussed automatons, one reason for the development of these robots is the immediate emotional connection of humans when they encounter a machine in the form of a human (Sorbello et al. 2014) or an animal (Wada and Shibata 2007). These bio-mimicking robots are therefore used primarily for human-machine interaction.

Another reason for the development of bioinspired robots is the utilization of optimized concepts from nature for application in technology. Over the course of millions of

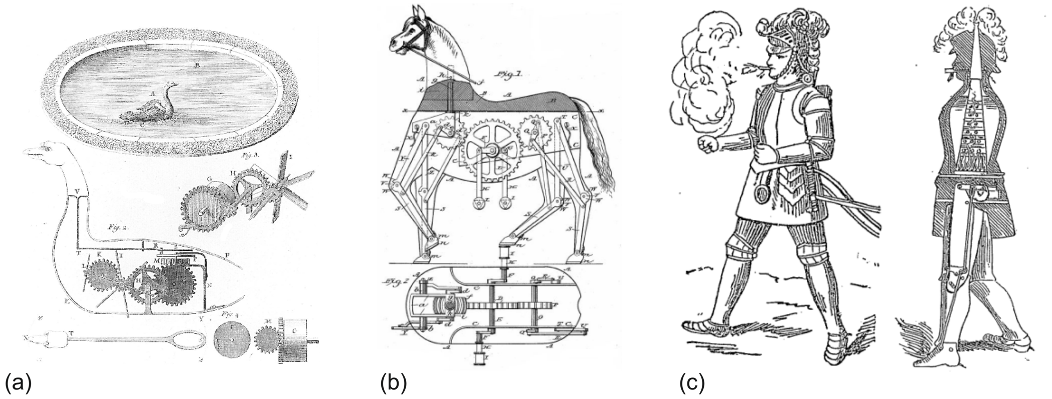


Figure 1.1.: Drawings of automatons. (a) shows a function drawing of Maillard's mechanical swan (adopted from Académie des sciences (1735)). Using the depicted mechanism the swan was able to propel itself in water and to move its head simultaneously. (b) illustrates a concept for a mechanical horse that was planned to be driven by foot pedals (adopted from Académie des sciences (1735)). (c) shows a drawing of the *Mechanical Man* as invented by Moore (adopted from NYT (1893)).

years, animals have evolved that are adapted to their respective ecological niche (Darwin and A. Wallace 1858). In the development of a technical system that needs certain properties that can be found in animals, the evolved features can be taken as a starting point for further optimization. However, a direct replication of the solution as found by evolution is rarely reasonable since animals most likely need to comply with other requirements than technical systems. These differences must be considered in the selection of a biological model and in the abstraction of its features for the implementation on a robot.

Beside these rather technology-driven adaptations of biological features, robots have also been used extensively to model biological sensorimotor systems (Webb 2001). The *biorobotics* approach is followed if a purely virtual emulation of a system is considered inadequate for the intended research, e.g., if the interaction with the environment is too complex to be modeled. For example, biorobots have been used to get a better understanding of the visual homing strategies used by insects to find back to their nest (Möller 2000) or to locate a mate by phonotaxis (Webb and Scutt 2000). Although the admissible level of abstraction used for the transfer of biological features to a technical representation is disputed (Webb 2001), robots are widely used in biological research.

In order to understand their biological foundations as well as to find suitable abstractions for technical application, various strategies for locomotion of animals have been studied intensively and the effort to reproduce them in robots is still continuing. Robotic fish (Katzschmann et al. 2016), birds (Send et al. 2012), flying insects (Ma et al. 2013), snakes (Rollinson 2014), and even microbes (Hou et al. 2010; Paskarbeit, Beyer, et al. 2016) have been constructed based on observations of the biological models. Most animal-like robots, however, use legs for locomotion. The expansion of robots into everyday life calls for the adaptability to the modern environment of humans. This includes the need to use stairs and to operate on ground that was not designed for wheeled locomotion in the first place. Also for locomotion in natural, uneven terrain that was not yet made accessible for wheeled vehicles, legged robots are in advantage. However, the advantage of flexible adaptation to uneven terrain in legged locomotion comes at the cost of potential mechanical instabilities.

The plethora of bioinspired legged robots (discussed in more detail in section 1.1) can be roughly divided into the categories of *dynamically* and *statically stable*. Robots of the first type are usually modeled after biped or quadruped animals. During locomotion with less than three feet on the ground, the body will always tilt in one direction.[1] This tilting, however, can be counterbalanced by the next step, that—ideally—induces a compensating moment (Raibert 1986). Therefore, the maintenance of *dynamic stability* requires fast control.

Systems that rely on *static stability* must have at least three feet on the ground at all times. Moreover, the projection of the COM (Center Of Mass) along the vector of gravity must be located within the support polygon spanned by the leg tips (see also section 1.2). Therefore, at least four legs are required for statically stable walking, allowing one leg at a time to be lifted off the ground for repositioning. Systems with more than four legs can therefore lift multiple legs at the same time for protraction without necessarily compromising static stability.

The development of ever faster processor units facilitated the design of dynamically stable robots that achieve high speeds during running. However, statically stable robots are still being developed both for safety-critical applications in rough terrain as well as for the research on locomotion of multi-legged animals. For both of these domains, especially six-legged robots have been used frequently since they can utilize a wide range of different, statically stable gaits and they allow the research of insect-like leg coordination.

In this work, the mechanics and control of the hexapod robot Hector (HExapod Cognitive auTonomously Operating Robot) are presented. The robot is intended for biology-related research and the evaluation of bioinspired control approaches for the application in a technical system. For a general overview on legged robots, the development of walking machines is summarized in section 1.1. Usually being mechanically more complex than wheeled robots, a challenge in the operation of legged robots is the coordination of the legs to produce stable gaits. Different bioinspired or purely technical concepts for gait generation are therefore discussed in section 1.2. One of the main mechanical features of Hector is the application of compliant actuators in the leg joints. Adopted from the compliance of muscles, compliant actuators are increasingly used to facilitate human-machine interaction, to compensate for sensor and control errors, and to store energy during repetitive tasks. For the application in a rather slow walking hexapod robot, compliant actuators are mostly preferable due to their compensation

[1] If the size of the feet is increased, even less than three feet on the ground can be sufficient to maintain stability without intervention of a controller. Within this work, however, point-like contacts are assumed, of which three are the minimal requirement for static stability.

ability in rough terrain. In section 1.3, different concepts for compliant actuation are presented.

1.1. Legged Robots

> Before we dreamed of driverless cars,
> we dreamed of horseless horses.
>
> *Garber (2012)*

Wheel-driven robots have many advantages over legged robots: Both the mechanics as well as the control are simpler. On hard and smooth ground, wheel-based vehicles are likely faster than legged robots. Also, in general, they are more energy-efficient since no energy is required for maintenance of the posture. However, wheel-based robots need environments that are suitable for wheels, therefore sufficiently smooth and with the right amount of friction for traction. Therefore, in situations in which no continuous path is available, legged robots are in advantage over their wheeled counterparts. Since they are able to use discrete footholds, the mobility of a legged robot is limited only by the quality of the best footholds within reach, whereas a wheeled robot is limited by the worst segment along its path (Raibert 1986). In uneven terrain, legs are able to decouple the robot's body from the track, thus smoothing the unevenness. Another advantage of legged robots is that they can use their limbs also for object manipulation whereas wheeled robots need specific manipulators in addition to the wheels for this task. However, to play off these advantages, legged robots need sophisticated planning algorithms that are able to identify appropriate footholds and maintain the stability of the system at all times.

Table 1.1 gives a chronological overview of the development of legged automatons and robots throughout history, focusing on but not limited to multi-legged systems. The table also lists whether the robots are constructed using compliant actuators and—if applicable—the biological model. The first legged automatons were probably mechanical horses (Goodwin 1867; Rygg 1893) as depicted in fig. 1.1 (b). However, without horse-inspired controllers, the automatons were not able to compete with the adaptability appreciated of real horses. In 1893, Moore presented a steam-powered biped (depicted in fig. 1.1 (c)) that looked and walked almost like a human (NYT 1893). However, fastened to a horizontal bar that was fixed in the center to prevent tumbling, the mechanical man was limited to walking in circles. The restrictions of mechanical components in the design of controllers prohibited the breakthrough of automatons as replacement for horses and human workers.

More elaborate walking machines were realized in two different approaches in 1968. The *Iron Mule Train* (see fig. 1.2 (a)) was supposed to carry loads through rough terrain (Morrison 1968). It used eight legs, which were mechanically coupled so as to ensure that at all times at least four legs were in contact with the ground. The idea was to couple multiple of these automatons with the first in the line being powered by a motor pulling the following merely passive walkers along.

The other approach was the *General Electric Quadruped Truck*, depicted in fig. 1.2 (b),

Table 1.1.: Selection of legged automatons and robots

name	**year**	**# of legs**	**controller**	**DoF/ leg**	**compliance**	**biological model**	**mass [kg]**	**references**
The Steam Man	1893	2	mechanical	2	-	human		NYT (1893)
Phony Pony	1967	4	finite control algorithms	2	-		50	McGhee (1967)
General Electric Quadruped Truck	1968	4	human-in-the-loop	3	-		1400	Mosher (1968)
Iron Mule Train	1968	8	mechanical	0.25	-			Morrison (1968)
Genghis	1989	6	AFSM	2	-			Brooks (1989)
Robot I	1992	6	bioinspired, WALKNET coordination rules	2	-			Beer et al. (1992) and Espenschied, Chiel, et al. (1993)
Lauron I	1994	6	ANN	3	-	*Carausius morosus*	11–13	Berns et al. (1994)
Tarry I	1994	6	ANN	3	-	*Carausius morosus*	2.1	Amendt and Frik (1994)
Hannibal	1995	6	bioinspired	3	-		2.8	Ferrell (1995)
Lauron II	1995	6	multiple fixed gaits	3	-	*Carausius morosus*	15.9	S. Cordes and Bührle (1996)
Robot II	1996	6	generalized WALKNET coordination rules	3	spring and damper in *tibia*, controlled	stick insect		Espenschied, Quinn, Beer, et al. (1996)
Robot III	1997	6		5/4/3	pneumatic	cockroach	15	Nelson and Quinn (1999)
Lauron III	1999	6	behavior-based	3	-	*Carausius morosus*	18	Gaßmann et al. (2001)
Tarry II	1999	6	WALKNET	3	-	*Carausius morosus*	2.9	Frik et al. (1999)

Continuation of table 1.1

name	**year**	**# of legs**	**controller**	**DoF/ leg**	**compliance**	**biological model**	**mass [kg]**	**references**
Scorpion I	2000	8		3	-	scorpion	9.5	Spenneberg et al. (2007)
Scorpion II	2000	8		3	-	scorpion		Spenneberg et al. (2007)
Scorpion III	2001	8		3	spring and damper in *tibia*	scorpion		Spenneberg et al. (2007)
Scorpion IV	2002	8	CPG + posture control + reflexes	3	spring damper in *tibia*	scorpion		Spenneberg et al. (2007)
Lauron IV	2004	6	behavior-based	3	-	*Carausius morosus*	27	Rönnau, Kerscher, et al. (2009)
Robot V (ajax)	2006	6		5/4/3	pneumatic			Kingsley et al. (2006)
Max	2007	6	WALKNET coordination rules	3	-	stick insect	23	Pfeiffer (2007)
DLR Crawler	2008	6	WALKNET coordination rules	3	controlled		3.5	Görner, Wimböck, Baumann, et al. (2008)
AMOS-WD06	2008	6	CPG	3	-	stick insect	4.2	Manoonpong et al. (2008)
Scarabaeus	2008	6		3	-			Bartsch and Planthaber (2009)
Space Climber	2010	6	CPG + posture control + reflexes	4	spring and damper in *tibia*			Bartsch, Birnschein, F. Cordes, et al. (2010)
Lauron V	2013	6	behavior-based	4	spring and damper in *tibia*	*Carausius morosus*	42	Rönnau, Heppner, Nowicki, et al. (2014)

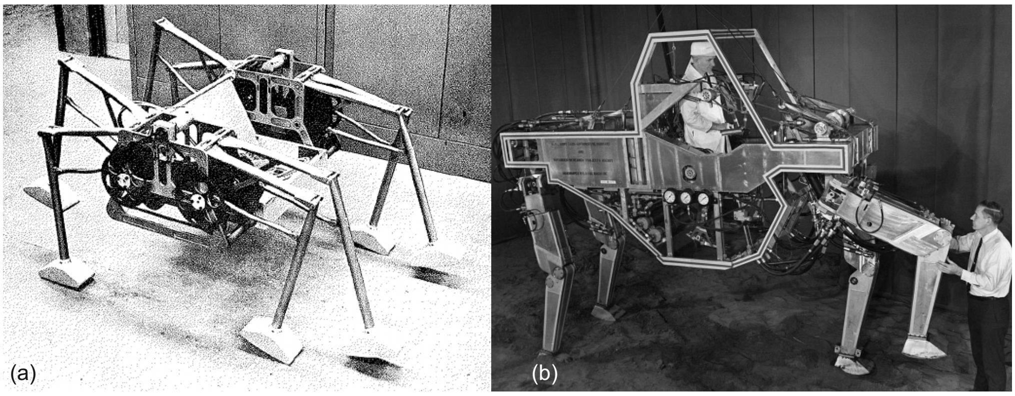


Figure 1.2.: Early walking machines. (a) shows one segment of the *Iron Mule Train* (picture adopted from (Morrison 1968)). (b) shows the *General Electric Quadruped Truck* whose legs were controlled by a human operator from within the machine (Granger, NYC — All rights reserved.).

that weighted 1400 kg (Mosher 1968). Instead of using a fixed gait defined by coupled mechanics, it used a human operator/driver for gait control. The operator had to control three degrees of freedom per leg. The positions of the machine's hind legs were controlled by the operator's feet, the front legs by his hands. According to Raibert (1986, p. 8), a "surprising agility" in the control could be achieved within twenty hours of training. However, according to Todd (1985, p. 15), the control task was very demanding and therefore the vehicle could be controlled only for a "few minutes at a time".

The obvious need for an automatic control of the gait pattern—leaving the operator for high-level commands—inspired McGhee and Frank to use a non-mechanical computer for control (Todd 1985, p. 23), which was realized in the development of the quadruped *Phony Pony* (Frank 1968; McGhee 1967). Using finite control algorithms, the robot was able to perform crawl and trot gaits. The latter, however, could be achieved only by lateral extension of the feet to stabilize the robot.

The development of fast non-mechanical computers allowed also for the development of dynamically stable robots. The first one-legged robot with a flight phase that was able to move autonomously in 3D space was the *hopping machine* by Raibert et al. (1984).[2] The underlying concept was later also expanded to four legs (Raibert 1986). The advantage of this approach is the ability to achieve higher speeds during running (as compared to statically stable walking). Since the maintenance of dynamic stability anyway requires fast controllers and reflex-like behaviors to compensate, e.g., for errors in measurement of the sensors, minor disturbances of the system (slipping of feet, small impacts, etc.) can be counterbalanced automatically.

However, for applications in rough terrain for which maximum safety is demanded,

[2] In 1945, a manned, one-legged hopping tank was patented that was supposed to use a similar concept (H. W. Wallace 1945).

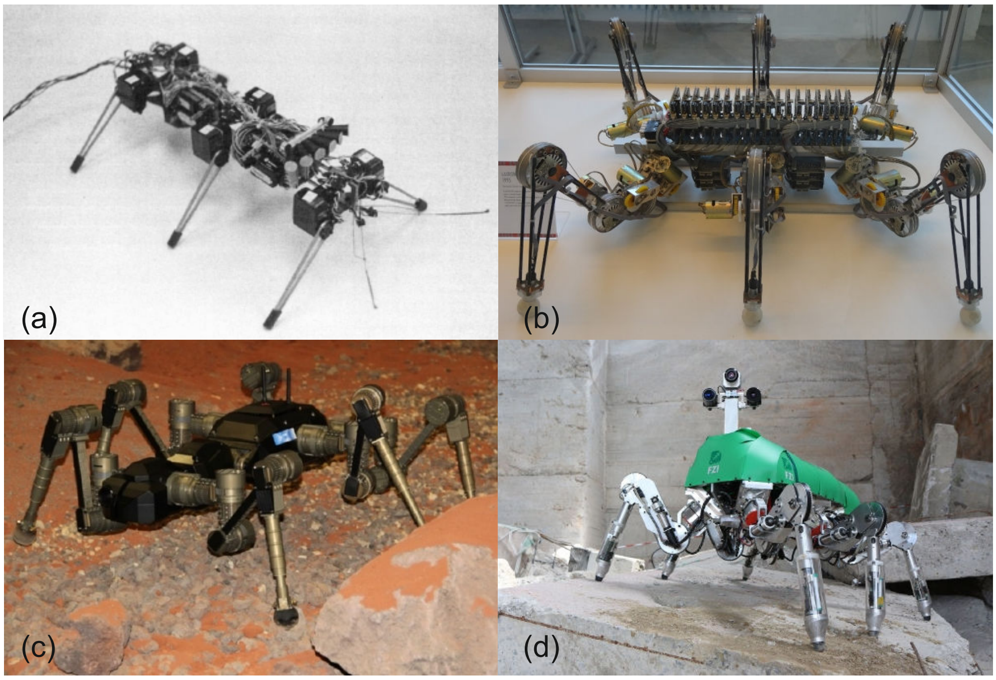


Figure 1.3.: Early and current hexapod robots. (a) *Genghis* (picture adopted from Brooks (1989)), (b) *Lauron* at the *Deutsches Museum* in Munich (courtesy of Alexander Entinger), (c) *SpaceClimber* (picture adopted from DFKI GmbH & University of Bremen (2016)), (d) *Lauron V* (picture adopted from Rönnau, Heppner, Nowicki, et al. (2014)).

slow, yet statically stable walking is desirable. Usually, missions on other planets, within volcanic craters, or nuclear power plants are given as examples (Bares and Wettergreen 1999; Bartsch, Birnschein, Römmermann, et al. 2012; Tedeschi and Carbone 2014). Especially to reproduce the climbing abilities of insects, robots with six legs have been studied intensively. One of the first robots that resulted from these studies was *Genghis* (Brooks 1989, shown in fig. 1.3 (a)). It has six legs with two actuators per leg. Additionally, it features an infrared light sensor that was used to detect a human which the robot could follow. The controller consisted of a network of FSMs (Finite State Machines) that were augmented by timers. Communication between these AFSMs (Augmented FSMs) was realized by additional registers that influenced their states. These registers could be modified by hierarchically higher parts of the network. This allowed implementations of tripod walking (two groups of three legs alternate with stance and swing phases) as well as wave gaits (deriving its name from the forward-directed metachronal wave of protractions of the legs along the body). *Genghis*' successor, *Hannibal*, used three actuators per leg, force sensors to measure loads on the legs and to detect collisions, angle and velocity sensors in the actuators, and ground contact sensors at the feet (Ferrell

1995). On *Hannibal*, different insect-inspired controllers as proposed by biologists could be tested (Ferrell 1995; based on the works of Cruse (1976a,b, 1979a, 1980a,b, 1990), Cruse, Bartling, Cymbalyuk, et al. (1994), Pearson (1976), and Wilson (1966)). In parallel, the first *Lauron* robot, shown in fig. 1.3 (b), was developed (Berns et al. 1994). The design of the legs was based on the model of the stick insect *Carausius morosus*. This includes the slanted angle of the first joint axis of each leg relative to the main body (Cruse 1976b). For the leg control, an ANN (Artificial Neuronal Network) was trained with joint angle data obtained from moving an unactuated wood model of the robot. In the later versions of *Lauron*, *Lauron II-V*, the mechanics and sensors were successively improved (S. Cordes and Bührle 1996; Gaßmann et al. 2001; Rönnau, Heppner, Nowicki, et al. 2014; Ziegenmeyer et al. 2009). In the current version, *Lauron V*, which is shown in fig. 1.3 (d), each leg is equipped with a spring-loaded elastic foot and in addition 3D force sensors. For navigation, inertial measurement units and a GPS receiver are implemented. Using different cameras (stereo, 360°, and time-of-flight cameras) mounted at the head and the back of the robot, information on the environment can be obtained. For the control, dedicated gait controllers are used for tri-, tetra-, pentapod, and free gait. In contrast to its predecessors, *Lauron V* uses four actuators per leg. This is intended to increase the workspace and give the robot the option to orient the tibiae normal to ground (Rönnau, Heppner, Pfotzer, et al. 2013).

The robots *Scorpion I-IV*, as the name implies, have eight legs. Unlike *Lauron*, the design of the Scorpion robots does not mimic the morphology of the biological model. Therefore, e.g., the alignment of the joint axes and the number of DoFs (Degrees of Freedom) differ between the legs of the robots and a natural scorpion (Bowerman and Root 1978). The controller is based on CPGs (Central-Pattern Generators) that have been expanded by posture control and reflexes. Modifying the activation level of the CPGs, different gaits can be elicited (Spenneberg et al. 2007). Whereas the *Scorpion*-line robots have eight legs with three DoFs per leg, their six-legged successor *SpaceClimber* (shown in fig. 1.3 (c)) features four actuators per leg and spring-loaded dampers in the tibiae (Bartsch, Birnschein, F. Cordes, et al. 2010).

Other robots besides *Lauron* that were modeled after the physiology of stick insects are the *TUM robot* (Pfeiffer 2007) and *Tarry* (Amendt and Frik 1994). On these robots, the insect-inspired coordination rules found by Cruse (1990) were implemented (see chapter 4). In *Tarry IIb*, for the first time, local positive velocity feedback was tested on a legged robot as a concept for stance speed control (Schneider, Cruse, et al. 2006). For this purpose, the legs were equipped with springs that added serial compliance to two of the joint drives. This robot was also used to test all components of Walknet, a bioinspired controller consisting of distributed ANNs for swing, stance, targeting, etc. (Schmitz, Schneider, et al. 2008). Details on Walknet are given in section 1.2.2 and chapter 4.

1.2. Gait Control

The appeal of legged robots is their adaptability to maneuver in uneven terrain as compared to wheeled robots due to the high number of DoFs of the legs. However, this also increases the difficulty in the coordination of movements to prevent the robot from falling and subsequent crashing. In general, two kinds of stability can be defined: *static* and *dynamic stability* (E. Garcia et al. 2002). *Static stability* is characterized by a support region that is created by at least three feet in contact with the ground. The projection of the COM along the vector of gravity must be within this support region. If the COM leaves the support region, the robot will topple. On flat terrain, the support region is identical with the convex hull of the ground contacts. In general, e.g., on curved surfaces with slanted foot contacts, the support region does not need to coincide with the convex hull of ground contacts. In extreme cases, it might even be completely outside of the convex hull of ground contacts (Bretl and Lall 2008). For the computation of the actual support region, the friction at the ground contacts must be considered together with the normal of each contact. Since this requires considerable knowledge about the environment and the properties of the ground, this approach is not universally applicable. Within this work, non-slippery contacts will be assumed for which the support region coincides with the convex hull of ground contacts, further called support polygon. Thus, the goal is to keep the projection of the center of gravity at all times within the support polygon. To quantify the stability of a system, different measures were proposed, e.g., the horizontal distance of the COM to the closest line of the support polygon. To ensure the stability of moving robots, usually a safety margin is applied such that the robot can be stopped at any moment during the movement without endangering its stability.

In contrast, the controllers developed to maintain *dynamic stability* actively induce loss of static stability (Raibert 1986). Therefore, if the projection of the center of gravity is outside the support region and the robots starts to tilt, a corrective step must be performed that counteracts this tilting motion. Even if this corrective step does not restore static stability, the consecutive correction steps can be used to stabilize the robot such that it does not crash into the ground over an extensive period of time.

While most mammals use static stability for slow and dynamic stability during fast locomotion, it has been assumed for a long time that insects use exclusively static stability since they have six legs that can be comfortably divided into two pairs of tripods (Dean and Wendler 1984; Hughes 1952; McGhee and Sun 1974). However, newer experiments revealed that also some insects use dynamic gaits during fast locomotion, e.g., when trying to escape (Full and Tu 1991).

The stabilizing effects of insects' inherent compliance has been successfully adopted for the control of hexapod robots during fast locomotion (Cham et al. 2002). However, whereas bi- and quadruped robots continuously advance in dynamic locomotion, the majority of hexapod robots relies on the static stability criterion. This is especially important if the robot is to be used in unknown terrain in which a crash might risk its mission. Depending on the mechanical configuration of the robot, different gait controllers can be used for locomotion. The easiest in terms of implementation are the *fixed gaits* that merely replay a pre-defined gait pattern (see section 1.2.1). Based on biological research, *bioinspired gait controllers* have been developed that attempt to replicate the elegant leg coordination of insects (see section 1.2.2). A third category are *technical free gait controllers* that share the adaptability of the bioinspired gait controllers without relying on biological findings but rather on optimization algorithms (see section 1.2.3).

1.2.1. Fixed Gaits

Legged automatons, e.g., the *iron mule train* (Morrison 1968), used fixed mechanisms to coordinate the legs. However, also modern robots often use fixed, preprogrammed gaits due to their simplicity and reliability. If using multiple actuators—unlike the *iron mule train*—the type of fixed gait can be adapted. In *Rhex*, for example, the gait is adapted based on the environment (Haynes and Rizzi 2006): For plain surfaces, a tripod is used, whereas for stair climbing a metachronal wave gait is used. For many robots like the *OSU Hexapod*, *Lauron*, or *AMOS-WD06*, a gait selector is used to switch between predefined gaits (Berns et al. 1994; Klein and Briggs 1980; Steingrube et al. 2010). Also in the *DLR-Crawler*, fixed gaits are used as an alternative to the more complicated coordination concepts (Görner, Wimböck, and Hirzinger 2009) that will be presented in section 1.2.2.

An example for a mechanically fixed, yet adaptable gait generation is realized in *Whegs* (Allen et al. 2003). It uses a single motor to actuate all of its six wheel-leg-hybrids. However, the legs are coupled via compliant springs that enable variances in its gait pattern and allow it to climb over obstacles.

1.2.2. Bioinspired Gait Control

In contrast to fixed gaits which are used in many technical solutions for gait generation, insect locomotion differs from these idealized gaits. Usually, a tripod-like coordination pattern is used for fast, a tetrapod-like for medium, and a metachronal wave gait pattern for slow walking. However, the transitions between these states are continuous (Cruse 1980b; Strauß and Heisenberg 1990; Wosnitza, Bockemühl, et al. 2013; Zollikofer 1994a). This fluent speed-dependency of gait patterns has been summarized as *free gait* (Cruse, Dürr, et al. 2007).

Several different models and rule sets have been designed to explain the inter-leg coordination in insects. Probably the first attempt to formalize the coordination of hexapods was made by von Holst (1935). He reported that nearly all insects—including those with amputated legs—follow the two rules

1. adjacent legs of one side alternate in stance and swing,
2. diagonal legs work synchronously.

This, however, restricts the coordination to strict tripod walking. Even in a less strict interpretation that would include tetrapod walking, most transient coordination patterns gaits are excluded by this definition. In 1952, Hughes suggested two coordination rules that incorporate most coordination patterns, including tetrapod and wave gait:

1. “No fore or middle leg is protracted until the leg behind has taken up its supporting position.”

2. “Each leg alternates with the contralateral limb of the same segment.”

However, some insects show step patterns in which the metachronal wave travels first on one side of the body from rear to front and then on the other side (Bert 1866; Hughes 1965) and some insects use contralateral leg pairs synchronously (La Greca 1943) which both contradict rule 2. Therefore, Hughes assumed rule 1 to be more significant (Hughes 1965). Yet, since some insects lift their hind legs permanently during walking (Wille 1924), also the first rule is not strictly obeyed by all insects. Wilson (1966) suggested to concretize Hughe’s two rules (see also Hughes and Mill 1974) based on results by Wendler (1964) on stick insects by adding three more rules:

3. “Protraction time is constant.”

4. “Frequency varies (retraction time decreases as frequency increases).”

5. “The intervals between steps of the hind leg and middle leg and between the middle leg and foreleg are constant, while the interval between the foreleg and hind leg steps varies inversely with frequency.”

As pointed out by Burns (1973) and Delcomyn (1971), the constancy of protraction time is not valid for all insects at all walking speeds. Still, adult stick insects, in particular *Carausius morosus*, which are among the most studied insects regarding hexapod locomotion, keep the swing duration nearly constant at all speeds (Wendler 1964).[3] Although the rules do not explain the step patterns of all insects, they at least describe the step patterns of stick insects to a certain extend.

Wilson (1966) suggested that the mechanisms that coordinate the legs could be modeled by coupled relaxation oscillators of which each represents one thoracic ganglion. The coupling between the oscillators might be either excitatory or inhibitory. Having slightly different resonance frequencies for the three leg pairs with the hind leg oscillators having the highest frequency, a weak coupling would force the oscillators to operate at the same frequency but with a suitable phase shift to comply with the coordination rules. This approach was experimentally tested by Wendler (1968) using an analog computer and six relaxation oscillators—one oscillator per leg. A schematic of the oscillator network is depicted in fig. 1.4 (a). The oscillators are coupled contralaterally in a bidirectional way and ipsilaterally in a unidirectional way from back to front (in each pair of ipsilateral oscillators, the posterior oscillators acts as *controlling oscillators*, the anterior oscillators as *controlled oscillators*). With this setup, Wendler was able to simulate different coordination patterns. A similar concept was presented by Graham (1977) as shown in fig. 1.4 (b). In this model, the six legs are represented by six oscillators that are coupled

[3] In addition to the constant swing time shown by adult *Carausius morosus*, the first instar form also shows tripod patterns in which the swing duration is inversely proportional to the walking speed, thus adapting it to all walking speeds (Graham 1972). As a result, a tripod gait is used over the entire speed range.

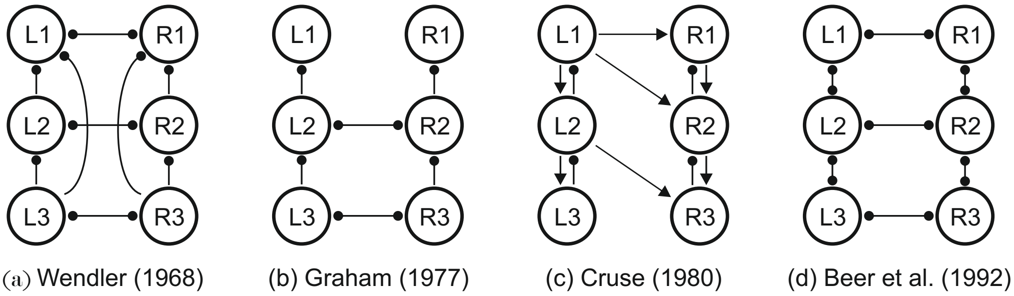


Figure 1.4.: (a,b,c) Hypothetical coordinating influences between the leg controllers of stick insects (Cruse 1980b; Graham 1977; Wendler 1968) and (d) exemplary implementation on a hexapod robot (Beer et al. 1992). Excitatory influences are symbolized by arrows, inhibitory influences by dots.

by inhibitory influences from back to front and in addition contralaterally for the hind and middle legs.

In 1980, Cruse presented a concept, in which an excitatory influence was assumed to work ipsilaterally from front to back, contralaterally from the front left leg to the front right leg, and also diagonally between the left front and middle legs to the right middle and hind legs (see fig. 1.4 (c)). Cruse hypothesized that a protraction might be initiated in the controlled leg if the controlling leg crosses a velocity-dependent threshold position during its stance phase.[4] Dependent of the localization of this threshold (and the configuration of the velocity-dependence), the diagonal connections were supposed to facilitate concurrent protractions of the diagonal leg pairs (consisting of controlled and controlling legs), whereas all other connections were supposed to facilitate alternating protractions of the respective leg pairs. In addition to the excitatory influences, inhibitory connections were assumed between ipsilateral neighboring legs from back to front.

A more detailed model named WALKNET (see chapter 4) was presented by Cruse in 1990. This model combines different reactive coordination rules with neural networks for the leg control to simulate bioinspired locomotion in a bottom-up approach.

For the application in a hexapod robot, the concepts were adapted by Beer et al. (1992). As fig. 1.4 (d) shows, the network resembles the concepts of Wendler and Graham with bilaterally inhibiting connections between all neighboring legs. This network corresponds to Hughes' rule 1.

All of these models assume individual controllers for each leg that are merely coupled by excitatory or inhibitory influences. In fact, experiments on stick insects have shown that each leg has its own controller close by: If the nerves between head and body are cut, the legs still perform the stance-swing-cycles as before although no connection to the CNS (Central Nervous System) exists (Graham 1979). However, if the nerves are

[4] This influence was later included in WALKNET as *rule 3* (see chapter 4).

cut between the body segments, the coordination pattern changes substantially (Dean 1989). Therefore, some kind of decentralized coupling between ipsilateral legs seems to exist that supports coordination.

Due to the rhythmic motions of the legs, usually, oscillators are assumed as leg controllers. However, the type of input varies throughout the different concepts. Early simulation models, e.g., by Wendler (1964), were based on a pure CPG approach, therefore without sensory input. Current models that use CPGs usually incorporate sensory information into the controller framework. As such, the oscillators can be influenced by the current state of neighboring leg controllers (stance/swing) and the current position (Beer et al. 1992; Delcomyn 1999; Manoonpong et al. 2008). On the other side of the spectrum, reactive systems can be classified. These models, such as Walknet (Cruse 1990), do not require a dedicated pattern generator. Instead, the sensory information is used to trigger reactions. For example, the position of the leg tips (approximated by an ANN based on the joint angles) is used to trigger the switch between stance and swing phase. In turn, the detection of ground contact at the end of swing phase triggers the start of the stance movement.

1.2.3. Technical Free Gait Controllers

One of the advantages of the bioinspired gait controllers is their general adaptability, e.g., to different walking speeds but also to disturbances of the stance-swing-cycle due to uneven terrain. However, also technical approaches that are not based on biological models show similar properties.

Salmi and Halme (1996) developed a free-gait algorithm for application on the six-legged robot *Mecant I*. They defined the LPS (Leg Phase State), a six-component vector that represents the durations until the legs are assumed to switch between swing and stance. By definition, the estimated time until a swinging leg reaches its target position is represented by negative value. Correspondingly, the time a leg could stay in stance phase until it reaches the limit of its workspace is represented by positive values. During each control cycle, the leg with the lowest positive LPS-component is identified. If the leg would reach the limit of its workspace within the next control cycle, it is lifted for swing. To prohibit instabilities, a *gait state* is selected that ensures stability. For this purpose, the legs with negative LPS-values close to zero (swinging legs that are close to their target position) are considered to be switched to stance phase. Of all possible *gait states* that can be achieved by this measure, the statically most stable is selected.

Derived from the concept of LPS, Alexandre and Preumont (1996) introduced a rule-based gait controller for the hexapod *ULB walking machine*. The state of the whole system can be represented by six phase diagrams, in which the LPS-components of consecutive legs are plotted against each other. Analyzing these diagrams, Alexandre and Preumont (1996) showed that states represented by some regions within these diagrams are likely to cause instabilities (e.g., both legs in swing phase) whereas others provide static stability. Based on this analysis, they defined rules that lead to avoidance of the instable LPS-regions, facilitating statically stable, speed-adaptive gaits.

1.2.4. Restrictedness-Driven Free Gait Control

Conceptually, the restrictedness-driven gait controller by Fielding and Dunlop (2004) is similar to the previously described technical gait controllers: If a leg gets close to the limits of its workspace, it will be lifted. By introduction of the restrictedness R_i, an artificial measure that indicates the "lack of freedom" (as coined by Fielding and Dunlop) of the i-th leg, a flexible way for the definition of the workspace limits was established. R_i may be composed of different sources of restrictedness, such as the angle and torque limits of the actuators. $0 \leq R_i < 1$ indicates that the leg tip is within its desired workspace, therefore, currently not restricted. As a consequence, the leg can be moved in every direction. $R_i = 1$ indicates that the leg tip has left the permissible workspace and its movement is therefore completely restricted. To maintain the mobility of the leg, the leg controller must therefore ensure that the restrictedness stays in the interval $[0, 1[$. As mentioned, the overall restrictedness R_i can be composed of multiple functions $R_{i,k}$ that represent different constraints. Based on n of these restrictedness-components, the overall restrictedness of the leg can be computed by

$$R_i = \sum_{k=0}^{n} s_{k,m} R_{i,k} \tag{1.1}$$

with factors $s_{k,m} \in \{0, 1\}$ that define which components to include in the summation. The introduction of $s_{k,m}$ allows to include/exclude restrictedness components depending on the state m of the leg, therefore, whether it is in swing or stance phase.

Beside the mentioned angle and torque limits of the individual joints, additional restrictedness components were introduced to let the leg tip avoid the singularity along the axis of the α-joint (see section A.4) and to avoid collisions with other legs by including the proximity to the tips of ipsilaterally neighboring legs into the overall restrictedness.

Based on the restrictedness value, the transition from stance to swing phase is triggered. If the restrictedness of a leg exceeds a certain threshold—therefore if the leg is close to the limit of its workspace—the leg lifts and swings to a less restricted position. In order to prevent instabilities if adjacent legs swing at the same time, a similar inhibitory rule as proposed by Beer et al. (1992, see fig. 1.4 (d)) was introduced that prohibits the stance-swing transition of a leg as long as one of its direct neighboring legs is in swing phase.

The selection of a suitable target point for the swing phase was solved by selecting a *home position* in the workspace of the leg. Ideally, this point should be crossed by the stance trajectory. However, since the stance trajectory is not known a priori, for the planning of the target point it is assumed that the current direction of movement will not change. Therefore, in order to define a swing target point, starting from the home position going in the opposite direction of the stance movement, a point will be chosen that is at the desired distance from the *home position*. In order to stay within the leg workspace during swing phase, the leg's restrictedness is monitored during the movement and compared to the expected change of restrictedness. If the restrictedness rises faster than expected, the leg is lowered in order to end the swing phase.

In chapter 5, the concept of restrictedness will be combined with the coordination rules of WALKNET for the bioinspired, omnidirectional control of HECTOR.

1.3. Compliant Actuation

In position-controlled robots, the inherent compliance of the mechanical structure is usually considered disadvantageous since, in general, it reduces the positioning accuracy and the possible bandwidth of control. As a consequence, robotic structures have been reinforced to reduce this influence (known as "the stiffer, the better" according to G. A. Pratt and Williamson (1995)).

However, due to the increasing need of direct cooperation between robots and humans, a need for safety precautions arose. One of the measures taken to increase the safety is a compliant behavior of the robot. This means that the robot does not try to reach its desired position at all costs. Instead, it compromises between the goal to reach the position and the force that pushes it away from its target. Thus, if the robot collides with a human co-worker or a part of its environment, the force that will act between the robot and the obstacle is reduced. Fatal injuries like the one described by Collins et al. (1985), in which a human was crushed between a robot and—ironically—a safety pole, can be prevented by this measure.

In addition to the increase of safety, compliant robots have also shown advantages over stiff manipulators in tasks that require the manipulation of a constrained object (Yoshikawa and Zheng 1993). Often cited examples of such tasks are polishing, grinding, and deburring as well as peg-in-hole tasks (De Schutter et al. 1998; Liao et al. 2008; Mason 1981).

For legged robots, all legs that are in contact with the ground form closed kinematic chains. Thus, legged locomotion is similar to the manipulation of an object by multiple robot arms. Due to this similarity, the same mechanical and control concepts that have been developed for robot arms, in particular regarding compliance, can be utilized also for the control of legs.

The group of compliant actuators can be divided into the subgroups of *actively compliant* and *passively compliant* actuators. *Actively compliant* actuators use sensors for the measurement of forces/torques and based on these measurements, they adapt the desired target position. However, the underlying mechanism is still stiff (beside the unwanted, inherent compliance). Since a controller is needed for the simulation of the physical compliance, the reaction bandwidth is limited by the control frequency. During high-speed impacts or high-frequency disturbances, the robotic structure will therefore still appear stiff. Thus, the movements must be either slow or more elaborate control concepts must be used that are able to anticipate disturbances like leg touch-down (Byl et al. 2009). Examples of legged robots that use actively compliant actuators are the *DLR-Crawler* (Görner, Wimböck, Baumann, et al. 2008), *LittleDog* (Byl et al. 2009), and *COMAN* (Dallali et al. 2012).

The other concept, *passively compliant* actuators, utilizes inherently compliant mechanisms such as steel springs in the drive train. Also, different implementations of pneumatically actuated artificial muscles have been developed (Daerden and Lefeber 2002). The advantage of these actuators in comparison to the actively compliant actuators is their bandwidth-independent elasticity. However, due to the additional mechanisms, passively compliant actuators tend to be heavier. Especially in legged robotics, the bandwidth-independent compliance is beneficial since it protects the gearboxes from torque peaks during collisions, and simplifies the posture adaption to the terrain. The increased mass, however, is a considerable disadvantage for use in legged robots which might explain their minor appearance. Besides the advantage of terrain-adaptation, passively compliant actuators can also help to reduce the energy consumption by temporarily storing some of the kinematic energy in the springs, e.g., during hopping. However, to fully exploit the energy-storing, either the gearbox must be non-backdrivable[5,6] (Andeen and Kombluh 1988; Sensinger and Weir 2005) or the drive-side of the actuator must be equipped with a locking device (for examples see Plooij et al. 2015). For the relatively slow movements of the stick insect - inspired robot presented in this work, the energy-storing aspects are considered insignificant.

G. A. Pratt and Williamson (1995) were among the first to suggest the use of a serial elasticity in the drive train of robotic actuators. As in the original SEA (Series Elastic Actuator), the compliance is often added by a linear or torsional steel spring. In current passively compliant actuators such as the actuator presented by Tsagarakis et al. (2009), this mechanism was optimized regarding size and mass but the overall concept remains the same. A further advancement in the development of compliant actuators are VSAs (Variable Stiffness Actuators) and VDAs (Variable Damping Actuators) (for an overview see Vanderborght et al. 2013). These types of actuators feature tunable mechanisms that allow the task-dependent adaptation of stiffness and/or damping. Different concepts exist for the realization of the underlying mechanisms but since some kind of additional actuator is needed to tune the stiffness/damping they tend to be even heavier than SEAs.

Examples for the application of SEAs in legged robots are *StarlETH* (Hutter et al. 2012), *SpringFlamingo* (J. E. Pratt and G. A. Pratt 1998), *Veronica* (Van Ham et al. 2007), and *Tarry IIB* (Schneider, Cruse, et al. 2006).

1.4. Outline

In this work, the design of the robot Hector will be presented that is based on the morphology of the stick insect *Carausius morosus*. In **chapter 2**, an overview of the robot setup including the compliant joint actuators, the simulation and the communication frameworks is given. Although the morphology of the robot is inspired by a stick insect, several differences between the robot and its biological model remain. As the robot is to be controlled using an insect-inspired concept for leg coordination, the suitability of this concept for direct adoption in robot control is evaluated in **chapter 3** based

[5] *Non-backdrivable* means that application of an external torque at the output of the gearbox does not result in a movement of the input side.

[6] In order to act as energy storage the compliance must be located in the drive train at the output side of a non-backdrivable gearbox.

on available kinematic data of walking stick insects (Theunissen et al. 2014). Next, in **chapter 4**, the bioinspired coordination rules as deduced from forward walking stick insects (Cruse, Kindermann, et al. 1998; Schilling, Hoinville, et al. 2013) are evaluated regarding their ability to maintain the static stability of HECTOR during forward walking at two different speeds. As the data used for the development of the coordination rules was based only on forward walking stick insects, these concepts had to be adapted for the control of an omnidirectionally moving robot. The required modifications to allow for omnidirectional walking under consideration of the differences between insect and robot are described in **chapter 5**. To achieve similar results regarding gait generation as the classical WALKNET, the parameters that influence the leg coordination in the new, omnidirectional controller must be tuned. The process of walking optimization and corresponding simulation results for different walking scenarios are presented in **chapter 6**. Finally, in **chapter 7**, omnidirectional walking and terrain adaptation are presented using the real robot. The thesis concludes with a summary of the results and an outlook for future improvements in **chapter 8**.

2. Robot Design

Abstract: *The design of the robot* Hector *is supposed to reflect the scaled-up morphology of the stick insect* Carausius morosus. *Relevant properties of the insect that are to be adopted in the robotic setup are identified. However, due to the scaling and the technical limitations, several differences between the robot and its biological model remain. In order to transfer the compliant actuation of the insect's muscles to the robot, compliant joint drives for the use in the robot legs are designed with a focus on small size and low mass. Combining the six compliant legs with an exoskeleton-like body structure, the hexapod robot* Hector *was assembled. To simplify controller development, a simulation environment for* Hector *based on a physics engine was developed. Also, a communication framework was designed that allows for easy exchangeability of virtual and real robot.*

Some of the results and outcomes presented in this chapter have already been published in the following papers together with additional results from the specified co-authors: Paskarbeit, Annunziata, Basa, and Schneider 2013; Paskarbeit, Annunziata, and Schneider 2013; Paskarbeit, Schmitz, Schilling, and Schneider 2010a,b; Schneider, Paskarbeit, Schäffersmann, and Schmitz 2011, 2012

The robot, whose development is described in this work, is supposed to operate as a testbed for experimental bioinspired controllers. A key aspect in the transfer between biological model and technical system is the choice of relevant properties that can be reproduced after abstraction and despite the different side conditions in a technological implementation. Since the leg coordination in different walking situations is planned to be solved using the coordination rules found in stick insects (see chapter 4), the robot should combine geometrical aspects of the biological example which are relevant for walking with an adapted mass distribution, elasticity and lightweight construction.

2.1. The Biological Model (*Carausius morosus*)

The stick insect *Carausius morosus*[1] was first described by de Sinéty (1901). Native in southern India, it lives predominantly on branches of shrubs and trees, nourishing from the leaves (MacBride and Jackson 1915; Meissner 1909). *Carausius morosus* is often used as experimental animal since it is easy to rear, rather phlegmatic and therefore

[1] Originally, it was named *Dixippus morosus* (MacBride and Jackson 1915; Von Wattenwyl and Redtenbacher 1908).

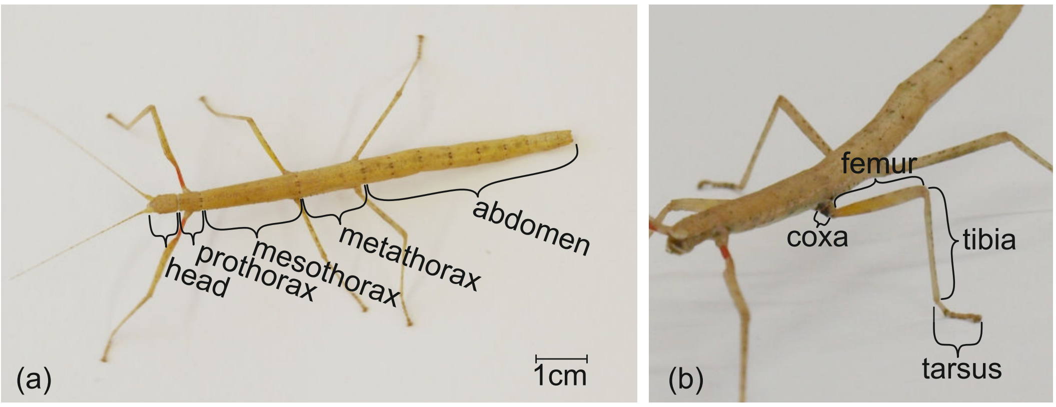


Figure 2.1.: (a) *Carausius morosus* in top view. The body consists of the three tagmata: *head*, *thorax*, and *abdomen*. The thorax is subdivided into the three segments *pro-*, *meso-*, and *metathorax*. (b) *Carausius morosus* in oblique view with labeled leg segments for the left middle leg.

easy to handle. Also, with a normal body length of 7-8 cm of females[2] and ~5.5 cm of males (Bauchhenß 1971), and the sprawled leg posture during walking, the insects are easy to monitor during experiments (Wendler 1964).

As all insects, *Carausius morosus* has a chitin-based exoskeleton that consists of three tagmata: *head*, *thorax*, and *abdomen* (Bässler 1983). Regarding potentially walking-related sensorization, the *head* is equipped with two antennae and two eyes. The *thorax* consists of three segments: *prothorax*, *mesothorax*, and *metathorax* (see fig. 2.1). The six legs of the insect are distributed pairwise onto the three thoracic segments with the onsets of the legs, the proximal attachment points of the legs at the body, closely located to the posterior end of each segment. Since the body segments are not connected rigidly with each other, they can be bent both horizontally and vertically relative to each other. The joint between meso- and metathorax has been tested to be passively bendable up to 40°. During normal behavior, however, only angles up to 30° have been observed for upward and 20° for downward bending (Cruse 1976a).

The *abdomen* consists of 11 segments. Since it is not equipped with legs, its role in walking is negligible. However, it may be used as additional support in order to prohibit backward tilting. Since the abdomen is roughly as long as the rest of the body, the overall COM of the insect is located close to the onsets of its hind legs (Cruse 1976b).

Insect legs consist of five limbs: *coxa*, *trochanter*, *femur*, *tibia*, and *tarsus*. These limbs are connected via joints that are named after the segments they connect (*thorax-coxa*-joint, *coxa-throchanter*-joint, etc.). In *Carausius morosus trochanter* and *femur* are rigidly merged to the so-called *trochanterofemur*. The *thorax-coxa-*, *coxa-trochanter-*, and *femur-tibia*-joints are formed as hinge joints. The axes of these hinge joints are

[2] In breeding cultures, *Carausius morosus* seems to reproduce nearly exclusively parthogenetically (Pijnacker 1966). Therefore, most publications refer only to females.

oriented as shown in fig. A.3. For simplicity, subsequently they will be called α-, β-, and γ-joint (*thorax-coxa-*, *coxa-trochanter*, and *femur-tibia*-joint, respectively).

The rotational axis of the first leg joint (α-axis) is not oriented vertically but slanted, which can be expressed by a rotation about two angles, in general known as ϕ and ψ angles (see figs. A.2 and A.3).[3] Due to this slanted onset, the function of the joints cannot be separated as it is possible, e.g., for the joints in the *Scorpion* robots (Spenneberg et al. 2007): The α-joint moves the leg back-and-forth, the β-joint moves up-and-down. Instead, a movement of the α-joint will also affect the vertical position of the leg tip. Correspondingly, a movement of the β-joint will also change the horizontal position.

The *tibia-tarsus*-joint is *passively* bendable in all directions. However, *actively*, the *tarsus* can only be raised and lowered relative to the *tibia*. It consists of multiple segments, which are equipped with adhesive pads that enable the insect to cling to smooth surfaces (Bässler 1983; Bullock et al. 2008). In addition, the insect has claws at the tip of the *tarsi* that allow it to cling to rough surfaces (Bässler 1983).

To sense the angles of the leg joints, the leg segments are equipped with hair plates in the cuticle close to the joint. A change of the joint angle will change the bending of the hairs, which is detected by sensor cells at the base of the hairs (Mittelstaedt 1950; Wendler 1964). For load-sensing, the insect's legs are equipped with *Campaniform Sensillae*, which are embedded in the cuticle and detect strain in the exoskeleton (Zill et al. 2004).

2.2. The Transfer to a Technical System

As it is currently technologically impossible to replicate the stick insect in all its particulars, compromises must be found between the level of detail and the technical effort that is needed to achieve it. Striking differences between stick insects and current insect-inspired robots are:

materials The exoskeleton of insects consists mostly of chitin. Using biomineralization, the exoskeleton is hardened locally where needed (Weiner and Dove 2003). At other positions, it is flexible to allow for the movement of the limbs. Compared to this extremely optimized production process, technical systems are rather simplistic. First approaches using tailored fiber placement allow for a local strengthening of a structure by stitching with carbon fibers (Uhlig et al. 2013). Thus, the stability can be increased locally at a minimal increase of mass.

size/mass Although stick insects are among the bigger insects, they are still much smaller than most current robots.[4] Robots of equal or even smaller size exist (Haldane et al. 2013; Hoover et al. 2008), yet, they tend to be simpler regarding their sensorization.

[3] In some older publications (e.g., Cruse (1976b)), these angles were named α and β. To avoid misconceptions regarding the leg joints, in this work, they will always be referred to as ϕ and ψ.

[4] Actually, the longest currently known insect is a stick insect of type *Phobaeticus chani* with a body length of 357 mm (Hennemann and Conle 2008)

Directly coupled to the size is the mass of the system since bigger systems tend to be heavier. Due to the different materials and construction also the mass distribution and therefore the location of the COM will likely differ.

energy supply Whereas insects digest other organisms to gain energy, robots are usually supplied by some kind of electronic power source or combustion engine. However, some robots also produce energy by digestion (Wilkinson 2000).

actuation All animals are powered by some kind of muscles whereas robots typically use electric motors for propulsion. One of the most relevant differences between biological muscles and classical, technical drives regarding their effect on the connected structures is the compliance of the muscle fibers. Due to the non-linearity in the spring characteristics, the resulting stiffness of the connected joint can be varied by co-contraction of antagonistic muscles (Zakotnik et al. 2006). In contrast, mechanical drives are usually engineered to be as stiff as possible, and lack the ability of stiffness regulation. Also, compared to mammals, the damping in the joints of insects is very high (M. Garcia et al. 2000), which is rarely replicated in robots.

sensorization The general concept in nature seems to be the use of a multitude of sensors that each have a comparably low resolution. However, by sensor-fusion of a high number of these low-resolution sensors a sufficiently precise measurement can be obtained. For example, in the legs of *Carausius morosus*, the joint angles are detected by fields of 15-30 hairs that are bent depending on the flection of the joint (Bässler 1983). The defect of one of these sensors reduces the accuracy of the overall measurement but the remaining sensors may still produce results that keep the animal operable. In robotics, usually, a small number of high-precision sensors (e.g., angle encoders) is used. In systems that are not explicitly designed to be redundant, the defect of a single high-precision sensor will likely result in a black-out of the whole system.

The focus of the project is centered on the transfer of locomotion and sensor-processing concepts from insects to a technical system. To use the bioinspired control approaches in the robot controller as they are deduced from walking stick insects, the robot needs six fully-articulated legs with three degrees of freedom each, and the legs should replicate the morphology of the insect legs regarding the length ratios of the leg segments. As the leg kinematics of the insects exhibits a singularity along the α-axis (see section A.4), a slanted mount of the legs as found in stick insects is also practical since this swivels the α-axis and therefore the singularity inwards, away from the main workspace of the legs.

To keep the mechanics as simple as possible, a reliable electric power supply is favorable although it is by no means bioinspired. The actuation mechanism should be reliable and simple as well, however, the compliance of the muscles is a feature that reduces positioning problems considerably during walking over irregular terrain: The legs will compensate for small positioning errors on the hardware-level without the need of immediate intervention by the walking controller. The actual implementation of the compliance is considered to be of minor relevance. A muscle-like compliance would be beneficial for direct adaption of insect-inspired joint controllers but it is not strictly necessary in this project since the focus of research will be on higher-level control like leg coordination rather than single joint control.

The robot must be able to sense the leg's joint angles and it needs a reliable way to sense collisions of the leg during swing phase—either with the ground at the end of the swing phase or with obstacles.

The width of the robot should not exceed 80 cm to allow operation in buildings—in particular passing through standard doors. Since the robot will be used as platform for navigation-related sensor systems such as cameras for obstacle avoidance, it must be able to carry a payload and the sensor systems must be mountable on the robot, which defines the lower size limit of the robot.

2.3. Leg Actuator Design

As described above, the locomotor system is the main focus in the development of the robot. Therefore, several requirements in the design of the leg actuators were crucial for the operation of the robot:

high torque/mass ratio In human-inspired biped and most mammal-inspired quadruped robots, the weight of the body is supported by the structure if the knee-joints are straightened. However, since the robot is to be modeled after the stick insect which exhibits a sprawled leg posture (see fig. 2.1), the weight of the robot must be actively supported even during standing. With six legs and three actuators per leg, the robot will contain 18 joint drives in the locomotor system. Thus, the weight of the body segments including the control electronics and batteries is likely to be exceeded by the overall joint weight. Therefore, the maximum torque the drives must be able to exert depends mostly on their own weight. As a consequence, a high torque to mass ratio of the actuators is crucial for the robot design.

small size To achieve a comparable range of movement as the insect legs, the drives should be as small as possible. Due to the small length of the coxa in the legs of the stick insect compared to femur or tibia, the α- and β-joint drives of the robot will be located close to each other. Therefore, a small diameter of the drives is desirable. Also, since the leg onsets of contralateral legs are located nearby, the housings of β-joints of contralateral legs might collide with each other for extreme leg postures. To prohibit such collisions, the length of the β-joint drives should be as small as possible.

compliance Driven by muscles, the legs of stick insects show a certain degree of compliance. As mentioned, this is considered to be a relevant feature of bioinspired locomotion and should therefore be adopted in the design of the joint drives. An inherent compliance reduces the need for exact positioning of the feet and protects the gear boxes

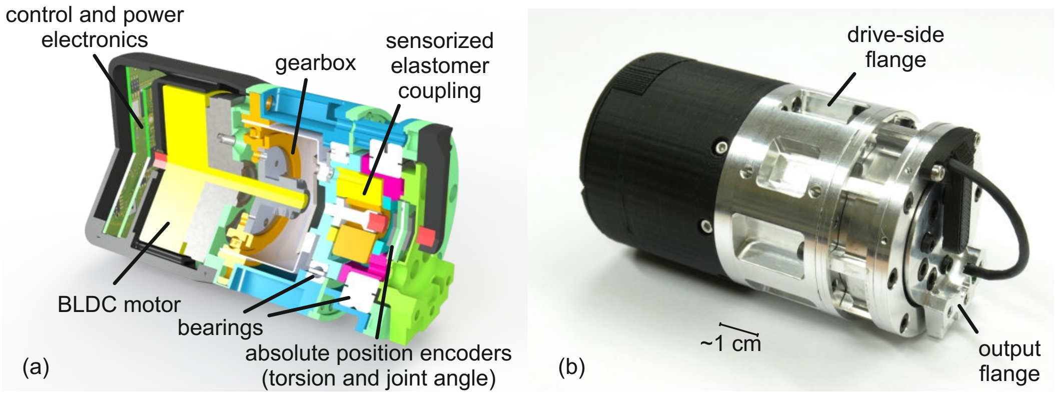

control and power electronics
gearbox
sensorized elastomer coupling
BLDC motor
bearings
absolute position encoders (torsion and joint angle)
(a)
drive-side flange
~1 cm
output flange
(b)

Figure 2.2.: (a) Colored section view of the final drive design. The connecting cable between the output-side position encoders and the control electronics is not shown. (b) Image of assembled joint drive. The bent cable at the output connects the position encoders with the control electronics in the back. The drive has a length of 90 mm and a diameter of 51 mm.

during collisions, e.g., the collisions between the leg tips and the ground at the end of the swing phases.

integrated control electronics To reduce the amount of wiring to the central controller, decentral control electronics should at best be integrated directly into the drives. Thus, the sensor signals can be preprocessed in each joint allowing for a reduced communication between the central controller and the periphery.

Due to these requirements, multiple design iterations were necessary to find a suitable combination of motor, gearbox, and compliant element. In the final design that is depicted in fig. 2.2, the actuators are powered by a BLDC (BrushLess DC) motor (EC 45 flat, 24 V, 50 W; maxon motor GmbH, Sachseln, Switzerland). The motor features an external rotor setup, which allows for a continuous torque of 83.4 mNm at a weight of 110 g. Beside the high torque/mass ratio, the motor was also selected due to its short length of 24 mm (41 mm including the axis) at a diameter of 43 mm.

As a consequence of the high torque/mass ratio and small size requirements, regular planetary gearboxes had to be precluded. Instead, a lightweight strain wave gear box was selected that features a gear ratio of 100:1 and a repeatable peak torque of 28 Nm at a mass of only 55 g (CPL-14-2A; Harmonic Drive AG, Limburg, Germany).

Between the output of the gearbox and the drive output, an elastic coupling was introduced into the drive train. This coupling consists of a deformable, star-shaped rubber inlet that is held between two metal hubs. One of the hubs is connected to the gearbox and the other is connected to the drive output. A torque that is acting on the output will therefore deform the rubber inlet, thus leading to a torsion of the coupling that can be measured by an integrated hall-effect based sensor (AS5245; ams

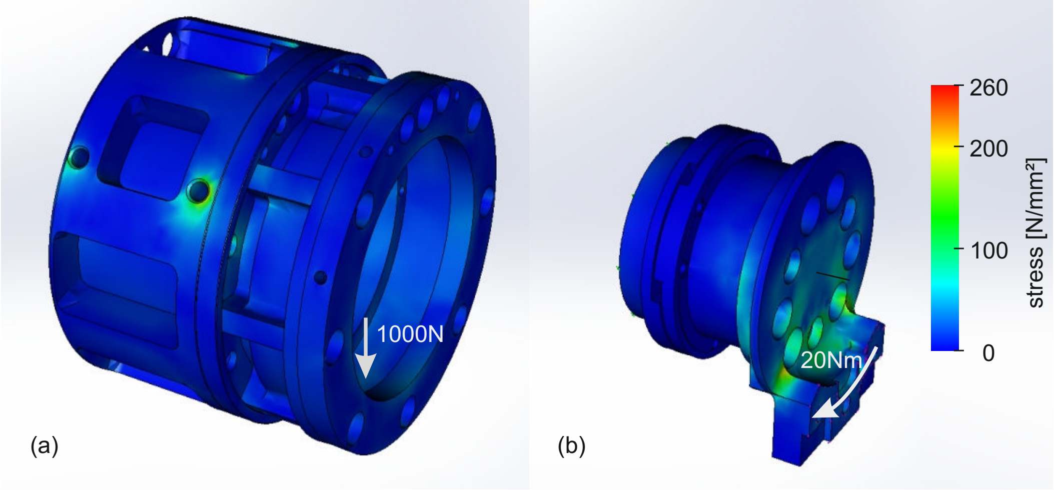


Figure 2.3.: Results showing the stress in the load-bearing parts of the actuator based on FEM (Finite Element Method) simulations. (a) Load case in which the housing is fixed at the drive-side flange and a bearing load of 1000 N is applied downwards at the output bearing. (b) Load case in which the input-side of the coupling is fixed and a torque of 20 Nm is applied at the output. Close inspection of the stresses shows that the stability of the material is not compromised.

AG, Premstätten, Austria) with a resolution of 12 bit. Another sensor, mounted back-to-back measures the output angle of the drive. A detailed description of the compliant coupling and the integrated sensors can be found in section 2.3.1.

To simplify the assembly of the robot, the drives use a cross roller bearing (RB2508UUCC0; THK, Ratingen, Germany) to support the output. This allows a direct connection of the drive to the adjacent leg segment without the need for additional bearings to absorb force and torque peaks. Since the deformation of the rubber inlet in the compliant coupling might produce forces that would act on the gearbox, a four-point-bearing (SAA10XLOK; Silverthin Bearing Group of Mechatronics, Inc., Preston WA, USA) was used to counteract these forces and protect the gearbox.

The control electronics was integrated at the input side of the drive close to the motor. It commutates the BLDC-motor and processes sensor data. Details on the electronics can be found in section 2.3.2. For the measurement of the torsion and output angles of the actuator a cable was routed from the control electronics toward the sensors within the compliant coupling (see fig. 2.2 (b)). To prevent shearing of this cable the angular workspace of the drive is limited to ±135°.

The metal parts of the actuator were manufactured from an aluminum alloy (EN AW-7075,5; Batz+Burgel GmbH & Co. KG, Friedberg, Germany). Using FEM simulations in SolidWorks (Dassault Systèmes, Vélizy-Villacoublay Cedex, France), the parts were optimized regarding low mass and high durability (see fig. 2.3). In the data sheet, the minimal offset yield point for the aluminum alloy is given as $R_{p0.2} = 260\,\mathrm{N/mm^2}$

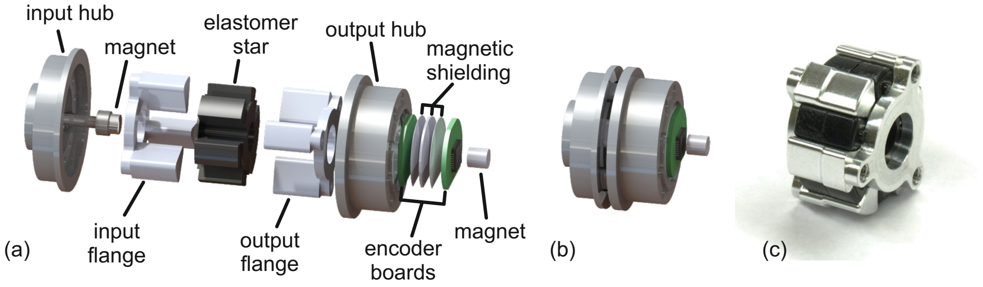


Figure 2.4.: (a) Rendered explosion view of the elastomer coupling assembly. (b) Rendering of the assembled elastomer coupling. (c) Image of the elastomer star with input and output hubs.

(Batz+Burgel GmbH & Co. KG 2016). According to the FEM simulations, the stress will not exceed this limit for the expected load cases. The housing of the actuator is constructed to withstand torques up to 40 Nm and forces up to 200 N acting between the drive-side flange and the output. To further reduce the mass of the actuator, the housing for the control electronics was 3D-printed from ABS (Acrylonitrile Butadiene Styrene). Thus, the overall mass of an actuator could be reduced to 0.39 kg. Since the diameter of the elastomer coupling allows for the integration within the inner rings of the bearings, an overall length of 90 mm at a diameter of 51 mm could be realized.

2.3.1. Elastomer Coupling

As for the complete drive, the major objectives in the design of the compliant element were a low mass and a small size. Usually, the compliance in SEAs is realized by torsional (G. A. Pratt and Williamson 1995) or linear steel springs (Tsagarakis et al. 2009). For the use of linear springs in a rotary drive, as presented by Tsagarakis et al. (2009), six linear steel springs were centered in a mechanism around a common axis. Another concept facilitates a hypocycloid mechanism (Thorson and Caldwell 2011). However, for the required degree of miniaturization these approaches were considered to be unfeasible. Thus, a custom design (shown in fig. 2.4) based on the principle of jaw couplings was favored. As depicted in fig. 2.5, the compliance is accomplished by the integration of a deformable elastomer inlay. Jaw couplings are commercially available from many manufacturers in different sizes. The material used for the elastic inlay is usually TPU (Thermoplastic PolyUrethane) or some other TPE (Thermoplastic Elastomer). Since elastomers show a much higher damping than steel springs, constant compression-relaxation cycles will result in heat development in the material. When heated, as their name suggests, Thermoplastic Elastomers become deformable, which results in the development of backlash in the coupling. To overcome this problem, a custom elastomer coupling was manufactured by *GKT Gummi- und Kunststofftechnik Fürstenwalde GmbH* (Fürstenwalde, Germany) using NBR (Nitrile Butadiene Rubber). The used material,

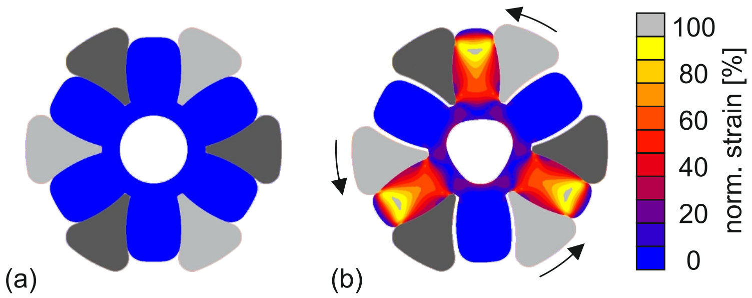


Figure 2.5.: FEM simulation results of an unbonded elastomer coupling (a) without and (b) with applied load. The colors in the section view of the elastomer depict the normalized strain of the material. Associated metal teeth of each flange are shown in the same color. In (b), due to the applied load, three of the elastomer lobes are compressed whereas the other three lobes do not deform since they are not bonded to the metal hubs.

SPP1838/81 (Gummiwerk KRAIBURG GmbH & Co. KG, Waldkraiburg, Germany), has a hardness of 82, Shore A and a low compression set of 4% (72 hours at 23°C). Another benefit of NBR is the option to chemically bond it to the metal faces of the hub. This would result in an increase of stiffness and would extinguish backlash even for prolonged use. A detailed analysis on the modeling of the elastomer coupling characteristics for the bonded version of the coupling was presented by Paskarbeit, Annunziata, Basa, et al. (2013). Due to problems with the bonding process, the maximal torque the coupling could withstand was limited to 10 Nm before the rubber detached from the metal flanges. Since this torque is not sufficient for the locomotion of the robot, currently, unbonded elastomer inlets are being used. As shown by Wright et al. (2012), a strong and durable bonding between elastomer and metal seems technically possible. Therefore, this approach should be further pursued.

Since the robot is equipped with unbonded couplings, the model for the bonded coupling (Paskarbeit, Annunziata, Basa, et al. 2013) is not applicable. However, to detect collisions with the ground or obstacles during the swing phase, a simpler model proofed to be sufficient. Figure 2.6 (a) shows the torque vs. torsion for cyclic load changes for the first version of the unbonded coupling. During the experiment, the coupling was repeatedly loaded up to ±16 Nm. Since the elastomer material has an inherent damping, the load curves show a hysteresis. The first version of the unbonded coupling was manufactured with the same design as the bonded coupling. However, since the NBR shrinks during the vulcanization process, a noticeable backlash can be observed in the hysteresis curves during load changes. As depicted in fig. 2.6 (b), using fits to the linear part of the loading curves, the amount of backlash was determined to be about 1.5°. In the final version of the unbonded coupling, this was considered in the design of the mold such that the vulcanized inlay even has a small oversize after the vulcanization, thus eliminating the backlash. For the estimation of the torque τ that acts on this backlash-

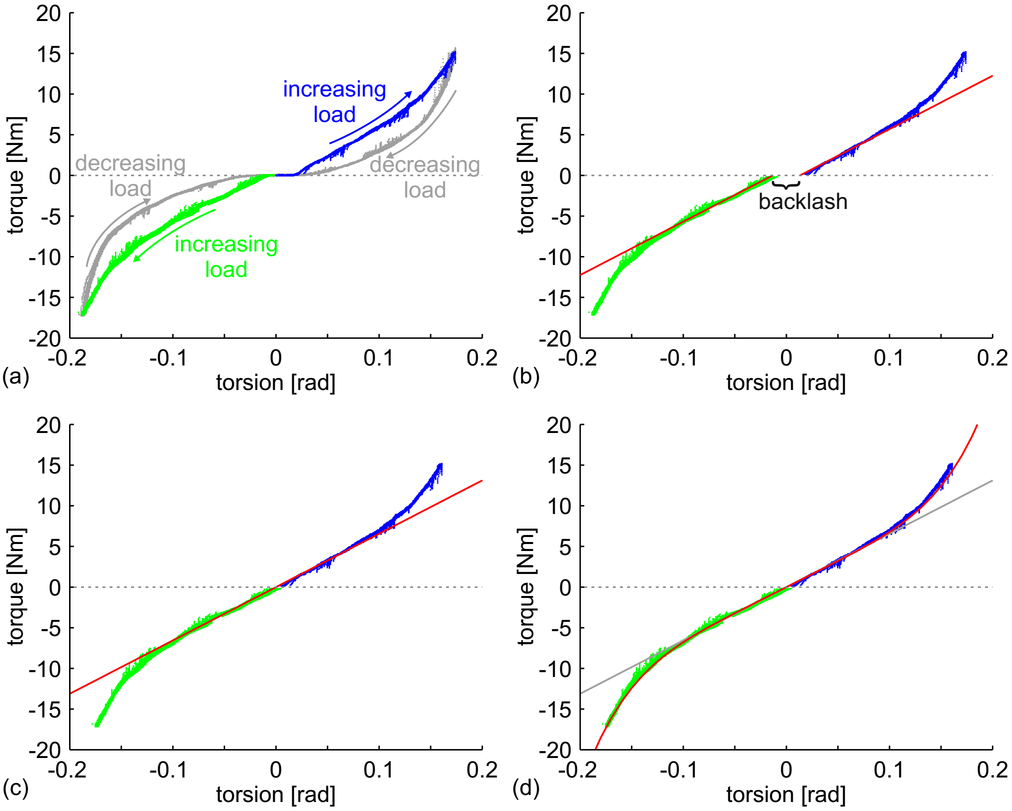


Figure 2.6.: (a) Hysteresis curve of the first version of unbonded elastomer couplings for several load cycles. The load cycle starts with an increase of the load (blue lines). During this load increase, only three of the six elastomer lobes are compressed by the flanges. At 16 Nm the torsional speed is reversed and the load is decreased (gray lines below the blue lines). Starting at a torsion of 0 rad and a torque of 0 Nm, the other triplet of elastomer lobes is being compressed (green lines) and the load increases again till -16 Nm. At this point, the load is decreased again (gray lines above the green lines) and the next load cycle starts. (b) Both load curves for increasing load. Since the coupling was manufactured without oversize, a noticeable backlash exists between the two curves. The angular width of the backlash is estimated by line fits (red) to the linear parts of the load curves (till ±0.13 rad). The line fits have an inclination of 65.59 Nm/rad. (c) Virtually merged load curves without backlash (assuming a backlash of 0.0265 rad $\approx$ 1.5°) (d) Virtually merged load curves with linear (gray, $\tau = 65.59\,\theta$ with the torsion θ) and fifth-order (red, $\tau = 36975.53\,\theta^5 + 64.05\,\theta$) fits.

compensated coupling based on the torsion θ, a fifth-order function was fitted to the data shown in fig. 2.6 (d):

$$\tau = 36975.53\,\theta^5 + 64.05\,\theta \tag{2.1}$$

Due to the hysteresis, the fit is only valid for increasing loads (in both directions). However, this is the relevant case, for which the torques must be estimated on a regular basis, e.g., to detect collisions of the leg with an obstacle (see section 5.4.3).

The torsion of the coupling is determined by a hall-effect based position encoder that is fixed to the output hub (see fig. 2.4 (a) and (b)). It measures the magnetic field of a cylindrical, diametrically magnetized permanent magnet that is fixed to the input side of the coupling. For the measurement of the absolute output angle of the drive, another angle encoder is mounted back-to-back to the torsion sensor. This encoder senses the magnetic field of an identical magnet that is fixed to the housing via a fillet (black element at the output side in fig. 2.2). Since the two sensors and their corresponding magnets are located close to each other, the fields of the magnets will influence both measurements. To reduce crosstalk, a magnetic shielding was introduced between the two sensor boards: The PCBs (Printed Circuit Boards) were separated by a laminate which consists of three layers of sheet metal; two outer layers of Netic S 3-6 (0.102 mm thickness) (Magnetic Shield Corporation, Bensenville, USA) and an inner layer of Co-Netic AA (0.635 mm thickness).

2.3.2. Integrated Control Electronics

The data acquisition and motor commutation is accomplished by an integrated control electronics. It is located at the input side of the drive, close to the BLDC motor and consists of two stacked PCBs. Images of the individual PCBs and the stacked assembly are shown in fig. 2.7. The PCB closest to the motor contains a hall-effect sensor (iC-MH; iC-Haus, Bodenheim, Germany) for the measurement of the motor angle and an 8-bit microcontroller (ATXMega128A1, Atmel Corporation, San Jose, USA) that is used for communication, sensor data acquisition, and to control the commutation of the BLDC motor.

The second board contains six N-channel MOSFETs (Metal–Oxide–Semiconductor Field-Effect Transistor; IRF6648; Infineon Technologies AG, Neubiberg, Germany), connected as three half-bridges for the commutation of the BLDC-motor. To drive the MOSFETs, a dedicated IC (Integrated Circuit; TMC-603A; Trinamic Motion Control GmbH, Hamburg, Germany) for BLDC-motor commutation is used that features a break-before-make logic and allows for current measurement via shunt resistors at the low-side MOSFETs.

For communication, an RS-485 transceiver is used. Due to the high currents in the motor coils and the resulting electromagnetic fields a differential communication was selected. On top of the serial interface a custom protocol named *BioFlex protocol* was implemented by Schäffersmann (2011) and Schneider et al. (2011). Key requirements in the development of the protocol were low CPU and memory usage. The communication framework will be discussed in more detail in section 2.6.

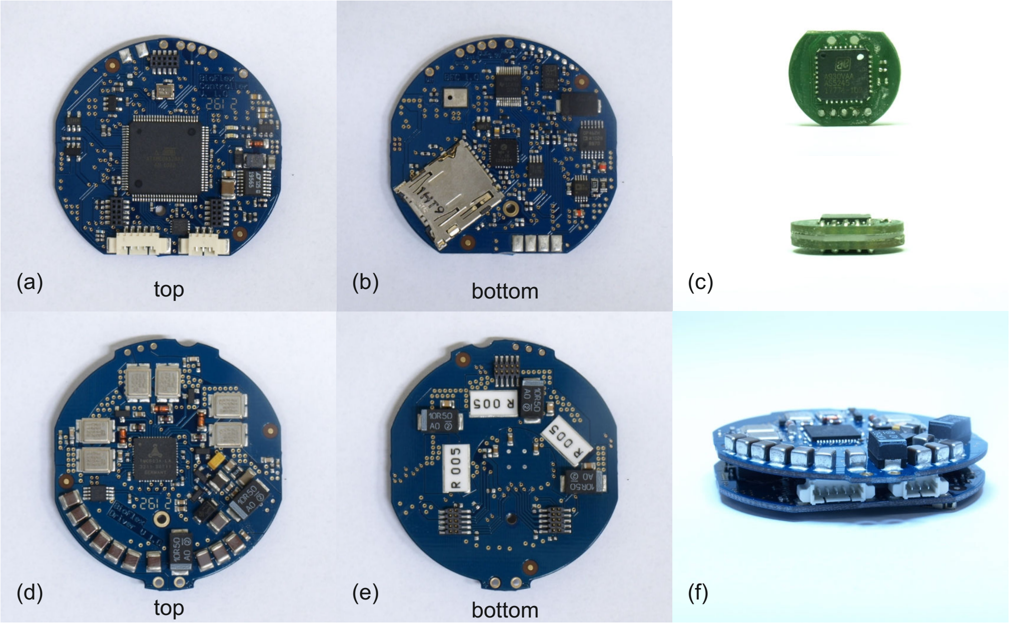


Figure 2.7.: Integrated drive electronics. (a) and (b) show the top and bottom view of the controller board. (c) sensor board stack for the measurement of the elastomer coupling torsion and the absolute output angle. (d) and (e) show the top and bottom view of the power board. The six rectangular elements are the six MOSFETs for the commutation of the three motor phases. The IC (Integrated Circuit) in the center of (d) is the MOSFET driver that is controlled by the microcontroller of the controller board. The three 5 mΩ shunt resistors on the bottom side (e) of the power board are used for the current measurement. (f) shows the electronics stack consisting of the controller and power boards. The stack has a maximal diameter of 48.5 mm and a height of 12 mm.

Different controllers have been implemented on the microcontroller that can be activated as necessary via the communication interface. To achieve smooth motions, usually the rotational speed of the motor is controlled by a rather simple PID-controller. However, to mimic the compliance of biological joint setups, e.g., an elbow joint with two muscles acting as flexor and extensor, also a more elaborate torque controller with friction compensation was implemented (Annunziata et al. 2013).

2.4. Hector— The Six-Legged Robot

Hector, the HExapod Cognitive auTonomously Operating Robot, is a walking machine based on the scaled-up model of the stick insect *Carausius morosus*. In its current state, it is actuated by 18 leg joint actuators, which are combined in groups of three to build up six legs. The main body of the robot is divided into three body segments that correspond to the thorax segments of the insect. Each of the body segments is connected to one pair of legs. To realize the bending of the body as observed in stick insects the body segments are connected with each other by body segment actuators that allow for bending in horizontal and vertical direction. However, at the time of writing, the body segment actuators have not yet been integrated into the controller framework and will therefore not be discussed.

The robot was intended as a research platform with a focus on bioinspired locomotion but also for walking-related experiments such as obstacle avoidance based on vision (see section 7.2). Beyond the sensor/actor domain, concepts have been developed by Schilling, Paskarbeit, Hoinville, et al. (2013) to allow the robot to find solutions for unforeseen problems during locomotion—hence the *C* in its acronym. Moreover, the robot is supposed to operate autonomously, which requires an onboard computer and batteries.

Therefore, in addition to the requirements regarding the actuation that are listed in section 2.3, further requirements were compiled for the design of the overall robot:

- The front segment should be easily replaceable to allow for integration of additional sensors.
- The middle segment should be big enough to house a PC/104 computer system in the central compartment.
- The hind segment is supposed to house the battery.
- The control and power components housed in the segments must be easily accessible.
- The workspaces of the legs should be as big as possible.
- The body segments must be slewable relative to each other in pitch and yaw up to angles of 20°.
- The mass of the body segments should be as low as possible.

Due to the requirement to integrate a PC/104 into the middle segment a scaling factor of approximately 1:20 between the insect and the robot was determined. The general appearance of the robot was designed by Achim Seemayer (supervised by Anke Bernotat) during his diploma thesis at the *Folkwang University of the Arts* (Essen, Germany). The central body segments were created by Martin Schulz during a student research project (*Großer Beleg*, supervised by Axel Spickenheuer) at the *Leibniz Institute of Polymer Research Dresden, Germany.* In close collaboration with these partners the final design was established under consideration of the aspired appearance, the production limitations, and the listed technical requirements. Details on the leg design and the definition of the body contour will be given in the following subsections.

	insect measures			robot measures
	front legs	middle legs	hind legs	for all legs
coxa [mm]	1.6 (±0.2)	1.5 (±0.2)	1.5 (±0.2)	30
femur [mm]	14.7 (±1.3)	11.4 (±0.9)	12.3 (±1.5)	280
tibia [mm]	13.7 (±1.4)	10.7 (±0.9)	11.8 (±1.7)	300

Table 2.1.: Mean leg segment lengths of *Carausius morosus* according to Cruse (1976b) and leg measures of HECTOR.

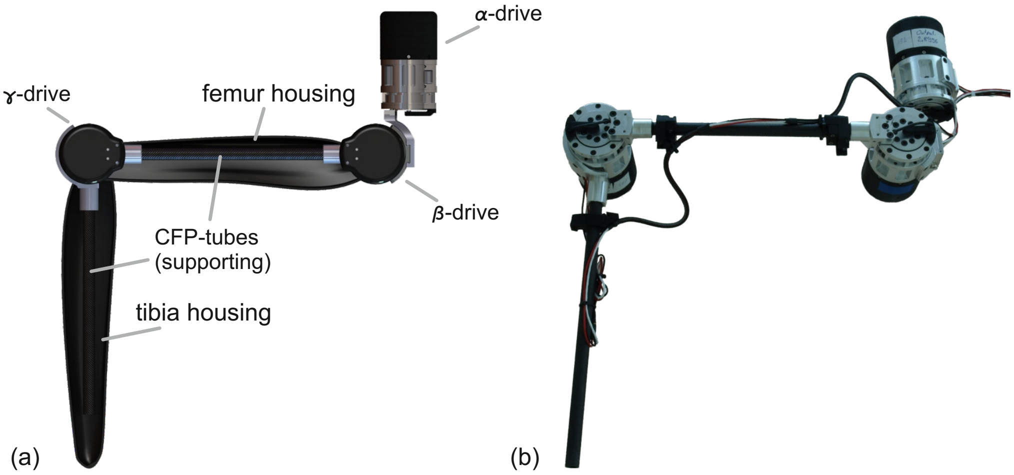


Figure 2.8.: Single leg of the robot. (a) rendered with leg segment housings attached, (b) image of an assembled leg with cabling between the drives.

2.4.1. Leg Design

With the stick insect legs as model, the robot legs have been designed to reflect the overall morphology. Characteristic of the biological model are the short coxa and the relatively long femur and tibia (see table 2.1). To realize the short coxa, the α-drive had to be mounted above the β-drive (see fig. 2.8 (a)). As described in section 2.3, the output of the drives is pivoted only on one side by a cross-roller bearing to reduce the length and the mass of the drive and to simplify the leg assembly. This allows a direct mechanical connection between the output flanges of the β- and γ-drives (see fig. 2.8). As a consequence of this setup, the femora of the robot legs are off-centered (not in the leg plane, see fig. A.5), whereas the tibiae, mounted at the input flange of the γ-joints, are centered again. Figure 2.8 (a) shows a rendering of the final leg design including the leg segment housings that were planned to be 3D-printed. In fig. 2.8 (b), an assembled leg without housings is depicted. Using this design, the femur and the tibia can be constructed from CFRP (Carbon-Fiber–Reinforced Polymer) tubes (11 mm outer diameter, 8 mm inner diameter; CG TEC GmbH, Spalt, Germany) and a minimal

	insect measures	robot measures
front legs [mm]	28.4 (± 2.0 mm)	578
middle legs [mm]	10.9 (± 0.7 mm)	218
hind legs [mm]	0.0	0

Table 2.2.: Positions of the leg onsets along the body axis relative to the hind legs for the biological model and the robot. The measures for the stick insect (*Carausius morosus*) were recorded by Cruse (1976b). (The variances of the leg positions given by Cruse (1976b) for the stick insect refer to measurements relative to the head segment. For the transfer to the hind leg reference they were assumed to compensate.)

set of required aluminum parts. Since the leg segment housings are not load-bearing, they are not functionally relevant for the leg although it provides a certain protection for the cables between the main body and the γ-drive and for optional sensor boards.

2.4.2. Body Design

Since the robot should reflect the morphology of *Carausius morosus* not only the legs were scaled from the biological model but also their relative positions were adopted. Table 2.2 lists the positions of the leg onsets along the body axis of insect and robot, both relative to the respective onsets of the hind legs. Diverging from the planned scaling factor of 20 between insect and robot, the positions of the front legs were shifted by 10 mm to the front to increase the space for the PC/104 in the middle segment. According to Cruse (1976b), the variance for the distance between the front and the middle legs of the insect is in the range of ±1.3 mm. Therefore, the shifted position of the front legs can still be considered biologically plausible.

For the lateral positioning of the legs the scaling had to be modified as well. The distances between contralateral leg pairs of *Carausius morosus* were measured to be in the range of ~1.5-5 mm with the smallest distance between the front legs. Due to the size of the actuators, an accurate scaling (~30-100 mm) would require a considerable reduction of the angular workspace of the legs since the housings of β-drives of contralateral legs might otherwise collide with each other. Therefore, in a first step, the desired movement angles of the leg joints were approximated based on the leg workspaces of stick insects (Cruse 1976b). Whereas the insect's middle leg workspaces are nearly centered, the front and hind leg workspaces are shifted relative to the leg onsets to the front and back, respectively. The desired angular limits for the leg joints of the robot are listed in table 2.3. To realize these angular movement ranges and still prevent collisions of the β-drive housings of contralateral legs, the distance between contralateral leg onsets had to be increased to 140 mm.

Before the design of the body segments could be started, the maximum available space had to be defined which is not intersected by any part of the legs for all combinations of joint angles within the desired joint angle workspaces. For this purpose, the movements

	front		middle		hind	
	left leg	right leg	left leg	right leg	left leg	right leg
α-range [rad]	[-0.6, 1.4]	[-1.4, 0.6]	[-1.0, 1.0]	[-1.0, 1.0]	[-1.6, 0.5]	[-0.5, 1.6]
β-range [rad]	[-1.6, 0.8]	[-0.8, 1.6]	[-1.6, 0.8]	[-0.8, 1.6]	[-0.8, 1.6]	[-1.6, 0.8]
γ-range [rad]	[-1.2,1.2]	[-1.2,1.2]	[-1.2,1.2]	[-1.2,1.2]	[-1.2,1.2]	[-1.2,1.2]

Table 2.3.: Desired joint angle ranges for the six legs of Hector. The values were approximated based on the leg workspaces of *Carausius morosus*. For the neutral positions (joint angles at 0°), see fig. A.2.

of the leg segments were simulated for the desired movement ranges of the joints.[5] Based on this simulation, the maximum available space was computed that is not intruded by any part of the legs. In the actual design process of the body segments, all parts of the body housing must lie entirely within this *maximum body space*.

Due to the asymmetry in the leg setup (as mentioned, the femora are off-centered), the workspace that needs to be clear for the movement of the leg segments changes depending on the mounting direction of the β- and γ-drives. To find a suitable configuration, the *maximum body space* was computed for different leg setups. Exemplarily, the results for three of these configurations are depicted in fig. 2.9. For each combination of x- and z-positions along the symmetry plane of the robot, the closest y-coordinate is plotted that can be reached by any leg segment (for joint output angles within the desired working ranges). In fig. 2.9 (a), the *maximum body space* is depicted for a leg setup, in which all β-joint output flanges are directed backwards (the γ-drives are always oriented in the same way as the β-drives). Therefore, also the femora are positioned rearwards relative to the leg plane (indicated in the pictogram on the right side of the figure). The graphic shows that due to the off-centered femur, the workspace of the hind legs extends inwards up to the symmetry plane of the robot (indicated by dashed lines in the robot pictograms). To prevent collisions between the hind leg segments and the body, this area would need to be spared in the body design. For the middle and front legs, directly behind the leg onsets, nearly vertical indentations must be left open for the femur to slew into. In fig. 2.9 (b), the *maximum body space* is depicted for a leg setup with all β-joint outputs and femora directed to the front. In comparison to the setup shown in (a), the vertical indention must now be left in front of each leg onset since the asymmetric orientation of the femur is reversed. As previously for the hind legs, in this configuration, the workspace of the front legs extends to the symmetry plane. Therefore, mounting of sensors at the front of the robot would not be possible as these might collide with the segments of the front legs.

Based on these results, a mixture of the two previous designs was chosen for the robot setup. In this configuration, the β-joint outputs of the front and middle legs are directed backwards, whereas the β-joint outputs of the hind legs are directed forwards.

[5] To reduce the complexity of the simulation, femur and tibia were modeled as cylinders with a diameter of 40 mm. The diameter of the cylinders was chosen such that they envelop the complex outline of the actual leg segment housings.

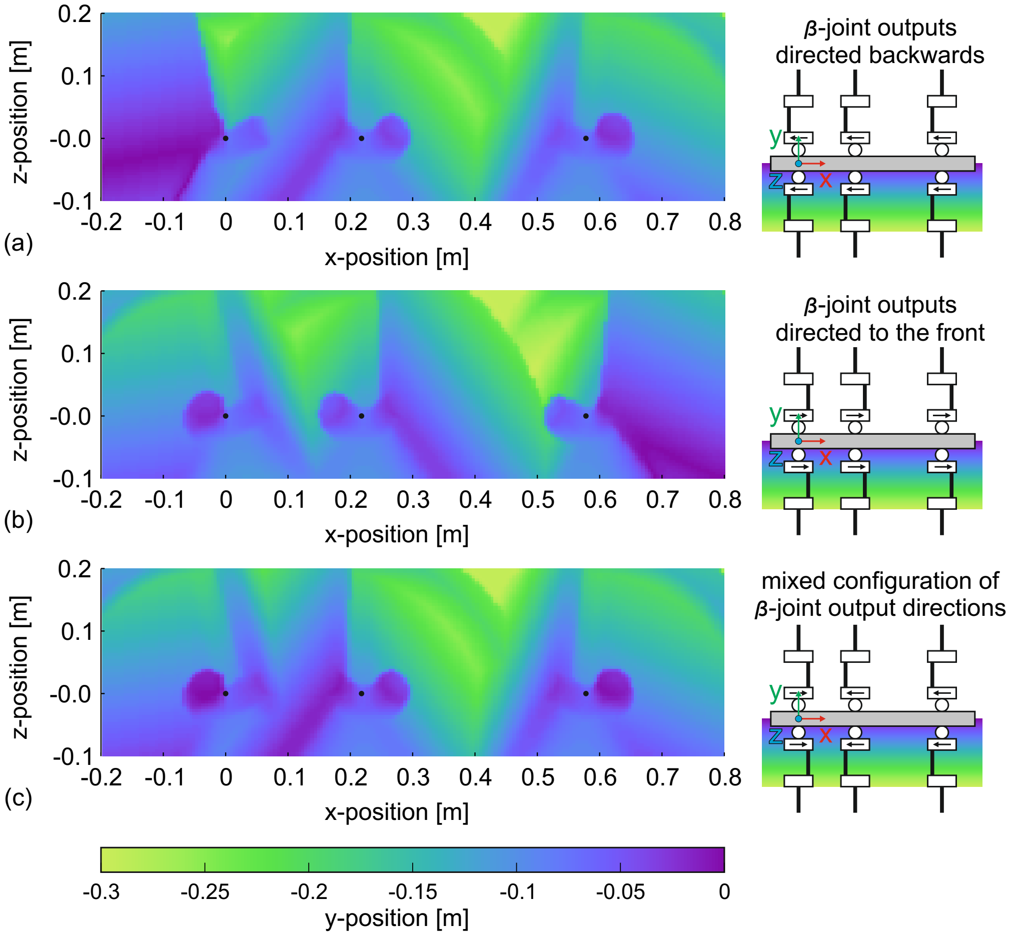


Figure 2.9.: Depictions of the *maximum body spaces* for three different mounting configurations of the legs (schematically illustrated in the respective pictogram on the right). The subfigures on the left indicate the *maximum body spaces*, therefore the maximum outlines for the design of the body segments such that collisions between the body and the leg segments can be precluded. The data is shown for the right side of the robot. However, since the leg workspaces are symmetric for both sides of the robot, also the outlines of the *maximum body space* will be symmetrical. Since the femora are off-centered relative to the leg planes (see robot pictograms on the right) the *maximum body space* depends on the mounting directions of the β- and γ-drives, and the corresponding position of the femora. (a) shows the configuration for all femora oriented towards the back of the robot. (b) shows the configuration with all femora oriented towards the front of the robot. (c) depicts a mixed configuration, in which the femora of the front and middle legs are oriented backwards whereas the femora of the hind legs are oriented to the front. The points around which the β-joints rotate (corresponding to the leg onsets in the insect) are marked by black dots. In y-direction, these are located at -0.07 m. Figure 2.11 shows the final body design in combination with the last configuration.

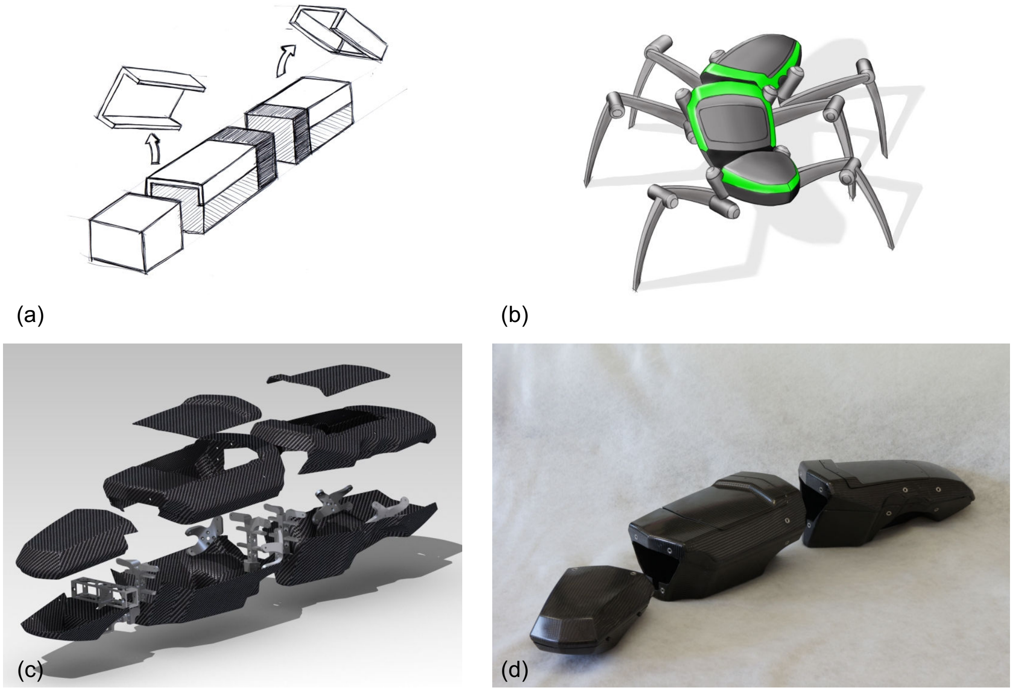


Figure 2.10.: Different steps in the design of the robot. (a) shows an early drawing that illustrates the segmentation of the main body and the concept of exchangeable elements: all white parts can be replaced, e.g., by 3D-printed plastic parts (courtesy of Achim Seemayer). (b) shows the first draft for the leg orientation shown in fig. 2.9 (c) (courtesy of Achim Seemayer). (c) Rendering of the final housing design. All segments are constructed of an upper and a lower housing (and a lid, if applicable) due to requirements of the manufacturing process (courtesy of Martin Schulz). (d) Image of the three body segments without legs and intersegmental drives (courtesy of Martin Schulz).

The resulting outline of the *maximum body space* is shown in fig. 2.9 (c). The thorax of stick insects is divided into three segments with the leg onsets close to the back of each segment. For the robot, a similar setup was planned. Since the body segments are intended to be slewable relative to each other, the outline of the body must be constricted at the transitions to prevent collisions between the body segments. This enables the front and middle legs to slew into the space between the body segments. For the hind legs, dedicated indentations had to be provided in the hind segment in front of the leg onsets.

With these restrictions for the maximum outline of the body segments, the design process was initiated. Figure 2.10 shows four relevant stages of this process. In (a), the conceptual drawing is depicted that shows the envisioned segmentation of the main body. As the computer and the batteries are to be mounted within the body housing, lids must be provided for easy access. Figure 2.10 (b) shows a preliminary design sketch by Achim Seemayer is shown for a left-side turn of the robot. This sketch visualizes the body segment actuators and the off-centered assembly of the femur joints.

Since one of the goals in the fabrication of the body housings from CFRP was the reduction of mass as compared to a metal body, the body segments are supposed to be self-supporting without the need of additional, internal support structures (e.g., an endoskeleton). A rendering of the body housing design that complies with this concept is shown in fig. 2.10 (c). The body segments are connected to each other by the body segment actuators. Since the front segment is supposed to be completely exchangeable, this segment was fabricated without the requirement to be self-supporting. Thus, the legs are connected directly to the frame of the body segment actuator. To fix the legs to the middle and hind body segments, small metal inlays were embedded into the CFRP hull to allow reversible connections between the legs and the housings using screws. These metal inlays were also used to define connection points for the segment actuators. By limiting the size of the middle and hind segment lids, and the housing of the front segment to a volume that is printable by an available 3D-printer (203 mm x 203 mm x 305 mm build size; Dimension SST 768, Stratasys, Eden Prairie, USA), an easy adaptability to various optional sensor systems was ensured. Thus, if a 360° camera is supposed to be mounted on the middle segment, the design of the corresponding lid can be adapted and printed. In the front segment, different sensors are imaginable—ranging from cameras for visual feedback to tactile antennae. Due to the modular setup, appropriate front segment housings can be designed for the different sensor systems.

The final housings are shown in fig. 2.10 (d). The bright metallic spots on the housing are the mentioned metal inlays, to which the legs and the body segment actuators are to be fixed.

In fig. 2.11, the final design of the body segments is shown again together with the limiting surface of the legs' workspaces that was already depicted in fig. 2.9 (c). Due to the asymmetry of the leg setup, the spaces between the body segments can be utilized to realize the angular working ranges of the leg joints.

The wiring within the segments is considerably reduced due to distributed control electronics of the actuators, as depicted in fig. 2.12. Therefore, it is sufficient to connect the joint drives with the power supply and the *BioFlex* bus to exchange data with the central controller. To connect the embedded PC/104-system to the RS-485–based *BioFlex* bus, custom transmitter boards were developed by Schäffersmann (2011) that allow to link via USB (Universal Serial Bus). These boards are also distributed onto the three body segments.

All electronic components in the robot are powered with the same voltage that was defined to be in the range of 15 to 40 V. It is either supplied by a LiPo (Lithium-ion Polymer) battery embedded in the hind segment or an external power supply. Therefore, it is unnecessary to route different voltages throughout the robot. However, this necessarily requires each component to convert the common supply voltage to the required voltages locally.

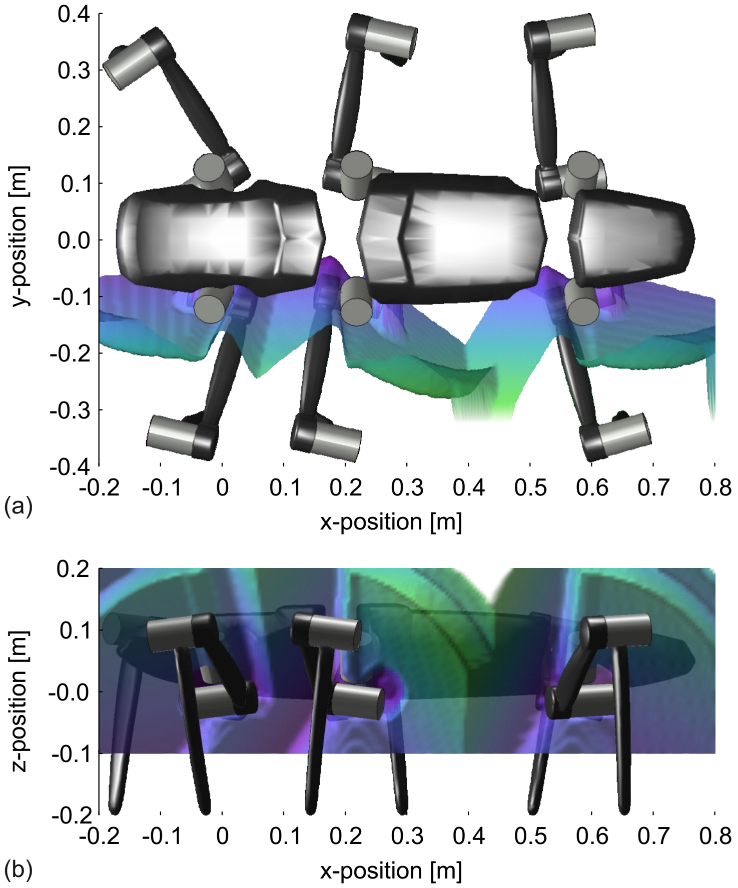


Figure 2.11.: Depiction of the final body design in (a) top and (b) side view in combination with the restricting surface already shown in fig. 2.9 (c). The colors of the restricting surface are not to scale with the colormap shown in fig. 2.9 due to the transparency and light effects.

A scaled image of *Carausius morosus* in direct comparison with a rendering of the robot is shown in fig. 2.13 (a). As can be seen, the dimensions of the thorax segments have been transferred together with the leg morphology to the technical system. The COMs of the biological and the technical system have been indicated in the illustration. Whereas the COM of the insect is located roughly between the hind leg onsets, the COM of the robot is shifted forwards, slightly in front of the middle leg onsets since the abdomen of the insect was not replicated in the robot setup. In fig. 2.13 (a), an image of the assembled robot is shown. The robot has a mass of 13 kg, which is mostly due to the combined mass of the leg joint actuators (18 × 0.39 kg ≈ 7 kg). The length of the body is 95 cm and the width during walking is roughly 60 cm due to the sprawled leg posture.

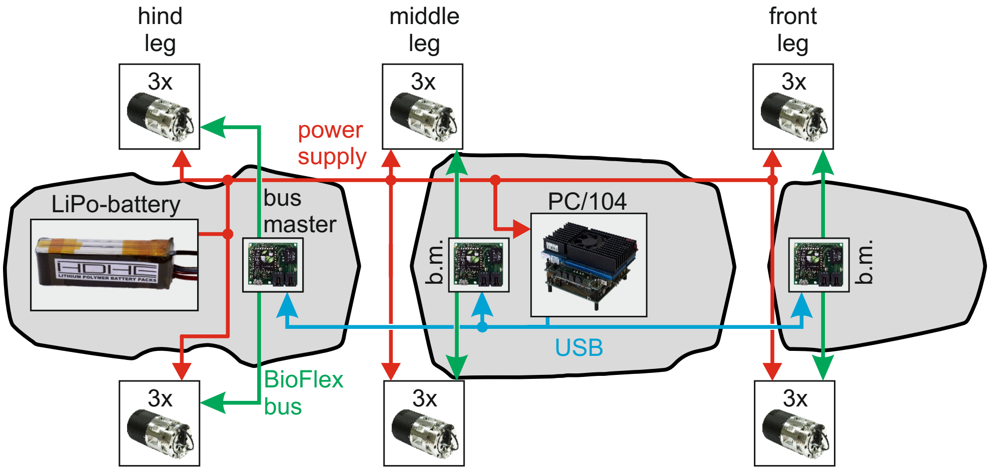


Figure 2.12.: Schematic of the wiring between the components of the robot. Since each of the actuators is capable of receiving/transmitting controller commands via the *BioFlex* protocol, they need to be connected only to the power supply (red lines) and the *BioFlex* bus (green lines). The bus masters (b.m.), responsible for transmitting messages to/from the clients, are connected to the main controller, a PC/104 computer, via USB (blue lines).

2.5. Robot Simulator

To evaluate experimental robot controllers in different surroundings without the risk of damaging the hardware, a simulation environment was created based on ODE (Open Dynamics Engine; Smith 2001). For the walking controller to interact with the simulation (both programmed as separate processes in Python and C++), a TCP/IP (Transmission Control Protocol/Internet Protocol) connection can be set up between the two processes. Via this interface, *BioFlex* bus compatible messages can be exchanged (Schneider et al. 2012).

The robot geometry as well as objects defining the environment can be dynamically created based on XML-formatted object definitions (see fig. 2.14 for the virtual representation of the robot walking in flat terrain). To simplify collision detection, the femur and tibia segments are usually represented only by *capsules* (cylinders with hemispherical ends) instead of trimeshes that can be used to represent the complex geometry of the leg segment housings. For the physics simulation, however, the inertias of the fully equipped leg segments were used.

The rotatory leg joint drives have been implemented geometrically (for the collision detection) as cylinders and functionally as hinge joints that are driven directly with a torque. As described in section 2.3, the real drives are equipped with compliant elastomer couplings. Since a detailed simulation of these couplings would considerably slow down the robot simulation, eq. (2.1) is used to approximate the spring characteristic

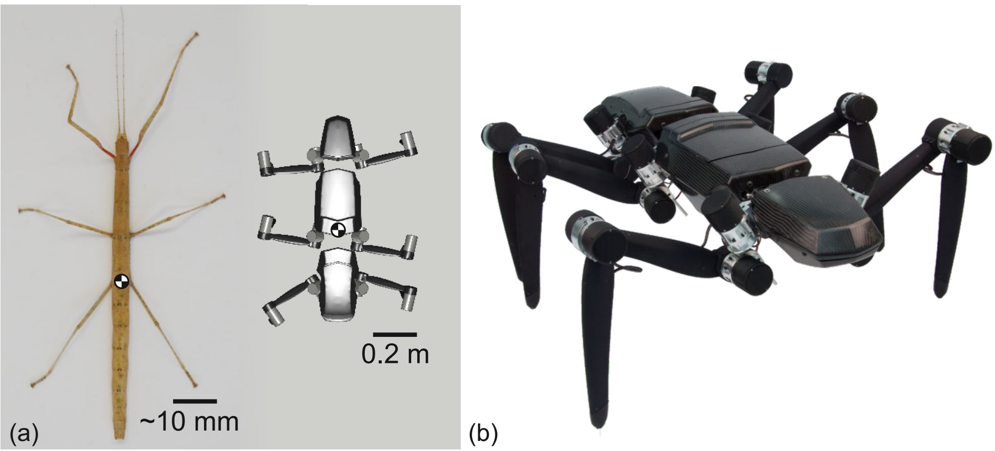


Figure 2.13.: (a) Images of *Carausius morosus*, the biological model, and rendering of HECTOR in top view. The scaling between model and robot is 1:20. The COMs of both systems are depicted by quartered circles. (b) Image of HECTOR.

of the elastomer coupling. In addition, a parallel damper with a damping constant of 0.4 N m s/rad was added in the simulated drives to account for the damping characteristics of the elastomer coupling.

To avoid numerical instabilities, the simulation frequency was fixed to 1 kHz with respect to the *virtual time* within the simulation. Dependent on the complexity of the simulated environment, the simulation frequency with respect to *real time* varies. With the robot walking on even ground, real-time execution can be obtained. In uneven terrain, due to the need to detect collisions between the legs and the trimeshes the ground is built of, the simulation frequency with respect to real time may drastically decrease. Therefore, a synchronization between the walking controller and the simulation is required: At the end of every walking controller iteration, a special command is sent to the simulation that triggers the execution of a given number of simulation iterations. After the execution of these simulation cycles, the simulator sends an answer to signal completion. Thus, the walking controller is synchronized to the *virtual time* of the robot simulator.

2.6. Communication Framework

As mentioned in section 2.5, the controller communicates with the simulation via TCP/IP whereas the joint drives presented in section 2.3 use a differential serial interface for communication. In the real robot bus masters are used to transmit messages received via USB from the computer to the serial interface and vice versa (Schneider et al. 2012). This is depicted in fig. 2.15. To unify the interface between the walking controller and the simulation or real robot hardware, an additional layer, the *BioFlex*

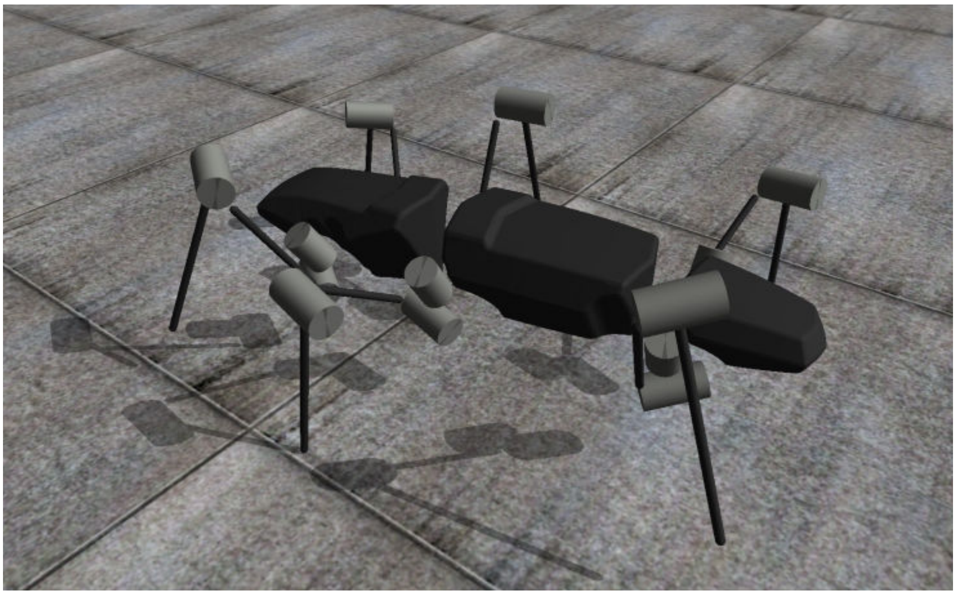


Figure 2.14.: Visualization of the simulated robot walking on flat terrain. The housings of the body are represented by trimeshes, the leg actuators by cylinders, and the connecting leg segments by capsules. By simplifying the representation of the detailed leg housing shown in section 2.4.1 the computational load can be reduced since collision detection for capsules is computationally less complex than for trimeshes. For the same reason shadows are only illustrated for those parts of the robot that are represented by simple geometric primitives. Since the body housings are represented by trimeshes the corresponding shadows are not visualized.

server, was introduced that manages the routing of messages between the TCP/IP interface and the USB ports. In the protocol defined for the *BioFlex* bus, the first two bytes of every message are reserved for the destination and the source identifier number.[6] With distinct IDs for the TCP/IP-connected controllers and the USB-connected bus masters and clients, the server is able to route messages in both directions. Keeping track of the requests sent to the clients and the corresponding replies, the server also detects packet loss and automatically re-requests a reply.

As the simulator does not always run in real-time, the walking controller is synchronized to the *virtual time* of the simulation. Thus, the controller does not use any timing on its own but relies completely on the external timing provided by the simulation. When connected to the real robot, this task must be performed by the server to keep

[6] For a detailed description of the protocol, see (Schäffersmann 2011).

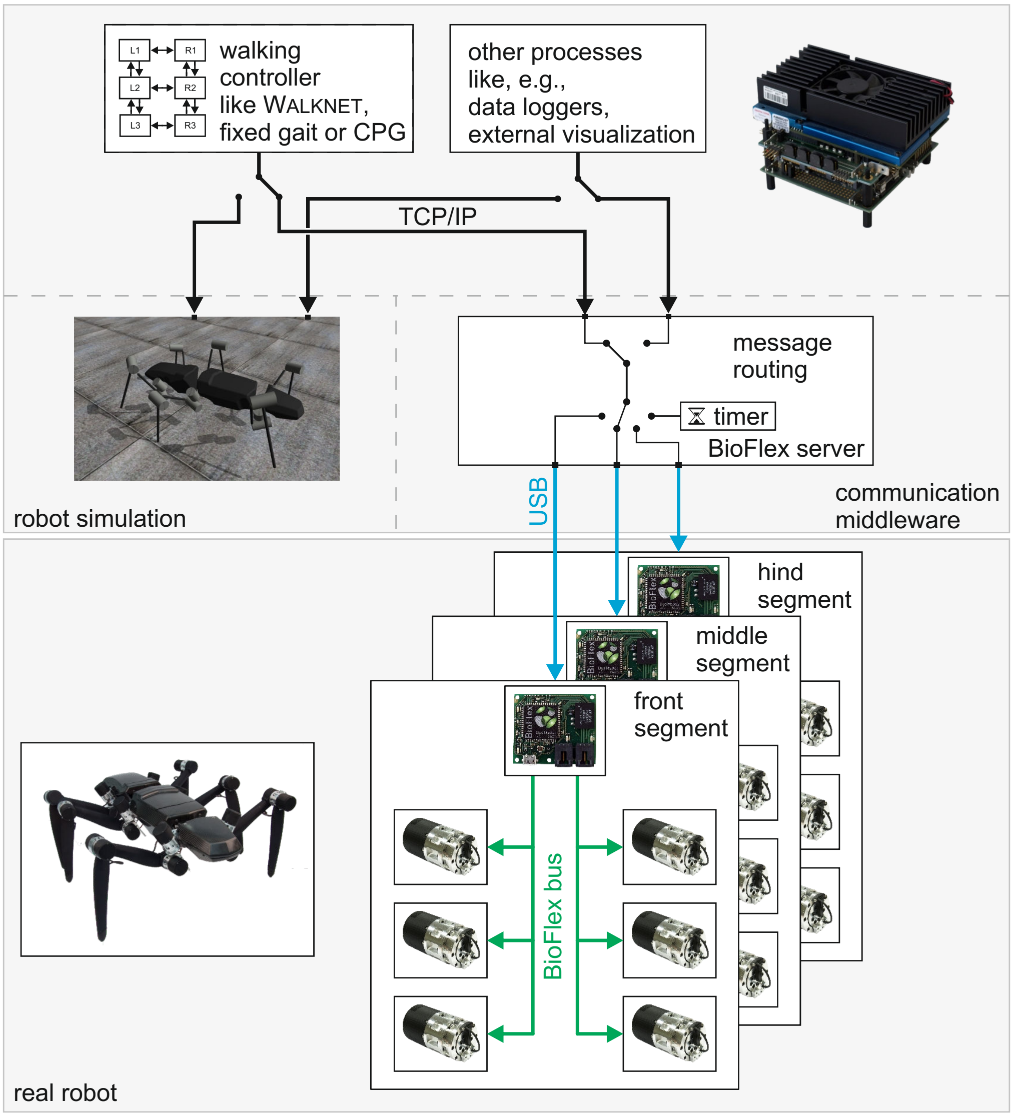


Figure 2.15.: Depiction of the communication framework for controlling the real robot or the virtual robot in the simulation. The robot controllers connect via TCP/IP either to the simulation or—if the real robot is to be used—the *BioFlex* server. In the latter case, the messages sent from the controllers are forwarded either to the timer module or to the respective *BioFlex* master via USB. In the last step, the message is sent to the client via the *BioFlex* bus. If an answer is required the client can send it to the controller using the reverse process.

the interface identical. Since real-time execution is inevitable for the operation of a real robot, the server uses the computer's hardware clock for the timing.

Using this communication schema, new walking controllers can be tested first with the simulation and then, without any modifications, on the real robot.

Summary

To test the applicability of bioinspired leg coordination for technical systems, a robotic platform is required that resembles the biological model as close as possible. However, due to the limitations of current technical fabrication, not all features can be transferred. Therefore, the features considered to be most significant must be selected. In the design of the six-legged robot Hector especially the morphology of the model insect *Carausius morosus* was adopted since this was considered to be most relevant for replication of insect-like walking behavior. Although an exact scaling of the stick insect measures was not accomplished—partially due to technical/mechanical requirements, partially to simplify the setup—the robot is considered to be suitable to test bioinspired control approaches.

Beside the insect-like morphology, also the compliance of the biological structures was transferred to the robot. Whereas the compliance in the insect joints is mostly due to the elasticity of the muscle fibers, in the robot the compliance was implemented by introduction of custom elastomer couplings in the joint drives. The resulting inherent compliance can be utilized for passive terrain adaption. Moreover, based on the torsion of the coupling the load in the drives can be estimated, which can be used to detect ground contact or collisions during swing phase. Since a permanent, durable connection between the elastomer of the coupling and the metal support structures could not be realized, currently an unbonded elastomer inlay is used in the joint drives. Although the inlay was manufactured with oversize, the development of backlash over a longer period of use cannot be precluded. During operation of the robot, however, no backlash was detectable.

Among the biological features that were not replicated in the robot are the adhesive pads and claws at the insects' tarsi that are used to hold on to the substrate. These features were explicitly omitted since they cannot be used to cling to all substrates (e.g, loose sand or gravel). Aiming at substrate-independent functionality of the robot, maintenance of static stability is a prime requirement for walking locomotion to avoid tilting. Technical equivalents of the adhesive pads and claws have therefore been considered redundant for fundamental robot operation. However, to increase the ability of the robot to climb steep slopes, a subsequent expansion of the legs by adhesive or gripping mechanisms might be required.

To simplify the development and evaluation of walking controllers, a simulation framework was developed that allows for preliminary tests on a virtual robot. For example, in chapter 4, the robot simulation is used to evaluate the walking controller Walknet. Also for the development of the novel, adaptable controller (see chapter 5) that considers the differences between model and robot, in a first step, the simulation framework was employed (see chapter 6). In chapter 7, finally, this newly developed controller is used for the control of the actual robot in different walking scenarios.

3. Evaluation of Static Stability of Walking Stick Insects

Abstract: *Insects, in particular stick insects, have often been assumed to maintain stability at all times since three legs remain in contact with the ground during tripod gait, the fastest regular gait that can be observed in many insects. However, being equipped with adhesive pads and claws at the tarsi, stick insects do not have to rely on the maintenance of static stability to prevent tilting. Different experiments with contradictory results have previously been published on this matter. Using kinematic data (by Theunissen et al. 2014), the static stability of stick insects is evaluated for multiple experimental data sets. It can be shown that the projection of the COM frequently leaves the support polygon. Therefore, stick insects do not permanently maintain static stability during walking.*

Some of the results and outcomes presented in this chapter have already been published in the following paper together with additional results from the specified co-authors: Paskarbeit, Otto, Schilling, and Schneider 2016

Many insects do not have to maintain static stability at all times since they can rely on their ability to cling to the substrate (Bullock et al. 2008). As a consequence, the loss of static stability does not necessarily result in tilting. Examples for insects that apparently do not sustain static stability while walking—at least not exclusively by using their legs—are *Tropidopola cylindrica* (La Greca 1943) and *Rhipipteryx chopardi* (Wille 1924). Typically only using front and middle legs during walking, the first often lifts contralateral, the latter diagonal leg pairs simultaneously. As a consequence, the abdomen are dragged over the ground. Nevertheless, both species are able to climb up branches.

In contrast, grass-cutting ants (*Atta vollenweideri*) try to keep static stability since they usually walk on loose substrates and therefore cannot cling to the ground. This is especially difficult when transporting objects larger than the insect itself (Moll et al. 2010; Zollikofer 1994b).

As mentioned in section 2.1, the stick insect *Carausius morosus* that was used as a model for the robot Hector lives mostly in bushes, climbing on twigs. It has been observed climbing along branches while hanging upside-down (Cruse, Ehmanns, et al. 2009) and the mechanisms that enable it to cling to various substrates have been studied (Bullock et al. 2008; Bußhardt et al. 2011). However, although stick insects possess mechanisms to cling to the substrate, they are often assumed to maintain static stability when walking on flat ground (Cruse 1985; Dean and Wendler 1984; Hughes 1952; McGhee and Sun 1974). Thus, they have frequently been used as model for bioinspired robots that rely on the maintenance of static stability (Amendt and Frik 1994; Berns et al. 1994; Cruse, Dean, et al. 1991; Espenschied, Quinn, Beer, et al. 1996; Frik et al. 1999; Gaßmann et al. 2001; Manoonpong et al. 2008; Pfeiffer 2007; Rönnau, Heppner, Nowicki, et al. 2014; Rönnau, Kerscher, et al. 2009, see also table 1.1). Regarding the maintenance of static stability of stick insects during walking only few experiments with contradictory outcomes have been conducted: Jander (1985) evaluated the static stability of stick insects that walked on a *locomotion compensator* (Kramer 1976), an actuated sphere with a diameter of 50 cm. According to the obtained results, during tripod gait the stance-swing transition of a leg triplet is triggered if the static stability can be maintained by the other leg triplet. Another experiment during which the insects walked from a glass plate onto a loose sand layer was conducted by Kindermann (2003). As soon as the insect stood with its front and middle legs on the sand and a hind leg lifted off for swing phase, the insect tilted backwards since the projection of the COM moved outside the support polygon of stancing legs and the insect could not hold on to the loose surface. Therefore, according to Jander (1985), the tested stick insects permanently maintained static stability whereas those tested by Kindermann (2003) did not. Since the experimental setups differ between these two trials, no definite conclusion can be drawn based on the available results.

Recently, stick insects have been equipped with markers by Theunissen, Hertrich, et al. (2014) to record their movements with a motion capture system. This data can be used to evaluate the static stability of stick insects during walking in flat terrain. Beside other kinematic data the database (Theunissen et al. 2014) contains the absolute positions of the tips of the tibiae and the respective onsets of the legs for nine walks of stick insects in flat terrain (three of the trials were performed with *Carausius morosus*). Moreover, the database contains information on the temporal start and end of stance and swing phases. Based on the tibia tip positions and the leg states (stance/swing), for each recorded frame the support polygon can be computed. As the positions of the tarsi were not recorded, the exact position of contact between each leg and the ground is unknown. However, the database contains at least information regarding the length of the tarsi for *Carausius morosus* (4.44–5.07 mm) that can be considered in the evaluation. Therefore, the actual point of contact between the leg and the ground could be shifted outwards by ∼5 mm. Assuming that the COM is roughly located between the hind coxae (cmp. Cruse 1976a), it can be assessed whether its projection along the vector of gravity lies within or outside the support polygon. In the former case, the insect is statically stable, in the latter it is unstable, which, as mentioned before, does not necessarily result in tilting since the insect might hold on to the substrate. To quantify the stability/instability, instead of a binary classification the distance between the supposed COM and the closest border of the polygon is computed (corresponding to the stability margin proposed by McGhee and Frank (1968)).

For one trial (“Animal11_100723_00_01”) the resulting data is shown in fig. 3.1. (a) shows the distance of the projected COM to the support polygon, (b) shows the gait pattern over time. As can be seen in (a), the COM stays most of time within the support polygon (positive distances). However, every time one of the hind legs lifts off, the

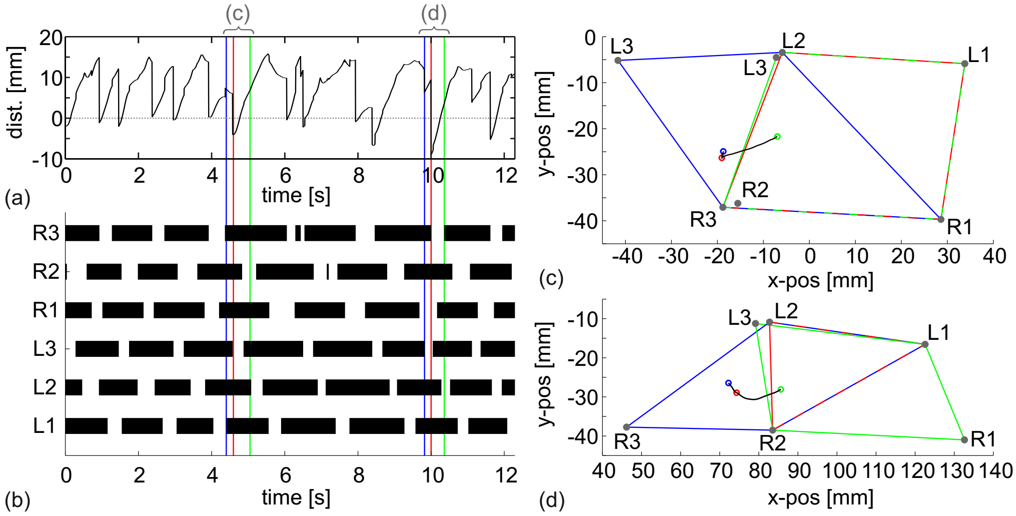


Figure 3.1.: Experimental data from walking stick insect *Carausius morosus* (based on data recorded by Theunissen, Hertrich, et al. (2014); trial “Animal12_110415_00_22”). (a) and (b) show the distance of the COM to the support polygon and the corresponding gait pattern over time. A positive distance in (a) corresponds to a statically stable posture, a negative distance denotes an unstable posture. In (c) and (d), exemplary foot positions are plotted for three points in time that correspond to the colored, vertical lines in (a) and (b). The connecting lines between the foot points represent the support polygons for the respective postures. The positions of the COM that correspond to the times for which the support polygons are shown, are marked by colored circles. The movement of the COM is plotted as a black line.

distance between the COM and the support polygon decreases abruptly. Often, this leads to a negative distance, which indicates instability. For two of these potentially instable situations the positions of the tibia tips of the stancing legs and the assumed positions of the COM are plotted in fig. 3.1 (c) and (d). The colored convex polygons represent the support polygons at the points in time marked by correspondingly colored vertical lines in (a) and (b) with the first line triplet belonging to (c) and the second belonging to (d). For both situations, during the first depicted frame (blue lines), the projection of the COM falls within the support polygon. In the second frame (red lines), the supporting hind leg is lifted, leaving the COM outside the altered support polygon. Due to the movement of the insect the COM is shifted forwards, reestablishing a stable situation (green lines). During the instable situation shown in (d), the distance between the COM and the support polygon reached down to -9 mm. Even if the measurement error of the motion capture system (0.1 mm according to Theunissen, Hertrich, et al. (2014)), the uncertainty regarding the exact contact between the tarsi and the ground (5.07 mm due to maximum length of tarsi), and the inaccuracy of the COM estimation (for *Carausius*

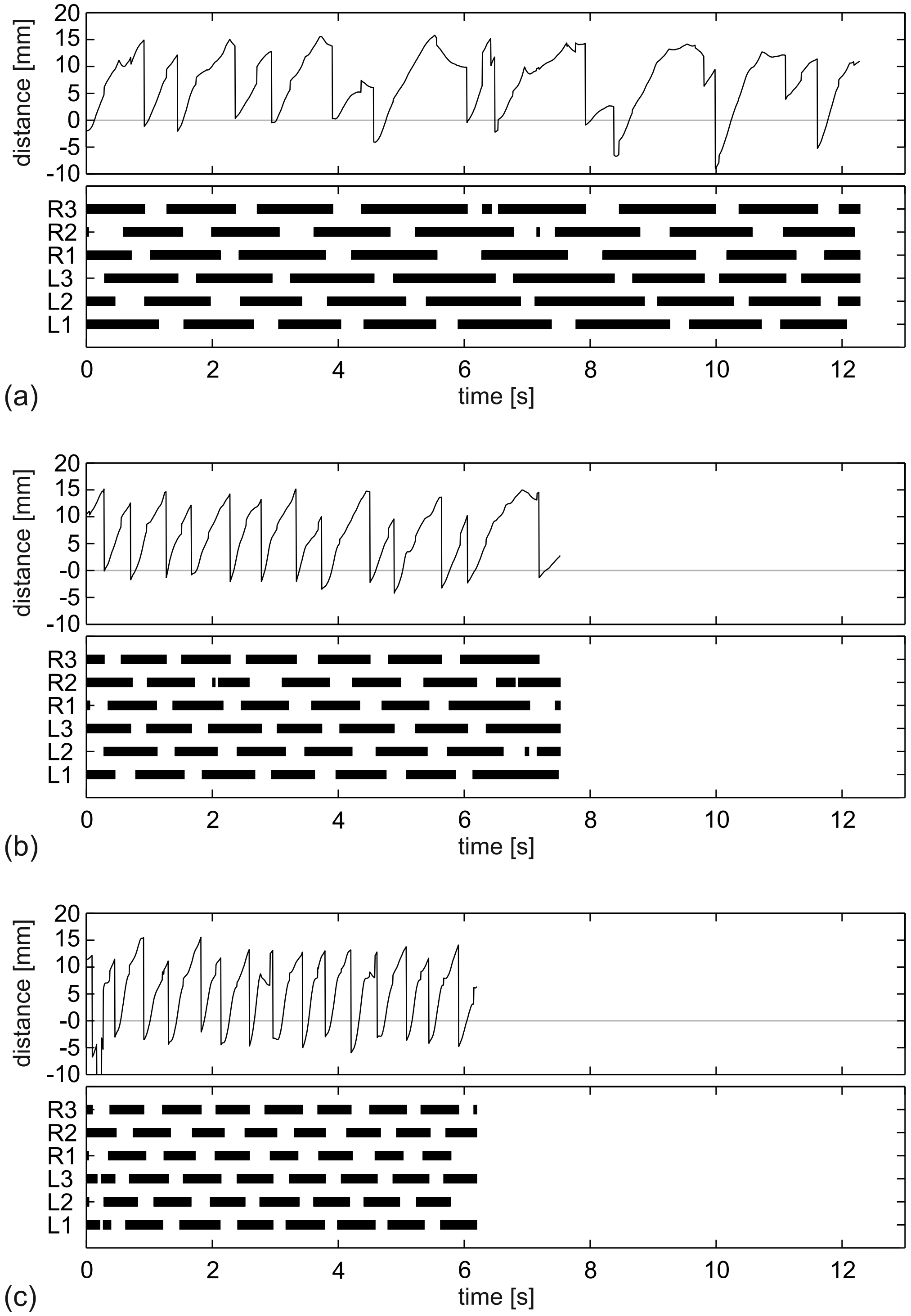


Figure 3.2.: Experimental data of walking stick insect *Carausius morosus* (based on data recorded by Theunissen, Hertrich, et al. (2014); trials "Animal12_110415_00_22", "[. . .]_23" and "[. . .]_32", shown in (a)–(c)). For each trial the distance of the COM to the support polygon and the corresponding gait pattern over time is shown (black bars indicate ground contact). A positive distance corresponds to a statically stable posture, a negative distance denotes an unstable posture. (a) was already shown in fig. 3.1.

morosus it is even assumed slightly behind the hind leg onsets) are considered, the maintenance of static stability at all times during the experiment is unlikely. However, due to the stick insect's ability to cling to the substrate, it does not tilt. As the video of the experiment shows, the abdomen of the stick insect is lifted off the ground at all times. Thus, it was not used to support the body. Therefore, to prohibit tilting, the insect must have used the adhesive pads or claws at its tarsi to hold to the substrate.

The database contains three trials for walks of *Carausius morosus* on flat terrain. For these trials, the gait patterns and the distance of the COM to the support polygon are shown in fig. 3.2. Additionally, the database includes three trials of *Aretaon asperrimus* and *Medauroidea extradentata* on flat terrain. The corresponding results are shown in figs. 3.3 and 3.4. According to Theunissen et al. (2015) the COMs of all three species are located "close to the thorax–coxa joints of the hind legs". In *Carausius morosus* it is "slightly posterior", in *Aretaon asperrimus* "slightly anterior" and in *Medauroidea extradentata* "almost between the hind leg coxae" (Theunissen et al. 2015). As no quantitative measures are given for the position, for the stability analysis it is assumed to be located exactly between the hind leg onsets. An incorporation of the exact position of the COM will probably increase the stability measure for *Aretaon asperrimus* and reduce it for *Carausius morosus*. Nevertheless, the analysis shows that the COM of the insects is often critically close to the support polygon when one of the hind legs starts its swing phase.

Summary

Stick insects have often been used as models for robotic systems since they have been expected to maintain static stability during walking. This view was supported amongst others by Jander (1985). However, experiments by Kindermann (2003) have indicated that stick insects do not maintain static stability at all times.

On Hector, the application of a stick insect - inspired controller for leg coordination was planned. Since the robot, as opposed to stick insects, does not possess any means to cling to the substrate, the maintenance of static stability is required to prevent tilting. To test whether stick insects would maintain static stability without these clinging mechanisms and therefore whether an adaptation of the biological walking controller would be suitable for the control of Hector, kinematic data of walking stick insects (published by Theunissen et al. 2014) was evaluated. The analysis of this data shows that the projection of the COM of stick insects is frequently located outside the support polygon (indicating instability) if one of the hind legs lifted off for swing phase. Some aspects of the insect locomotion (e.g., trajectories of tarsi) have not been recorded by Theunissen et al. (2014) since these were irrelevant for the originally intended research. Therefore, the exact locations of contact between the legs and the ground cannot be determined from the data. Also, the positions of the COMs of the insects can only be approximated. As a consequence, the instability of the insects during walking cannot be quantified as ratio of stable vs. instable periods. Nevertheless, even if all imprecisions are assumed in favor of stability, a permanent maintenance of static stability of walking stick insects

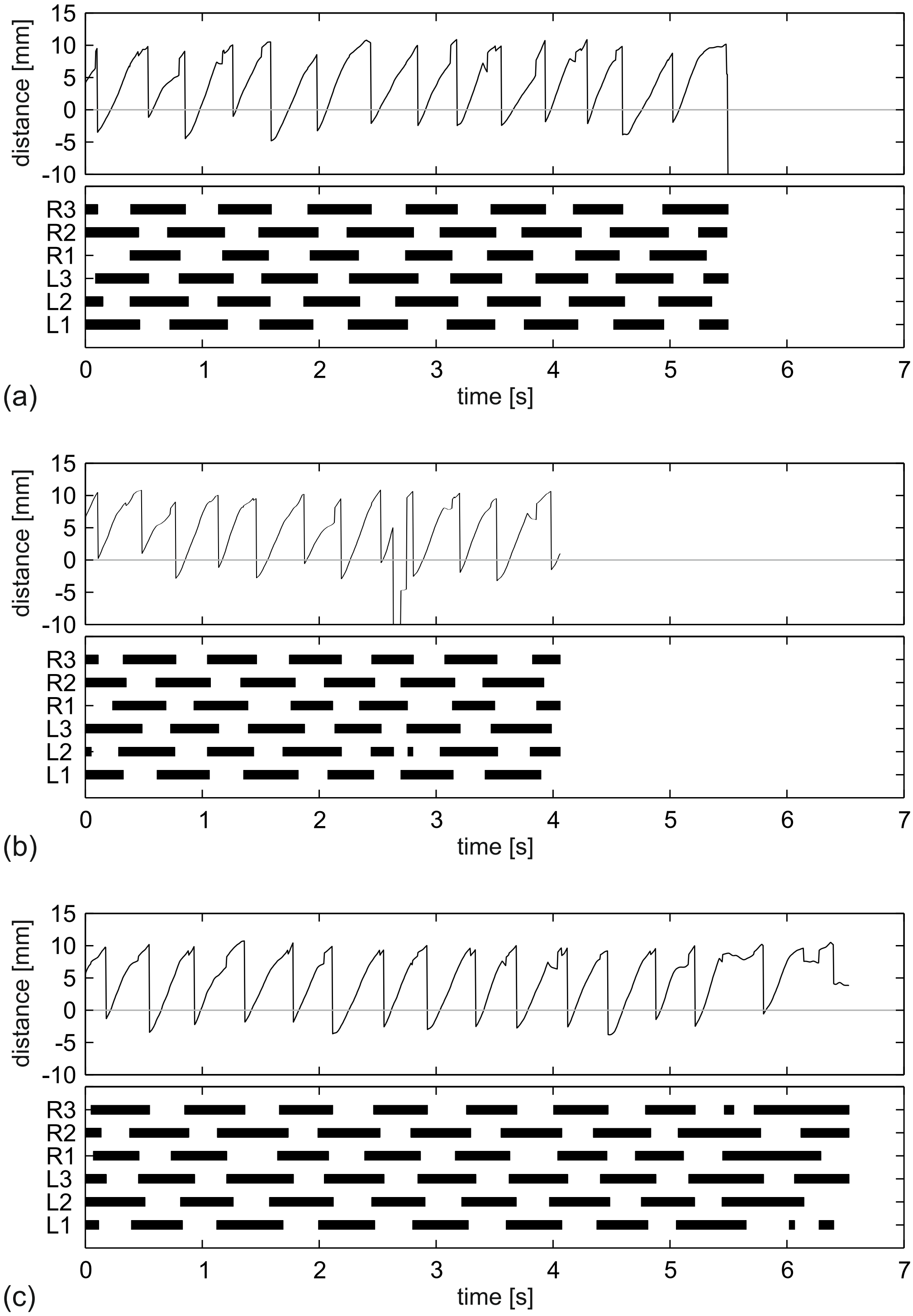


Figure 3.3.: Experimental data of walking stick insect *Aretaon asperrimus* (based on data recorded by Theunissen, Hertrich, et al. (2014); trials “Animal11_100723_00_01”, “[…]_04” and “[…]_05”, shown in (a)–(c)). For each trial the distance of the COM to the support polygon and the corresponding gait pattern over time is shown. A positive distance corresponds to a statically stable posture, a negative distance denotes an unstable posture.

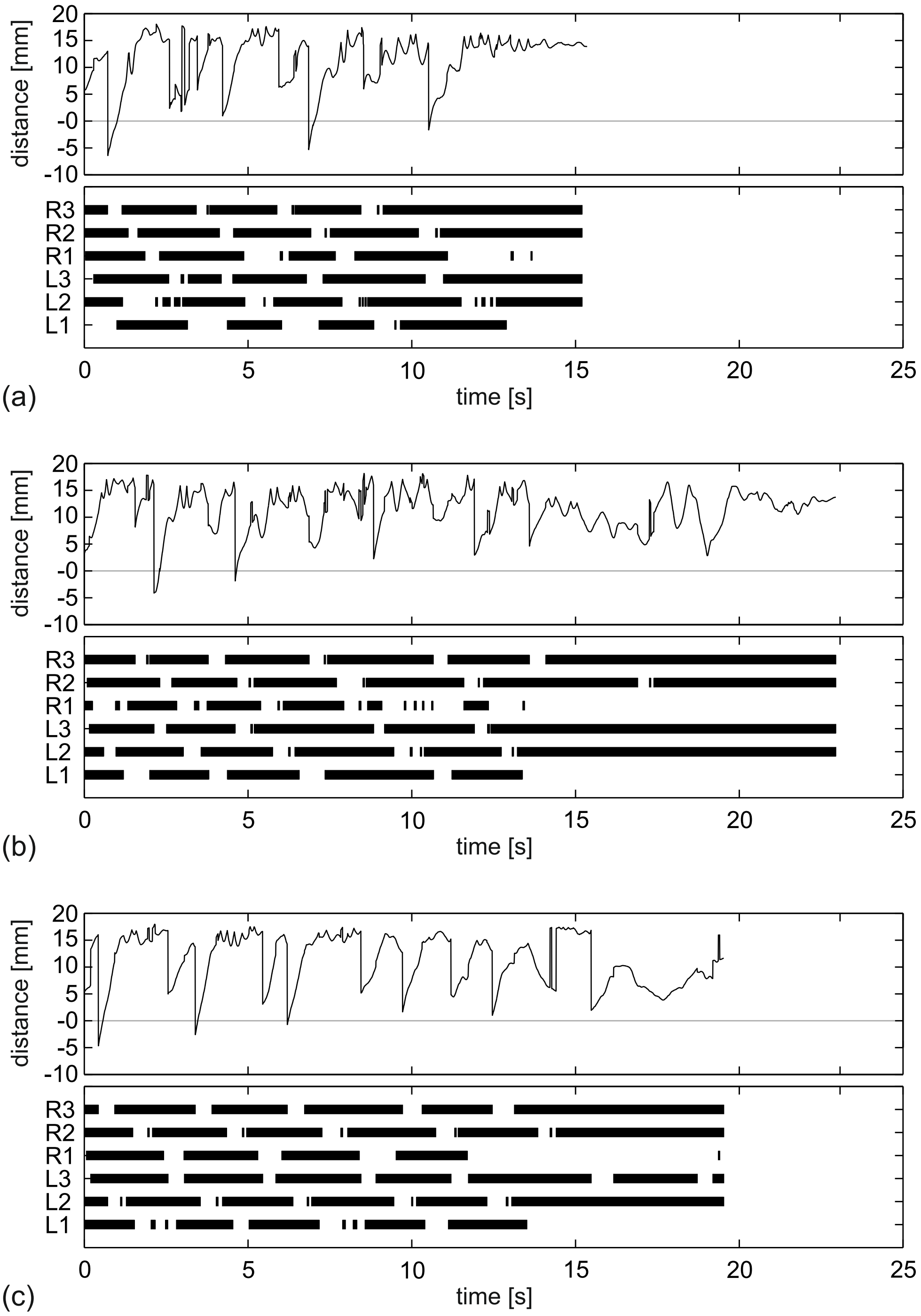


Figure 3.4.: Experimental data of walking stick insect *Medauroidea extradentata* (based on data recorded by Theunissen, Hertrich, et al. (2014); trials “Animal06_110919_00_15”, “[. . .]_16” and “[. . .]_31”, shown in (a)–(c)). For each trial the distance of the COM to the support polygon and the corresponding gait pattern over time is shown. A positive distance corresponds to a statically stable posture, a negative distance denotes an unstable posture.

seems unlikely. Since the stick insects, for which the kinematic data was recorded, had functioning adhesive pads and claws, these have likely been used by the insects to hold to the ground. Therefore, the insects did not have to maintain static stability to prevent tilting. Although the experiment by Kindermann (2003) already suggests the result, the kinematic data acquisition should be repeated on loose ground the insects cannot hold on to. This experiment would evaluate the ability of stick insects to maintain stability in situations, in which it is actually required to prevent tilting.

Although this evaluation shows that a direct replication of the biological controller would not be suitable for the control of robots that require static stability, Walknet, an abstracted stick insect - inspired controller, has often been used for the control of hexapod robots. To evaluate whether Walknet is able to achieve permanent stability, it was extensively tested in simulation for the control of the virtual robot (see chapter 4). In chapter 5, the bioinspired concept of leg coordination is fundamentally expanded to combine the adaptability of bioinspired leg coordination with the maintenance of static stability and additional technical requirements (e.g., adherence to torque constraints, avoidance of kinematic singularities).

4. Evaluation of Walknet Coordination Rules Regarding Maintenance of Static Stability

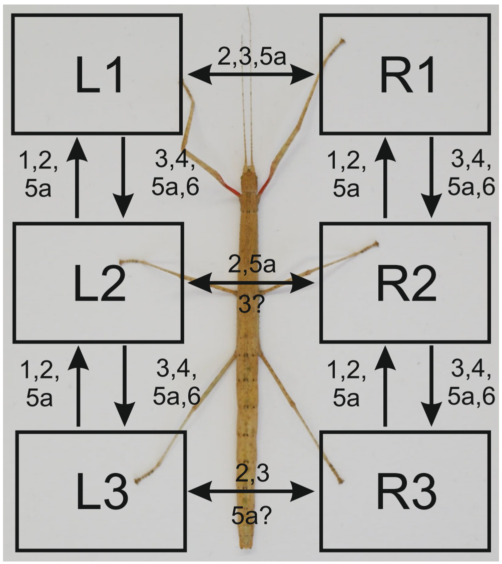
Abstract: *Based on observations of forward walking stick insects, a system of ANNs and distributed coordination rules has been developed by Cruse (1990). As a core concept, the so-called Walknet assumes individual controllers for each leg that decide between the desired movement (swing or stance) of the legs. The coordination between these controllers is established using rules that rely only on sparse information exchange between neighboring legs. Using this control concept, the stability of Hector is evaluated in simulation for two different walking speeds. Although the stability of the system can be maintained by the bioinspired controller for slow forward walking, the approach does not maintain permanent stability for fast walking.*
Some of the results and outcomes presented in this chapter have already been published in the following paper together with additional results from the specified co-authors: Paskarbeit, Otto, Schilling, and Schneider 2016

Walknet is a distributed ANN-based gait controller intended to simulate the behavior of stick insects during walking (Cruse 1990; Cruse, Kindermann, et al. 1998; Dürr, Schmitz, et al. 2004). However, the aim of the controller is not to mimic the exact structure of the neuronal network of the insect. Instead, a more abstract level was chosen for the layout of Walknet.

The key concept of Walknet is that each leg is controlled by a rather independent leg controller. Each of the leg controllers consists of ANNs for leg trajectory generation during the swing- and stance-phase (the corresponding ANNs are called *swing net* and *stance net*). The switching between these two states is performed based on proprioceptive information like leg tip position, ground contact and load sensing as well as sparse information on the current states of the neighboring legs. The latter is formulated in the form of so-called *coordination rules* which connect neighboring legs.

As discussed already in section 1.2.2, swing phases of adjacent legs usually do not overlap. This led to the conclusion that there must be an influence that inhibits the stance-to-swing switch of a leg if an adjacent leg is in swing phase at that moment. Based on this and similar observations the coordination rules were formulated.

Each rule is defined between a sending (controlling) and a receiving (controlled) leg. Using the coordination rules, a sending leg may influence the behavior of neighboring legs, e.g., by increasing or decreasing the receiving leg's disposition to switch from stance to swing phase. The coordination rules are based on behavioral experiments on stick



1. swinging sender inhibits start of swing phase of receiver leg
2. start of stance phase of sender facilitates start of swing phase of receiver leg
3. stancing sender excites swing of receiver leg such that the swing can be finished before the sender itself switches to swing
4. during swing, subsequent legs (receiver) target at the position of their preceding neighbors (sender)

5a. increase of resistance at the sender leg increases propulsive force in the receiver leg.

5b. load increase in the sender inhibits itself to start swing phase (sender=receiver)

6. if the receiver treads on the sender's tarsus, a correction step of the receiver will be triggered to free the trod-on tarsus.

Figure 4.1.: Influences between the six individual leg controllers according to Cruse, Bläsing, et al. (2004). The arrows depict the direction of influence of the coordination rules derived from studies on walking stick insects.

insects that usually walked on treadwheels or floating spheres with the thorax restrained to a holder (Cruse and Saxler 1980; Dürr 2005) or unconstrained on a horizontal surface (Graham 1972). Although some approaches to replicate leg coordination during curve walking were published (Dürr 2005), most works focus on forward walking. Therefore, in section 4.2, the ability of Walknet to maintain static stability is only evaluated for forward walking.

Current extensions of Walknet also incorporate cognitive and navigation functionality (Cruse and Wehner 2011; Schilling and Cruse 2017). In the original Walknet controller that will be used for the further analysis, however, the desired speed and direction of walking must be provided by an external source (in the experiments they are set to constant values). These specifications of walking speed and direction are directly used for the control of the legs. Therefore, the classical Walknet controller will always achieve the desired walking speed.

4.1. Description of Coordination Rules

In different behavioral experiments on stick insects, a coupling between the legs was observed which could be explained assuming the legs to influence each other by shortening or prolonging the stance phases of their neighboring legs (Cruse, Kindermann, et al. 1998). This was formalized in six coordination rules that describe the influences between the legs (Cruse, Kindermann, et al. 1998). As depicted in fig. 4.1, the coordination rules

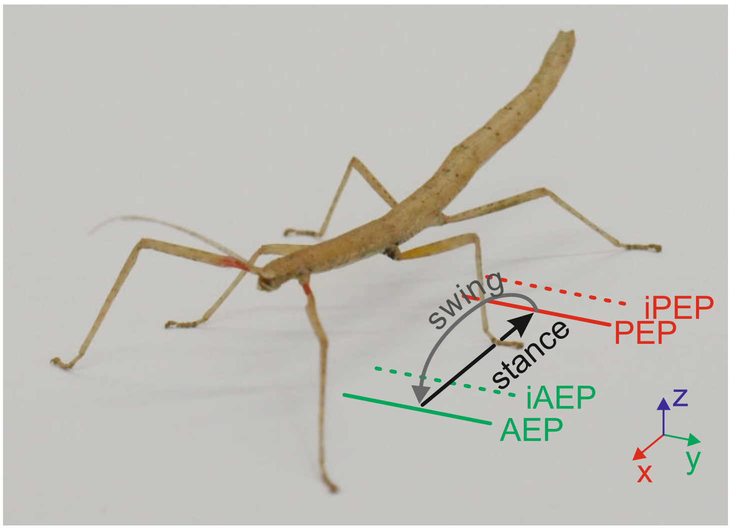


Figure 4.2.: Stick insect *Carausius morosus* with graphical depiction of the swing/stance cycle for the left middle leg. The relevant limits for the movement during stance phase are marked in red, the target lines during swing phase are marked in green.

are neither active between all pairs of neighboring legs nor in all directions. Moreover, the strength of influence of a coordination rule is specific for each of the respective pairs of legs that interact with each other (Dürr, Schmitz, et al. 2004).

As the original experiments that have led to the formalization of these rules have been conducted with forward walking insects, the rules have been formulated for forward walking as well. Therefore, as illustrated in fig. 4.2, the legs are assumed to always perform a movement cycle of stancing backwards, then swinging forwards (for backwards walking, this would be reversed). The position in space at which the leg switches from stance to swing phase is called PEP (Posterior Extreme Position), which translates to *rearmost position*. The position at which the leg detects ground contact at the end of the swing movement and thereupon switches from swing to stance phase is called AEP (Anterior Extreme Position), the *foremost position*. The AEP and PEP may vary as they are influenced by the states of the neighboring legs—their actual position can only be determined after the phase switch. However, AEP and PEP are also used synonymously to denote the points, at which the switches between stance and swing phases are planned to occur. Following this concept, the decision whether the leg should remain in stance phase is determined based on the PEP. When the leg tip reaches the PEP during the stance movement, the leg controller switches to swing phase. During swing phase, correspondingly, the leg trajectory targets at the AEP. When the leg reaches the AEP, the leg controller switches to stance phase. To coordinate the swing and stance phases of the legs, their AEPs and PEPs are shifted for- and backwards based on influences of neighboring legs. These shifts are mediated by coordination rules that

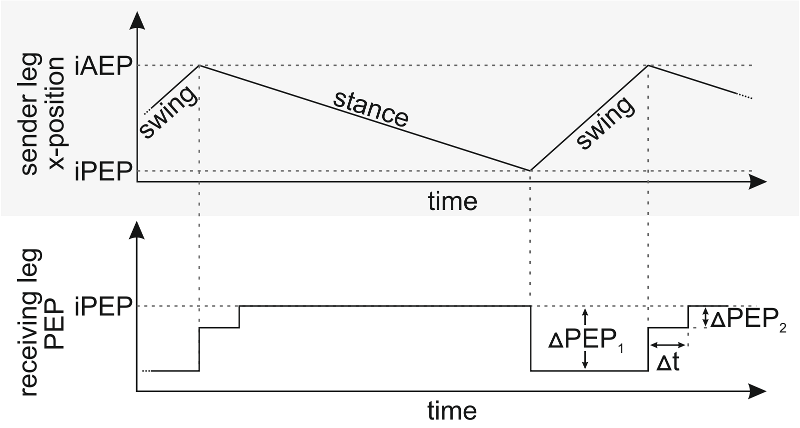


Figure 4.3.: Influence of rule 1 on the PEP of the receiver leg. The upper plot shows the sequence of stance and swing movements for the undisturbed sender leg (iAEP/AEP and iPEP/PEP coincide). The lower plot shows the change of the PEP of the receiver leg relative to its iPEP. When the sender leg is in swing phase, the PEP of the receiver leg is shifted backwards by ΔPEP_1. For a duration Δt that is defined by eq. (4.1) (see also fig. 4.4) after the sender leg touched down, the receiver leg's PEP is still shifted by ΔPEP_2.

will be described in the following paragraphs. Their static base positions, therefore the PEP and AEP if they are not shifted due to coordination influences, are denoted iPEP (intrinsic PEP) and iAEP (intrinsic AEP).

In most Walknet-related works, only the x-components of the leg tip position and the PEP are compared to initiate the stance-swing-switch. Thus, the PEP can be thought of as a vertical plane that is shifted forwards and backwards by the coordination rules relative to the iPEP. If the leg tip crosses this plane, the swing phase is initiated. Although alternative concepts for PEP and also AEP definition have been envisioned that will be described in section 4.1.1, the following descriptions of the coordination rules refer to the classical definition based only on the x-coordinates. Also, since the first three coordination rules have been explicitly formulated by Schilling, Hoinville, et al. (2013) for the application on Hector, the given measures always refer to the robotic system. Coordination rules 4–6 have only been formulated qualitatively and have not been implemented for the stability analysis. However, for completeness, their intended functionality will be described briefly.

Rule 1 delays the start of the swing phase of the receiver if the sender leg is in swing phase and additionally also for a short time after touch-down (Cruse 1990). This is accomplished by shifting the PEP of the receiver leg backwards relative to the iPEP. Figure 4.3 shows the influence of coordination rule 1 between two legs. This concept has been considered as explanation for wave gait generation in insects already by Wilson

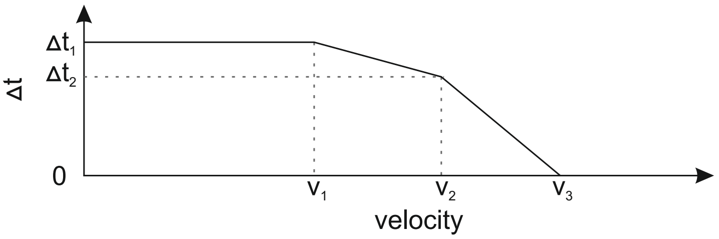


Figure 4.4.: Velocity-dependent period of time during which rule 1 shifts the receiver leg's PEP after the sender leg touched down.

(1966).[1] Experiments by Cruse and Epstein (1982) and Graham (1972) have supported the theory of anteriorly directed, inhibitory influences.

The time for which rule 1 inhibits the receiver leg after the sender leg finished its swing phase is assumed to be velocity dependent. Schilling, Hoinville, et al. (2013) suggested the following dependency of the deactivation delay Δt from the walking velocity v that is also visualized in fig. 4.4:

$$\Delta t(v) = \begin{cases} \Delta t_1 & \text{if } v \leq v_1 \\ \Delta t_1 + \frac{\Delta t_1 - \Delta t_2}{v_1 - v_2}(v - v_1) & \text{if } v_1 < v \leq v_2 \\ \Delta t_2 + \frac{\Delta t_2}{v_2 - v_3}(v - v_2) & \text{if } v_2 < v \leq v_3 \\ 0 & \text{if } v_3 < v \end{cases} \tag{4.1}$$

While the sending leg is in swing phase, the PEP of the receiving leg is shifted by ΔPEP_1. After the sending leg switched to stance phase the PEP is shifted by ΔPEP_2 for the duration given in eq. (4.1). The numerical values for the constants are:

$$\Delta PEP_1 = -0.20\,\mathrm{m} \tag{4.2}$$

$$\Delta PEP_2 = -0.05\,\mathrm{m} \tag{4.3}$$

$$v_1 = 0.26\,\frac{\mathrm{m}}{\mathrm{s}} \qquad \Delta t_1 = 0.27\,\mathrm{s} \tag{4.4}$$

$$v_2 = 0.4\,\frac{\mathrm{m}}{\mathrm{s}} \qquad \Delta t_2 = 0.2\,\mathrm{s} \tag{4.5}$$

$$v_3 = \frac{0.8}{1.5}\,\frac{\mathrm{m}}{\mathrm{s}} \approx 0.533\,\frac{\mathrm{m}}{\mathrm{s}} \tag{4.6}$$

Rule 2 exerts an excitatory influence on the receiver leg to start its swing phase shortly after the sender leg begins its stance phase. As depicted in fig. 4.5, the influence is activated after a period of Δt_1. The influence remains active only for a short interval of Δt_2.

[1] Wilson's preferred alternative to the inhibitory influence that prolongs the stance phase of the next anterior leg was an excitatory influence that shortens the stance phase of the posterior leg. This concept was also considered by Cruse (1979b) before it was reformulated as inhibitory connection.

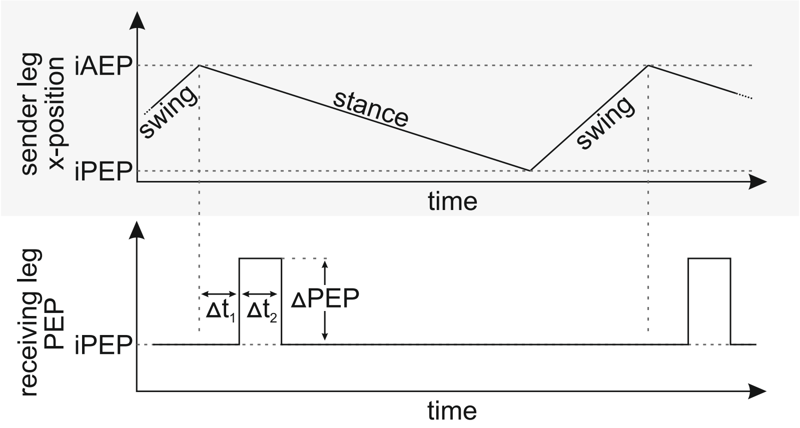


Figure 4.5.: Influence of rule 2 on the PEP of the receiver leg. The upper plot shows the sequence of stance and swing movements for the undisturbed sender leg (iAEP/AEP and iPEP/PEP coincide). The lower plot shows the change of the PEP of the receiver leg relative to its iPEP. When the sender leg is in ground contact after touch down for a period of Δt_1 the PEP of the receiver leg is shifted forwards by ΔPEP, thus facilitating a switch to swing phase. The PEP shift is maintained for a period of Δt_2.

This coordination rule could be demonstrated between ipsilateral (Cruse and Schwarze 1988) and contralateral legs (Cruse and Knauth 1989) in stick insects. However, the strength of influence differs for contra- and ipsilateral leg pairs.

The numerical values for the constants are

$$\Delta t_1 = 0.27\,\mathrm{s} \tag{4.7}$$

$$\Delta t_2 = 0.05\,\mathrm{s} \tag{4.8}$$

$$\Delta PEP = 0.08\,\mathrm{m} \qquad \text{for ipsilateral leg pairs} \tag{4.9}$$

$$\Delta PEP = 0.02\,\mathrm{m} \qquad \text{for contralateral leg pairs} \tag{4.10}$$

Rule 3 excites the switch to swing phase of the receiver leg such that is can finish its swing phase before the sender leg reaches the end of its workspace. The effect is only active while the sender leg is located within a defined region of its workspace (see fig. 4.6). As long as the leg is in this region, the PEP of the receiver leg is shifted forwards in order to excite a switch to swing phase. The amount by which the PEP is shifted is

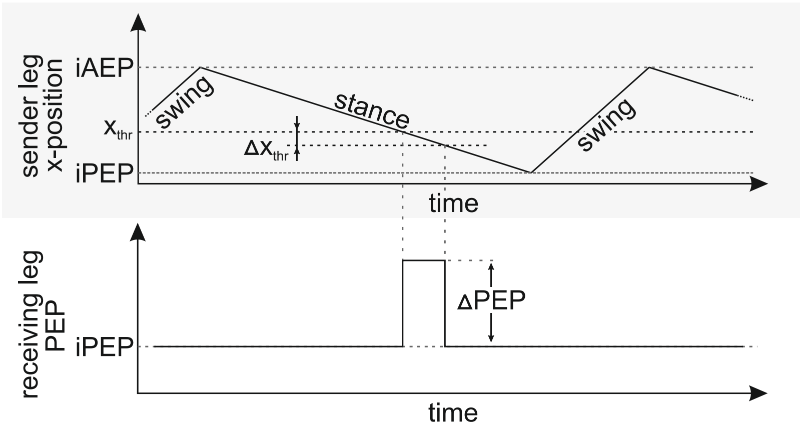


Figure 4.6.: Influence of rule 3 on the PEP of the receiver leg. The upper plot shows the sequence of stance and swing movements for the undisturbed sender leg (iAEP/AEP and iPEP/PEP coincide). The lower plot shows the change of the PEP of the receiver leg relative to its iPEP. When the sender leg arrives at position x_{thr} the PEP of the receiver leg is shifted forwards by ΔPEP. This influence is maintained until the sender leg moved Δx_{thr} further towards its PEP (see fig. 4.7).

specific for each leg pair as listed below:

$$\Delta PEP = 0.125\,\mathrm{m} \qquad \text{for ipsilateral leg pairs} \tag{4.11}$$

$$\Delta PEP = 0.03\,\mathrm{m} \qquad \text{for contralateral front leg pairs} \tag{4.12}$$

$$\Delta PEP = 0.0\,\mathrm{m} \qquad \text{for contralateral middle leg pairs} \tag{4.13}$$

$$\Delta PEP = 0.01\,\mathrm{m} \qquad \text{for contralateral hind leg pairs} \tag{4.14}$$

The region the sender leg must be within to activate the influence shifts depending on the walking speed as depicted in fig. 4.7. The functions to compute the threshold are:

$$x_{\mathrm{thr}}(v) = \frac{iAEP - iPEP}{1 + e^{4m_i\frac{(v_{1i}-v)}{m}}} \qquad \text{for ipsilateral leg pairs} \tag{4.15}$$

$$x_{\mathrm{thr}}(v) = x_1 + \frac{v_{1c}}{iAEP - x_1}v \qquad \text{for contralateral leg pairs} \tag{4.16}$$

with m_i being the slope of the curve's tangent at v_{1i} (see fig. 4.7 (a)). The numerical values for the constants are

$$v_{1i} = 0.37\frac{\mathrm{m}}{\mathrm{s}} \quad m_i = 3.75\,\mathrm{s} \qquad \text{for ipsilateral leg pairs,} \tag{4.17}$$

$$v_{1c} = 1\frac{\mathrm{m}}{\mathrm{s}} \quad x_1 = iPEP + \frac{iAEP - iPEP}{2} \qquad \text{for contralateral leg pairs.} \tag{4.18}$$

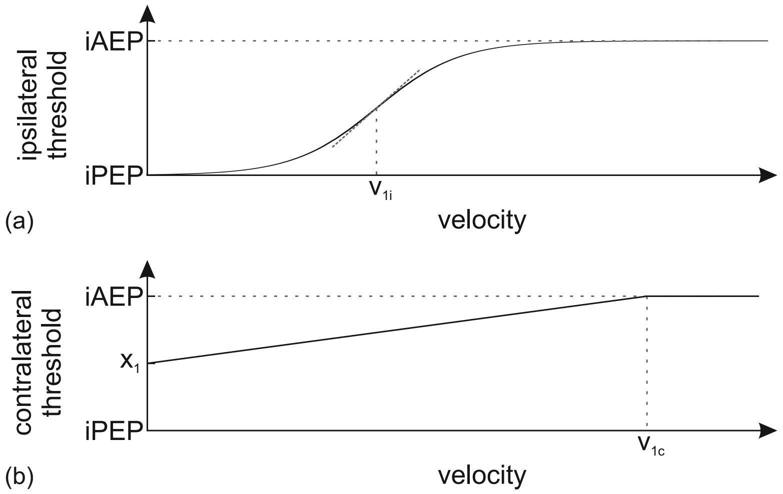


Figure 4.7.: The threshold x_{thr} at which the influence of rule 3 becomes active depends on the velocity of the robot. Different functions are used for (a) ipsilaterally neighboring legs (see eq. (4.17)) and (b) contralateral neighbors (see eq. (4.18)).

Rule 4 shifts the receiving leg's AEP, the target position of the swing movement, close to the current position of the sending leg (Cruse 1979b; Cruse, Dean, et al. 1991). This rule acts only ipsilaterally and backwards. Therefore, hind legs target at the middle legs' and middle legs target at the front legs' positions.[2] This rule is presumed to help the legs find contact if the insect climbs on branches with sparse footholds.

Rule 5a increases the backwards-directed force of the receiving leg if the sender leg registers an increase in resistance (Cruse, Bläsing, et al. 2004).

Rule 5b prolongs the stance phase of a leg if it detects a high load. In this case, sender and receiver leg are identical.

Rule 6 enforces a correction step if the receiving leg trod on the sending leg's tarsus (treading-on-tarsus reflex; Schmitz and Haßfeld 1989). This is assumed to avoid stumbling.

The delays of rules 1 and 2 might partially reflect the low conduction velocity in the nerve fibers of insects. Conduction velocities in the range of $0.5-11$ m/s have been reported for different insects (Chapman and Pankhurst 1967; Pearson et al. 1970). Assuming a similar range of conduction velocities in the stick insect the transmission from a front leg *tarsus* to the ipsilateral middle leg *coxa* would require $4-95$ ms (at a distance of 47.5 mm, Cruse (1976b)). However, this effect cannot be the sole cause of the delays in coordination rules 1 and 2 as these reach up to 270 ms. Also due to the video-based, therefore purely

[2] Also, the front legs target at positions, where the corresponding ipsilateral antenna detected a collision (Schütz and Dürr 2011).

kinematic data acquisition that was used for the development of Walknet, another effect might contribute to the delays: Since the contact forces between the insect legs and the ground were not measured, the stance phase was assumed to start as soon as the *tarsus* touched the ground. The insect, however, might detect ground contact with a delay since the load in the leg must rise above a certain threshold before ground contact is perceived. Thus, for a short time, the leg controller would remain in swing mode although the tarsus has already touched down.

4.1.1. Alternative PEP Definitions

As previously described, the (i)PEP is often defined as an x-position the leg must cross to initiate the swing phase. During experiments with stick insects, however, Burkamp (1996) found that the legs tend to start the swing phase earlier if the tarsus position is shifted downwards or outwards. To reproduce the dependency of the x-coordinate of the PEP on the y- and z-coordinates of the leg tip position, a feedforward ANN with three hidden units was trained (*PEP net*). With the input of the three joint angles of a leg, the *PEP net* decides whether the leg should switch to swing phase or remain in stance phase. Using this neural network, the experimentally obtained data from stick insects can be reproduced with a high accuracy. In analogy to the straight plane, the (i)PEP can be thought of if only the x-coordinates are considered, the result of the *PEP net* can be thought of as a slightly curved version of this plane. However, also in the model presented by Burkamp (1996) this curved surface is shifted along the x-axis due to the influences of the coordination rules. Since the surface does not completely envelop the workspace of the legs but rather stretches to the side endlessly, backward walking is not possible with these (i)PEP definitions. Also turning is problematic since the movements of the legs have sidewards components (see fig. 6.14). For sharp turns, situations may occur in which legs are completely extended before they would reach the PEP surface.

To overcome this problem, Espenschied and Quinn (1994) defined a combined measure based on the leg tip position $\boldsymbol{p}$, the constant *home position* $\boldsymbol{p}_\mathrm{h}$ that is located in the center of the workspace of each leg, and the velocity vector $\boldsymbol{v}$ of the leg tip:

$$x = \frac{(\boldsymbol{p} - \boldsymbol{p}_\mathrm{h}) \cdot \boldsymbol{v}}{\|\boldsymbol{v}\|} \tag{4.19}$$

The result, the scalar x, is compared to the value of the PEP to determine whether the leg should switch to swing phase. For a constant leg tip velocity this corresponds to the definition of the PEP as vertical, cylindrical surface whose radius is varied by the coordination rules. By considering the variable leg tip velocity $\boldsymbol{v}$, a switch to swing phase is delayed if the leg moves towards the home position, the center of the PEP-cylinder.[3] The AEP is modeled as cylindrical surface as well, however, with a constant radius. The target position for the swing movement is evaluated by reverse prolongation of the last leg tip velocity of the stance movement, starting from the home position. The intersection

[3] Omitting the velocity component of eq. (4.19), this approach was also used by Görner, Wimböck, and Hirzinger (2009) to utilize the coordination rules for the generation of omnidirectional walking.

of this line with the AEP-cylinder is used as swing target. Although the definitions of AEP and PEP are not bioinspired, they allow the robot to walk omnidirectionally since the workspaces of the legs are limited horizontally in all directions (Espenschied, Quinn, Chiel, et al. 1995). However, using cylindrical, vertically unlimited surfaces, this method is not applicable for extreme leg postures with very high or low stance heights. In these cases, the leg might be completely stretched, therefore unable to move further, before it reaches the PEP-surface.

4.1.2. Previous Implementations of the Coordination Rules for Application on Robots

Since the coordination rules introduced above do not explicitly guarantee static stability, most roboticists have used additional mechanisms that are meant to increase or even enforce stability. The most prominent discrepancy in the interpretation of the coordination rules between biologists and roboticists is the direction of rule 1. In the stick insects rule 1 seems to be directed exclusively backwards. Also, since it only delays the protraction of the controlled leg, it does not actually prohibit adjacent legs to swing at the same time. To increase the stability of robotic systems, rule 1 was often interpreted as a strict prohibition for adjacent legs to swing simultaneously (Beer et al. 1992; Fielding 2002). On the walking machine *MAX* by Pfeiffer (2007), rule 1 acted rostrally between ipsilateral legs (in accordance with the biological findings) but was also used contralaterally between legs.

Few implementations adhere to the coordination rules as defined by Cruse, Kindermann, et al. (1998). Among these are the implementations by Schmitz, Schneider, et al. (2008) and Ferrell (1995). Schmitz, Schneider, et al. (2008) focused on the application of positive feedback controllers in order to replace a conventional kinematic body model. For this implementation an analysis regarding the static stability of the robot was not conducted (A. Schneider, personal communication, October 2015). Ferrell compared three different bioinspired controllers on the robot *Hannibal* regarding their ability to reproduce the continuous coordination patterns of stick insects and regarding the static stability that can be obtained. Although the biological coordination patterns could be reproduced, the robot showed occasional instabilities during walking.

Flannigan et al. (1998) used rules 1 and 3 on an eight-legged crab-like robot. In this work, the legs were divided into two groups—four legs each—that moved in unison. Therefore, stability is not an issue as long as the COM does not exit the support polygon of each individual group.

Görner and Hirzinger (2010) used the first three coordination rules but added a strict prohibition of simultaneous swing phases of adjacent legs, which is equivalent to rule 1 acting strongly in all directions.

Espenschied, Chiel, et al. (1993) employed the coordination rules as described by Cruse and did not detect instability as long as all three rules were active. However, it was pointed out that the strength of rule 3 was tuned very high as compared to the biological data of Dean (1991) to gain stability.

coordination rules	tunable parameters						
rule 1 ipsilateral	ΔPEP_1	ΔPEP_2	v_1	v_2	v_3	t_1	t_2
rule 2 ipsilateral	ΔPEP_{ips}					t_1	t_2
rule 2 contralateral	ΔPEP_{con}						
rule 3 ipsilateral	ΔPEP_{ips}		v_{1i}	m_i			
rule 3 contralateral	ΔPEP_{con}		v_{1c}	x_1			

Table 4.1.: Parameters influencing leg coordination, listed for coordination rules 1–3.

4.2. Evaluation of Static Stability of Hector at Varying Coordination Rule Strengths

Following the evaluation of stability of walking stick insects that is described in chapter 3, the stability of Hector was evaluated in simulation using the coordination rule parameters published by Schilling, Hoinville, et al. (2013). A preliminary test showed that the stability depends strongly on the initial posture of the robot. Although static stability could be maintained during walking if the robot was initialized in specific postures, other starting postures led to instable locomotion. To evaluate the ability of the classical coordination rules to coordinate the swing and stance phases of the legs such that the stability of the robot is permanently maintained, different parameter sets were tested for the control of the robot in the simulation framework (see section 2.5).

4.2.1. Test Setup for Evaluation of Static Stability

As described in section 2.2, there are considerable differences between the robot and the biological model. To evaluate the classical, bioinspired coordination rules regarding their usability in the control of Hector, they were tested based on the implementation published by Schilling, Hoinville, et al. (2013). Since only coordination rules 1–3 were explicitly formulated and implemented by Schilling, Hoinville, et al. (2013), the evaluation was constrained to these. As discussed in section 4.1.2, most implementations of Walknet intended for the control of robots omit coordination rules 4–6 as well.

As a quality measure for the evaluation, the static stability of the system is used: If the projection of the robot's COM along the gravity vector stays within the support polygon the robot is considered statically stable; if the projection is outside, the robot is considered unstable. During each experiment the fraction of stable controller iterations per total iterations is determined. A fraction of 1 implies that the robot remained statically stable at all times during the experiment, a lower fraction signifies occasional instabilities. However, the consequences of these instabilities like tilting were not further evaluated.

The relevant variables that influence the leg coordination as specified in section 4.1 are listed, ordered by coordination rules, in table 4.1. For a comprehensive analysis, each of these 17 variables would need to be tested for a representative selection of values in all possible combinations. Moreover, in the reference (Schilling, Hoinville, et

coordination rules and modified parameters	tested parameters				
rule 1 (ΔPEP_1)	-0.5 m	-0.375 m	-0.250 m	-0.125 m	0.00 m
rule 2 ipsilateral (ΔPEP_{ips})	0.0 m	0.063 m	0.125 m	0.188 m	0.25 m
rule 3 ipsilateral for front legs (ΔPEP_{ips})	0.0 m	0.063 m	0.125 m	0.188 m	0.25 m
rule 3 contralateral (ΔPEP_{con})	0.0 m	0.063 m	0.125 m	0.188 m	0.25 m

Table 4.2.: Sets of parameter values that were tested in all combinations.

al. 2013) the coordination strengths of rules 2 and 3 vary for different combinations of leg pairs, which would further increase the number of test parameters. Since this would considerably exceed the available computing time, only the strengths of the coordination rules, the ΔPEPs, were varied as these are considered most relevant for leg coordination (see table 4.2 for sets of tested values). To further reduce the number of variables, the ratio of ΔPEP_1 to ΔPEP_2 in rule 1 and the ratio of ΔPEP_{ips} to ΔPEP_{con} in rule 2 were kept at 4:1 as in the implementation described by Schilling, Hoinville, et al. (2013). In rule 3, due to the different equations for ipsilateral and contralateral excitation of swing, the ΔPEPs for contralateral and ipsilateral influence were varied independently. However, the ratio of the contralateral coordination strengths for front and hind legs was held constant at 3:1. Therefore, only four parameters were tested systematically. The distribution of values that were chosen for the evaluation (see table 4.2) were not aligned to the published parameters (see section 4.1 Schilling, Hoinville, et al. 2013) since a continuous progress of the stability measure was expected. Therefore, results for the parameter combination that was published by Schilling, Hoinville, et al. (2013) must be interpolated based on the data presented in fig. 4.10.

Each experiment was conducted starting from three different postures (see fig. 4.8 (a)) to eliminate the case of coincidentally good combinations of coordination rule strengths and starting postures. Correspondingly, the results will always refer to the worst of the three trials.

4.2.2. Static Stability During Forward Walking

To evaluate the suitability of the bioinspired coordination rules for application on the hexapod robot Hector, the static stability of the robot was evaluated in simulation during forward walking at two different speeds: 0.15 m/s ("slow") and 0.25 m/s ("fast").[4] Although the robot was supposed to walk straight forwards, the actual directions deviate from the supposedly straight line. This is due to the elasticity in the joints that influences the direction of walking. To counteract the variance of the trajectories if the robot

[4] Since static stability is achieved more likely at lower speeds with more legs in stance phase, both tested speeds were chosen to be rather low (in the lowest range for the computation of the deactivation delay; see eqs. (4.1) and (4.4)).

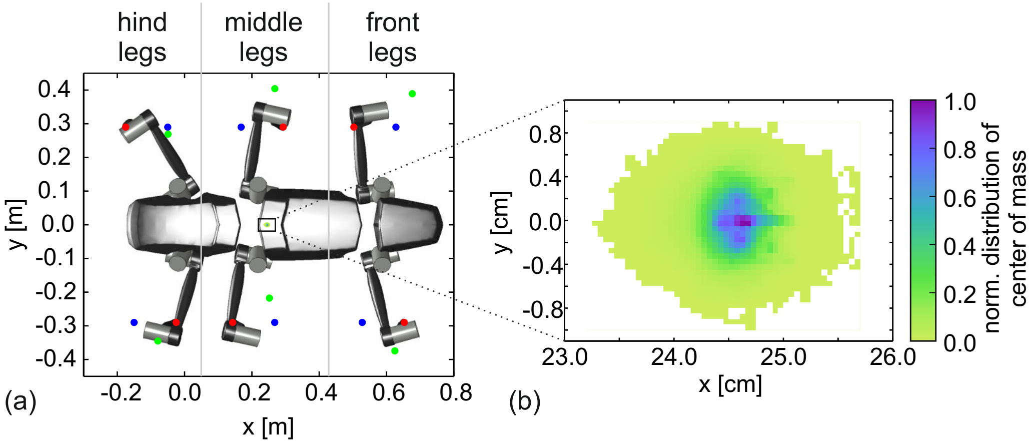


Figure 4.8.: (a) shows the three different starting postures of the robot, depicted by three differently colored points per leg. In all postures the legs start with ground contact. In the center of the plot, marked by a black frame, the probability distribution of the COM is depicted. The exact position of the COM varies depending on the posture of the robot. The plot shows the normalized probability distribution over all conducted experiments. In (b), a magnified plot of the probability distribution is shown.

deviated from the straight trajectory, it was directed back towards it (using a simple direction controller with P-characteristic). Exemplary trajectories of the COM of the robot during forward walking are shown in fig. 4.9.

To detect ground contact, force sensors were simulated at the leg tips. At a minimal force of 1 mN between the leg and the ground, ground contact was assumed. Beside this binary ground contact detection, the exact contact force was not considered in the control. Therefore, just as assumed during the kinematic observations on stick insects that led to the development of Walknet, the stance phase of a leg is started immediately after touch-down, therefore practically independent of the contact force.

Figure 4.10 (a) and (b) show the fraction of stable iterations per total number of iterations for different coordination rule parameters at two walking speeds in four-dimensional plots. In (a), the results for slow walking are depicted. On the major axes (lower x- and left y-axis), the parameters for rules 1 and 2 are varied. On the minor axes (upper x-axes, right y-axes), the parameters of rule 3 for ipsilateral and contralateral influences are varied. Therefore, each of the 25 subplots corresponds to one parameter combination of rules 1 and 2 and gives the variation of stability for different ipsilateral and contralateral parameters of rule 3. As can be seen, static stability was maintained for different parameter combinations. It is noticeable that a continuous stability (marked by red frames) is only reached for lower values of the contralateral parameter of rule 3 in the range of 0-0.063 m. The deactivation of rule 1, corresponding to values of 0 m, results in

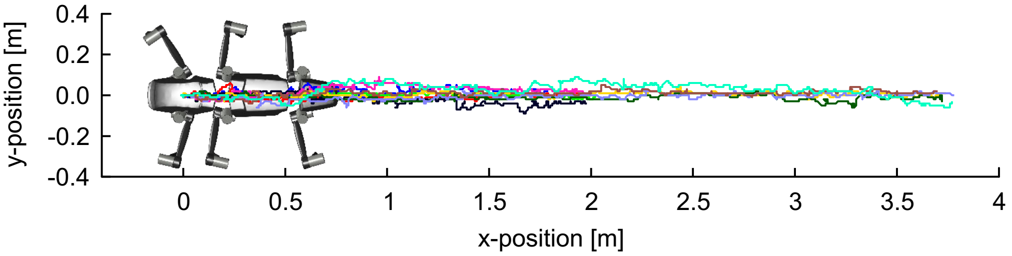


Figure 4.9.: Exemplary trajectories of the COM for straight walking. The trajectory was corrected by means of a direction controller (P-characteristic) if the robot deviated from the desired path.

a lowered stability. For none of these parameter combinations, permanent stability was achieved. Since the subplots often show vertical bars of equal stability, the ipsilateral influence of rule 3 seems to be of minor relevance.

As mentioned before, the distribution of the test values for the four ΔPEP-parameters was not aligned to the values that were published by Schilling, Hoinville, et al. (2013). However, for reference, the relative positions of the published values are marked by green dots on the respective axes. Except for the ipsilateral influence of coordination rule 3, these do not coincide with the values, for which the stability was evaluated in this experiment. Therefore, multiple parameter sets, highlighted by green frames, are in the vicinity of the published parameter combination. For slow walking only one of these close-by fields overlaps with a region of static stability (marked by red frames). In general, however, the regions of static stability are located at higher ΔPEP-values of rule 2 compared to the published values.

For fast walking at a speed of 0.25 m/s, static stability could not be achieved for any parameter combination. The corresponding results are depicted in fig. 4.10 (b). Still, for certain parameter sets a high stability fraction can be achieved. For example, the combination of rule 1 acting with -0.125 m, rule 2 acting with 0.125 m, rule 3 acting with 0.063 m contralaterally and 0.25 m ipsilaterally reaches the highest stability fraction of 0.94. This parameter set also achieves static stability during slow walking. However, the result connotes that, while walking fast, the system was instable during at least 6 % of the iterations. Compared to the parameters published by Schilling, Hoinville, et al. (2013), the regions of highest stability are located at higher ΔPEP-values of the ipsilaterally-acting rule 3.

For the evaluation of static stability, the COM of the robot was computed during each controller iteration. A subsequent analysis of all COM positions, depicted in fig. 4.8 (b), showed that the COM shifts only within a small region during walking. Since the α-drives do not move relative to the body of the robot and the relative movements of the β-drives are restricted by the short coxae, only the relative positions of the γ-drives influence the location of the COM noteworthy. The γ-drives, however, amount only to $\sim$ 20 % of the mass of the robot. Moreover, the influences of the legs usually counterbalance since

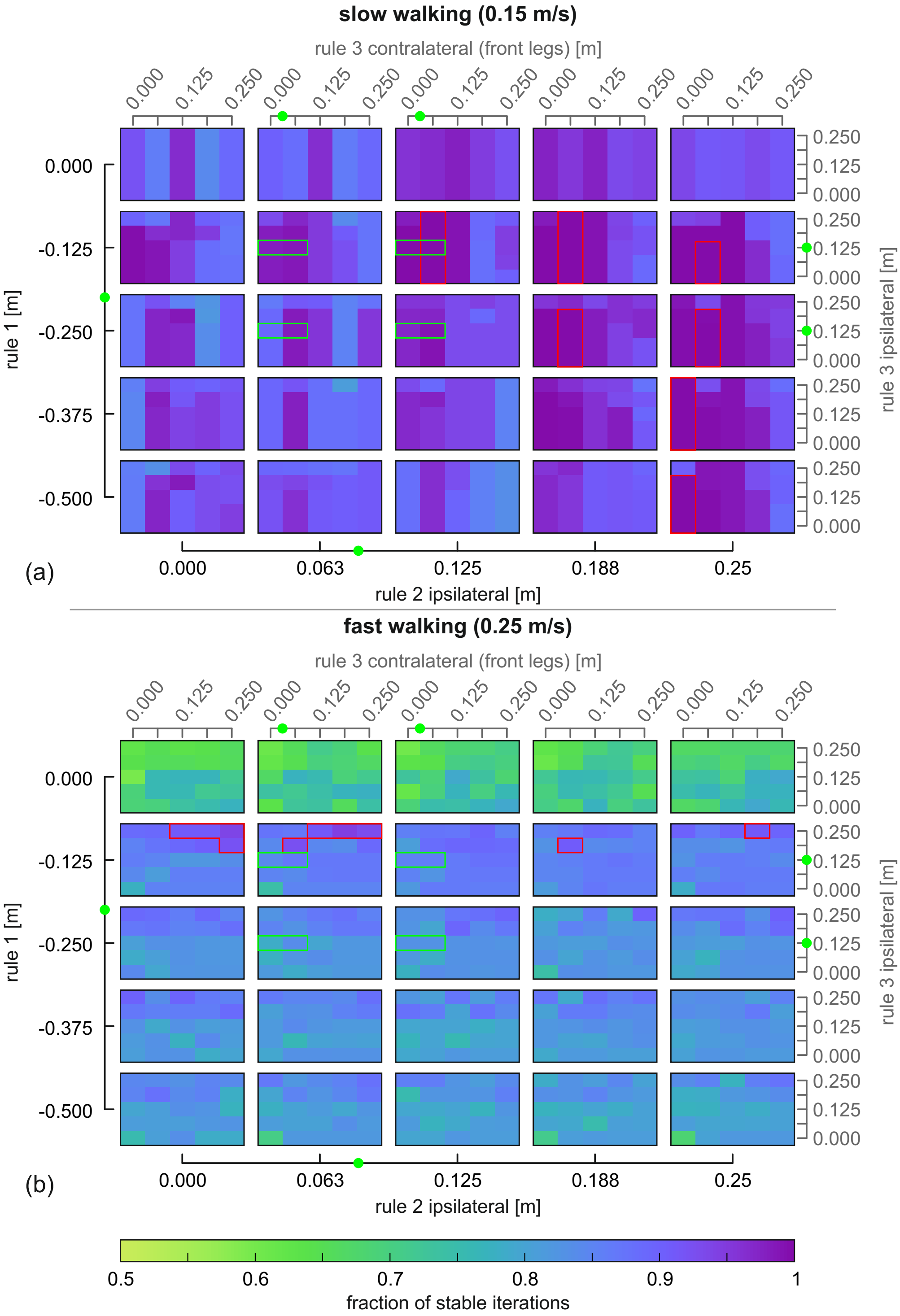


Figure 4.10.: Fraction of statically stable iterations per total number of iterations for experiments, during which the robot was supposed to walk straight forwards. (a) shows the results for slow walking (0.15 m/s), (b) shows the results for fast walking (0.25 m/s). In (a), all parameter combinations that reach static stability at all times are marked by red frames. In the results depicted in (b) continuous static stability is not reached by any parameter set. However, the 10 best results (stability fractions exceeding 0.92) are highlighted by red frames. The parameter sets close to the values published by Schilling, Hoinville, et al. (2013) are highlighted by green frames. For reference, the locations of the published values are marked by green dots on the axes.

some of the legs are in the anterior range of their workspaces whereas the other legs are in the posterior ranges. Therefore, the relative position of the COM is almost constant.

In the stick insect most of the mass is concentrated in the body segments of thorax and abdomen, whereas the legs are rather light-weight. Thus, the relative position of the insect's COM can be assumed to be nearly constant as well. With respect to this property, insect and robot therefore resemble each other.

Summary

The results of chapter 3 have already indicated that stick insects do not rely on the maintenance of static stability due to their ability to hold to the substrate with their tarsi. HECTOR and many other multi-legged robots (e.g., *Lauron I-V*, *Scorpion I-IV*), however, must maintain static stability at all times to prevent tilting since they lack the insects' adhesive pads and claws. Nevertheless, stick insects have often been used as model systems for the development of robotic walking controllers.

To evaluate the applicability of the bioinspired walking controller *WALKNET* (as proposed by Schilling, Hoinville, et al. (2013)) on HECTOR, the stability of the system was evaluated during forward walking at two different speeds. By systematic modification of the strengths of the coordination rules, multiple parameter sets were found that achieve static stability during slow walking. For fast walking, however, no parameter set could permanently maintain static stability.

During this evaluation, only four of the multitude of parameters that influence leg coordination within WALKNET were varied. Moreover, only the first three coordination rules have been implemented. Therefore, this analysis cannot be seen as proof that no parameter combination exists that might maintain static stability during fast walking. Probably, for most walking scenarios, a combination of parameters can be found that achieves a high if not perfect static stability. However, since the stability of the system is not incorporated on the control-level of WALKNET, it is also likely that for each combination of parameters, an unfavorable starting posture or walking speed can be found that would induce an instability.

Since the evaluation included only a small variety of settings, even for those parameter sets that achieved statically stable walking at slow speed, maintenance of stability cannot be guaranteed if, for instance, an unfavorable starting posture is chosen. On the other hand, occasional instabilities as detected during fast walking do not necessarily mean that the robot would inevitably tumble and crash. A leg that is in swing phase during the instability might stop the falling motion or the duration of the instability might be even too short to result in a notable motion at all. However, since the supervision of stability is not envisaged within the concept of WALKNET, instabilities and consequential crashes cannot be precluded.

In chapter 5, the control approach of WALKNET is extended to permanently maintain static stability and to support omnidirectional walking. Whereas in the classic implementation of WALKNET the desired walking velocity will always be realized by the controller (even at the cost of stability), in the extended controller the maintenance of stability is in the focus (if necessary at the cost of walking speed).

5. Bioinspired Control of a Hexapod Robotic System

Abstract: *The bioinspired control approach that is supposed to be used in the control of* Hector *is based on observations of forward walking stick insects (Cruse, Kindermann, et al. 1998). Due to physical and functional differences between the biological model and the robot, such as the inability of the robot to cling to the substrate, this control concept must be adapted for the application in the control of* Hector*. Also, extensions to facilitate omnidirectional walking are introduced for the different modules that are required for the control of the robot. To combine the omnidirectional walking with the classical bioinspired coordination rules, an* unrestrictedness *is defined for each leg. Based on the* restrictedness*, a concept developed by Fielding and Dunlop (2001), the* unrestrictedness *can be used to reformulate the coordination rules for the application in omnidirectional walking.*

Some of the results and outcomes presented in this chapter have already been published in the following paper together with additional results from the specified co-authors: Paskarbeit, Schilling, Schmitz, and Schneider 2015

The goals in the design of the walking controllers that were presented in section 1.2 can be roughly divided into the categories “confirmation of a biological hypothesis” and “control of a technical system”.

In general, the walking controllers of the *first category* emphasize the simplicity of the approach. Using neural networks, elegant and lean controllers can be created that replicate a versatile range of behaviors (Goldschmidt et al. 2012; Schilling, Hoinville, et al. 2013). Since the focus lies on the reproduction of animal behavior, the resulting walking controllers are most suitable for robotic systems that resemble the respective animal both in terms of their goals as well as morphologies and abilities. However, only few insectoid robots have comparable physical properties (including size and mass) as their biological model (counterexamples were presented by Baisch et al. 2011; Koh et al. 2015; Ma et al. 2013). Additionally, insects have abilities that are hard to replicate in a real robot, such as the ability to cling to the ground, which can be easily implemented in a simulation.[1] As a consequence, for robotic systems that deviate from the biological model, the results will deviate as well.

Walking controllers of the *second category* are typically designed to comply with the requirements of a specific technical system while performing specified tasks. Usually

[1] The tendency to use robots instead of simulations for these proof of concept applications is probably induced by a general skepticism regarding simulation results (Brooks 1992; Jakobi et al. 1995; Webb 2001).

the requirements include (static) stability at all times. These controllers are often either limited to certain predefined gaits (Kerscher et al. 2008) or they tend to be more complex and therefore computationally expensive (Chen and Kumar 1996).

In this chapter, a walking controller will be presented that combines the simplicity of bioinspired controllers with the requirements of technical systems. For this purpose it is important to identify the differences between the biological model and the technical system. The most important differences between HECTOR and its biological model the stick insect *Carausius morosus*—in the context of this work—are:

- The robot cannot cling to the ground.
- The robot does not have an abdomen that can be used to support the body if tilting backwards.
- The robot has comparatively low friction in the joints.
- The robot is scaled up from the measurements of the insect by a factor of 20.
- The robot has a much higher mass and a different mass distribution than the insect.
- The robot uses a computer rather than a biological neuronal network (higher signaling speed but lower parallel processing ability).

Due to these differences a stick insect - like walking controller cannot be used directly for the control of a robot like HECTOR. However, despite these differences, certain aspects of the walking controller of stick insects would be beneficial for the control of HECTOR. For example, the adaptability of the gait patterns, which has been observed in stick insects, cockroaches (Bender et al. 2011), flies (Wosnitza, Bockemühl, et al. 2013), and robots that are controlled by WALKNET, would be advantageous in the control of HECTOR. A key concept for these adaptive gait patterns is a relatively constant swing duration over a range of different walking speeds. As a consequence, during slow walking, the number of legs that are in contact with the ground is increased as compared to fast tripod walking. By trend, if more legs are in stance phase, the stability margin (see section 5.5.1) of the system will be increased and the power consumption will be decreased since the legs' weight does not need to be sustained. Thus, the adaptive gait pattern generation would be beneficial for a walking robot. WALKNET, however, was designed based on observations of walking stick insects which, as shown in chapter 3, do not permanently maintain static stability. Since the main focus during the development of WALKNET was the replication of insect gaits, maintenance of static stability was not included as requirement in the controller design. Additionally, WALKNET was developed based on the data of forward walking stick insects and is therefore only suitable for the realization of straight and slightly curved walking trajectories. To achieve omnidirectional walking, another requirement for HECTOR, the leg controller must be expanded.

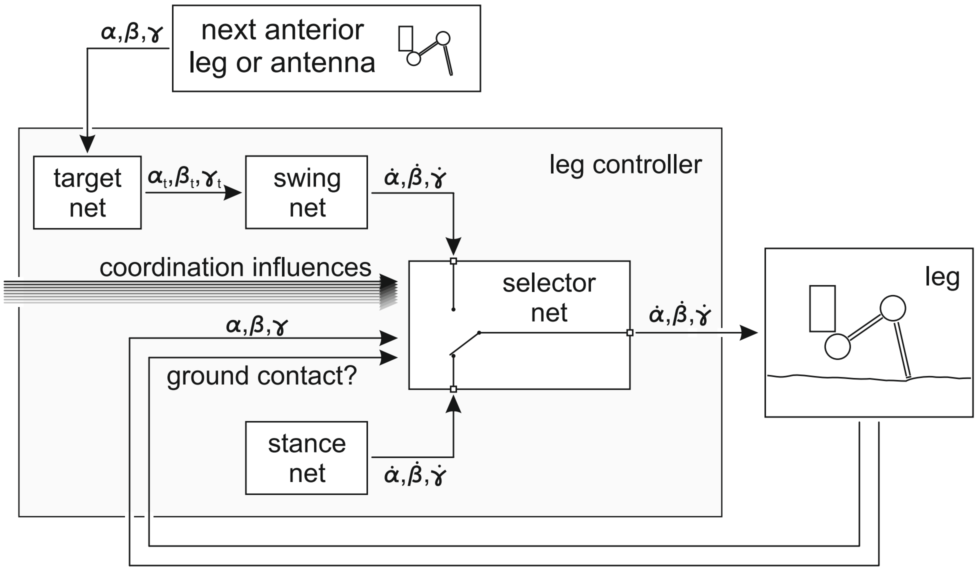

α,β,γ
next anterior leg or antenna
leg controller
target net
$\alpha_t,\beta_t,\gamma_t$
swing net
$\dot\alpha,\dot\beta,\dot\gamma$
coordination influences
selector net
$\dot\alpha,\dot\beta,\dot\gamma$
leg
α,β,γ
ground contact?
stance net
$\dot\alpha,\dot\beta,\dot\gamma$

Figure 5.1.: Simplified block diagram of a leg controller of the classical WALKNET (based on Cruse, Bläsing, et al. 2004). The controller consists of different sub-units that are usually realized using ANNs. As output, the leg controller sends angular velocities to the leg joints and uses the current joint angles as input together with ground contact information. Also, the current position of the next anterior leg or the contact position of an antenna is used to generate a target position for the swing movement.

Another advantage of WALKNET over many other walking controllers is the modular setup that allows for extensions of the basic functionality. In the classical concept of WALKNET, each leg controller consists of multiple, individual ANNs. In fig. 5.1, a simplified schematic of a leg controller is depicted. Each of the ANNs serves a specific function. The *stance net* defines the leg trajectory during stance phase, the *swing net* correspondingly controls the swing trajectory. The decision whether a leg should stance or swing is made by the *selector net*. Based on the current state of the leg (position, ground contact) either the output of the *stance net* or the *swing net* is passed to the leg, thus triggering the respective behavior. As mentioned in section 4.1, during swing phase the legs of stick insects target at the position of their anterior neighboring leg (Theunissen, Vikram, et al. 2014; Wosnitza, Engelen, et al. 2013) or the contact position of an antenna in case of the front legs (Schütz and Dürr 2011). In WALKNET the position of the anterior neighboring leg is processed by the *target net* and serves as input to the *swing net*. For the software design of the new walking controller a similar, modular setup was aspired to simplify the development of new control modules.

However, for clarity, the sub-modules of the leg controller have been implemented without the use of ANNs in the new walking controller. Therefore, in the following they will be named *stance module*, *swing module*, etc. Nevertheless, an ANN-based implementation of the modules can be implemented and used alternatively.

Since the functions of the individual modules interact on various levels they cannot be explained sequentially. For example, the stability module interacts with the stance and swing modules, which in turn also influence the stability module. Therefore, in the following section the concept of the controller will be summarized to give a rough overview of the functionality before the details are explained in the further sections of this chapter.

5.1. Overview of the Walking Controller

At the very core of the walking controller, the *stance* and *swing modules* serve a similar functionality as in the classical WALKNET: They provide the goal points of the stance and swing trajectories. Yet, they have been expanded to allow for omnidirectional walking. For example, the stance module can be used to estimate the future leg tip trajectories that are required for the selector module to decide whether to swing or stance. Also, the stance module is expanded by a method to compromise between the stance heights of legs. This is required to allow for walking in uneven terrain.

Instead of the classical selector net of WALKNET that only compares the x-position of the leg tip to the threshold given by the PEP to decide whether to start the swing phase, in the new *selector module* the workspace of each leg is designed as volume that is restricted in all directions. This enables the generalization of the leg coordination based on the coordination rules of WALKNET for omnidirectional walking. The definition of the leg workspaces is composed of multiple sources, e.g., the restrictions of the angular workspaces of the individual leg joints. The sources for these workspaces are named *unrestrictednesses*. Also, whereas in the original concept the position of the leg tip is used to decide whether to switch to swing, the new implementation uses the estimated time until the limit of the leg workspace is reached. For this purpose, the future leg trajectories are estimated by the stance module based on the current walking direction. Using these estimates it is checked whether the leg would exit its workspace within a time period that will be called predetermination period Δt_{pd}. If the leg tip would remain inside the workspace, the stance phase continues; if it would leave the workspace within the predetermination period, the swing phase is initiated.

To transfer the concept of the coordination rules of WALKNET to the new implementation, rather than the positions of the PEPs, now the durations of the legs' predetermination periods are modified due to the states of neighboring legs. Thus, the stance phases can be shortened or prolonged to coordinate the legs. The first four coordination rules (see section 4.1) are adapted to this new concept. To distinguish them from the classical implementation (*coordination rules*), they will be referred to as *coordination influences* in the following.

As mentioned before, stability maintenance is a prime goal of the controller. For this purpose the *stability module* interacts with the rest of the walking controller. Before a switch from stance to swing phase is performed, the stability module checks whether the leg is required to remain in stance phase to ensure stability. The stability module is also responsible to check in advance for instable postures or postures that might result in deadlocks. These deadlocks could occur due to special combinations of leg tip positions and might prohibit or at least delay further movement. Among others, these anticipated problems are countered by a reduction of the walking speed.

As a means to interface the technological parts of the controller with those that have been adopted from Walknet, technical coordination influences were formulated that may shorten the stance phases, thus triggering a switch to swing. For example, if a leg that is essential for the maintenance of stability (and therefore required to remain in stance phase) is approaching the limits of its workspace, neighboring legs are facilitated to perform a swing movement. By this measure, ideally, the neighboring legs will be able to sustain the robot after they regained ground contact at the end of swing, thus allowing the previously indispensable leg to swing.

The cooperation of these modules and the sequence of operation during each controller iteration will be explained in section 5.8.

5.2. The Selector Module and the Definition of the Leg Workspaces

In classical Walknet, as described, the selector net decides whether the leg is supposed to swing or to stance. The decision is based on the current position of the leg, the ground contact status and the states of the neighboring legs (transferred via the coordination rules). In general, the selector module switches from swing to stance phase when the leg reaches ground contact at the end of the swing phase. The stance phase is then continued until the leg crosses the PEP threshold (see fig. 4.2). Many implementations of Walknet compare only the x-coordinate of the leg's foot point to the PEP threshold (Schilling, Hoinville, et al. 2013): If the leg is in front of the PEP, the leg stays in stance phase. When the leg passes the PEP, the leg lifts for swing phase. Therefore, the PEP threshold can be thought of as a plane perpendicular to the x-direction. Due to the coordination rules, this plane is shifted forwards or backwards based on the state of the neighboring legs in order to shorten or prolong the stance phases.

If only the x-component of the leg tip position is considered in the swing/stance decision, obviously, only forward walking can be realized. If the leg is moved in the opposite direction, e.g., during backward walking, this approach fails as the leg will never cross the threshold. As a workaround, the use of two independent thresholds has been proposed: One threshold for forward walking and one threshold for backward walking (Schilling, Hoinville, et al. 2013). During curve walking, when the leg is moved outwards (Dürr and Ebeling 2005), this concept fails again since the leg could move sidewards infinitely without reaching any of the two thresholds. Concepts that overcome these problems partially have been discussed in section 4.1.1. For example, the approach presented by Burkamp (1996) uses an ANN to create a curved surface to represent the PEP. Still, this surface does not restrict the leg workspace in all directions, which makes omnidirectional walking impossible. A more complex ANN could probably be trained to create a closed threshold surface around the workspace of the leg. However, since the training data for this network must be produced externally anyway, the respective algorithm itself can be used directly to control the stance-swing-transition of the legs. Therefore, the first step in the conception of a new walking controller is to define the restrictions of a leg's workspace, which allows (and confines) omnidirectional movements. For this purpose, the unrestrictedness measure will be introduced.

5.2.1. Unrestrictedness Definitions

As described in section 1.2.4, Fielding (2002) introduced the concept of *restrictedness* to quantify the freedom of a leg to move in arbitrary directions based on multiple constraints. For example, the workspace of a leg might be confined due to the angular limits of the leg joints, singularities of the inverse kinematics, or the proximity of neighboring legs. Multiple of these constraints can be combined to one restrictedness, which defines the workspace of the leg. By definition, the closer the leg gets to the limits of its workspace, the higher will be the restrictedness. In this work, the *un*restrictedness rather than the restrictedness will be used as measure to identify the limits of the Cartesian workspace or other restricting influences. The advantage to use the unrestrictedness (ranging from 0 ≙ "restricted" to 1 ≙ "unrestricted") as opposed to the restrictedness approach (ranging from 0 ≙ "unrestricted" to 1 ≙ "restricted") is that the unrestrictedness-values from multiple sources (angular joint limits, proximity to singularity, etc.) can be easily multiplied in order to get a combined unrestrictedness: If one of the sources senses that the leg is restricted (returning a value of 0) the product of unrestrictednesses of all sources will be 0, thus reflecting the restrictedness of the leg. This is used in section 5.2.2 to define combined unrestrictednesses for swing and stance phases based on different sources. Unrestrictedness definitions that are used in this work are:

joint angle unrestrictedness Each of the leg joints has a minimal and maximal output angle due to mechanical restrictions that should not be exceeded. For example, the limits in the α-joints are set due to the outline of the housing (see section 2.4.2). A transgression of the α-limits would result in collisions between the body housing and the leg segments. However, if the joints are within their angular work range ($\chi_{\text{min}} < \chi < \chi_{\text{max}}$ for all $\chi \in \{\alpha, \beta, \gamma\}$), the leg can move in all directions without risk of collisions. This case is represented by the joint angle unrestrictedness $u_{\alpha\beta\gamma}$ by a positive, non-zero value:

$$u_{\alpha\beta\gamma} = \begin{cases} \prod_{\chi=\{\alpha,\beta,\gamma\}} \frac{4(\chi_{\text{max}}-\chi)(\chi-\chi_{\text{min}})}{(\chi_{\text{max}}-\chi_{\text{min}})^2} & \text{if all } \{\chi_{\text{min}} < \chi < \chi_{\text{max}}\}_{\chi=\{\alpha,\beta,\gamma\}} \\ 0 & \text{otherwise} \quad . \end{cases} \tag{5.1}$$

If the joints are exactly at the middle position of their angular working ranges ($\chi = (\chi_{\text{min}} + \chi_{\text{max}})/2$ for all $\chi \in \{\alpha, \beta, \gamma\}$), the function returns an unrestrictedness of 1, thus indicating freedom to move the leg in all directions. The closer the joint angles get to the limits of their respective working ranges, the smaller will be the output of $u_{\alpha\beta\gamma}$. For angular positions at or even beyond the limits, the output is 0, which indicates the

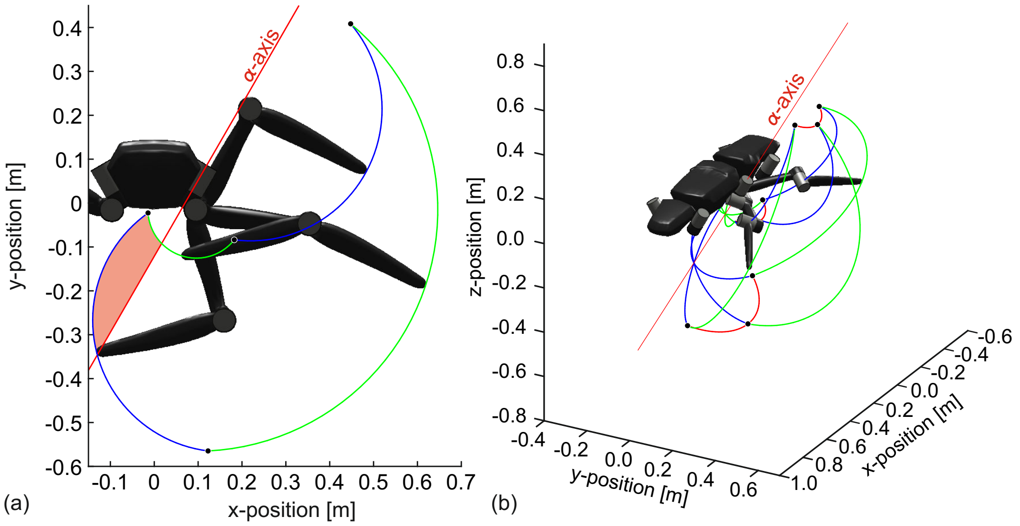


Figure 5.2.: Leg workspace due to angular joint limits. (a) illustrates the angular workspace due to the output angle limits of the β- and γ-drives for the left middle leg (view from front). The green and blue lines show the outline of the workspace. Green lines represent those segments of the outline that result from a movement of the β-drive with the γ-drive at its minimum or maximum output angle. Correspondingly, the green lines represent segments for which the γ-drive moves whereas the β-drive is at one of its limits. The axis of the α-drive is shown in red. Due to the singularity along this axis the workspace highlighted in red is not used during walking. The leg posture of the left middle leg is shown as overlay in four different postures along the workspace outline. (b) shows the outline of the angular workspace from a slanted perspective. As in (a), the blue and green lines show the outline of the angular workspace for β- and γ-drives at their limits, respectively. In addition, the red lines show the leg tip positions that are obtained if both, β- and γ-drive, are at their workspace limits and the α-drive is rotated. For reference, the robot body is shown with the left middle leg in two stretched postures.

restrictedness of the leg. See fig. 5.5 (a) for the resulting workspace, table 2.3 for the joint angle limits and fig. A.1 for the angle definition within the leg kinematics.

singularity unrestrictedness The leg kinematics has a singularity along the α-axis (see section A.4). Therefore, for leg tip positions along this axis, the α-angle cannot be determined uniquely by the inverse kinematics. Also the vicinity of the singularity should be avoided since a minimal change of the leg tip position might require big changes of the α-angle. Therefore, the shortest distance d_{xy} between the α-axis and the leg tip should not fall below $d_{\min}$ (see fig. 5.3). The singularity unrestrictedness u_{s} is computed

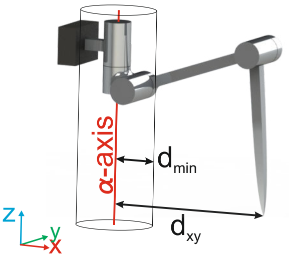


Figure 5.3.: Depiction of the singularity unrestrictedness. Due to the leg kinematics a singularity exists along the α-axis. To prevent problematic leg configurations the leg tip is supposed to maintain a minimal distance d_{min} from the singularity.

by

$$u_{\text{s}} = \begin{cases} \frac{d_{\text{xy}} - d_{\text{min}}}{(l_{\text{cox}} + l_{\text{fem}} + l_{\text{tib}}) - d_{\text{min}}} & \text{if } d > d_{\text{min}} \\ 0 & \text{if } d \leq d_{\text{min}} \end{cases} \tag{5.2}$$

with

$$d_{\text{xy}} = \sqrt{(x_\psi^2 + y_\psi^2)} \quad \text{and} \tag{5.3}$$

$$\begin{pmatrix} \boldsymbol{p}_\psi \\ 1 \end{pmatrix} = \begin{pmatrix} x_\psi \\ y_\psi \\ z_\psi \\ 1 \end{pmatrix} = {}^\psi\boldsymbol{T}_\alpha \cdot {}^\alpha\boldsymbol{T}_\beta \cdot {}^\beta\boldsymbol{T}_\gamma \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad . \tag{5.4}$$

The homogeneous transformation matrices ${}^\psi\boldsymbol{T}_\alpha$, ${}^\alpha\boldsymbol{T}_\beta$, and ${}^\beta\boldsymbol{T}_\gamma$ are used to transform the position of the leg tip from the F_γ coordinate system, which is located at the leg tip, to the F_ψ coordinate system, which is located at the leg onset (see fig. A.1). Since the z-axis of the F_ψ coordinate system is aligned with the rotational axis of the α-joint, the shortest distance d_{xy} between the leg tip and the α-axis can be computed based on the x- and y-components of $\boldsymbol{p}_\psi$.

To normalize the unrestrictedness the distance between the leg tip and the singularity region (defined by the radius d_{min} around the α-axis) is divided by the maximal length of the leg minus the radius of the singularity region. Thus, a maximally outstretched leg will result in $u_s = 1$. The closer the leg gets to the singularity region, the lower will be the output of u_s. At a distance of d_{min} or below, u_s yields 0. For the resulting workspace of the left middle leg see fig. 5.5 (b).

torque unrestrictedness The maximum torque each joint is able to produce depends mainly on the selection of motor and gearbox. For the setup described in section 2.3, a maximum torque of $\tau_{\text{max}} = 18\,\text{Nm}$ is assumed.[2] The joint torques that are required for the exertion of a given force antiparallel to the gravity vector depends on the posture of the leg (see section A.3). In fig. 5.4, the torques for the left middle leg's three joint drives are depicted for the exertion of forces at the leg tip in different directions. To ensure that each leg can support at least half the weight of the body in any given situation, the required joint torques are computed for a test force of 65 N at the leg tip. As long as the required joint torques are below the defined maximum torque of the drives, the torque unrestrictedness u_τ will return positive results:

$$u_\tau = \begin{cases} \prod_{\chi=\{\alpha,\beta,\gamma\}} \left(1 - \left|\frac{\tau_\chi}{\tau_{\chi,max}}\right|\right) & \text{if all } \{|\tau_\chi| < \tau_{\chi,max}\}_{\chi=\{\alpha,\beta,\gamma\}} \\ 0 & \text{otherwise} \quad . \end{cases} \tag{5.5}$$

The resultant workspaces for different orientations of the robot are depicted in fig. 5.5 (c,e).

leg tip distance unrestrictedness To prohibit collisions between ipsilaterally neighboring legs a minimum distance d_{min} must be maintained between their leg tips. This is especially relevant during swing to avoid treading on neighboring legs. Thus, an unrestrictedness is defined that considers the postures of a pair of ipsilaterally neighboring legs:

$$u_{\text{tip}} = \begin{cases} \frac{(x_{\text{tar,a}} - x_{\text{tar,p}}) - d_{\text{min}}}{\left\|\boldsymbol{O}_{\text{LCS,a}} - \boldsymbol{O}_{\text{LCS,p}}\right\| + 2l_{\text{max}} - d_{\text{min}}} & \text{if } x_{\text{tar,a}} - x_{\text{tar,p}} > d_{\text{min}} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

with

$$l_{\text{max}} = (l_{\text{cox}} + l_{\text{fem}} + l_{\text{tib}}) \tag{5.7}$$

with index a denoting the anterior and p denoting the posterior leg of the ipsilateral leg pair. $\mathbf{O}_{\text{LCS,a/p}}$ are the positions of the leg onsets and $x_{\text{tar,a/p}}$ are the x-components of the tarsus positions relative to the RCS (Robot Coordinate System), which is located between the onsets of the hind legs. u_{tip} yields 1 if the legs are stretched in opposite directions. The closer the leg tips get, the lower will be the output of u_{tip}. At or below a distance of d_{min}, u_{tip} yields 0. The resulting leg distance unrestrictedness is depicted in fig. 5.6.

[2] Even assuming a 100% efficiency of the gearbox, the maximum continuous torque of the actuator would be limited to 8.34 Nm (*continuous torque of the motor* × *gearbox ratio*). However, even during tripod walking, e.g., the β-drives of the middle legs must support the body only half the time and due to the slanted leg setup, the load for the β-drive is partially reduced during the stance movement. Thus, a higher maximum torque can be assumed for the workspace definition as long as the average power of the motor is not exceeded. During extensive walking experiments with HECTOR (see chapter 7), the specified constraint of 18 Nm proved to be adequate since the motors did not overheat.

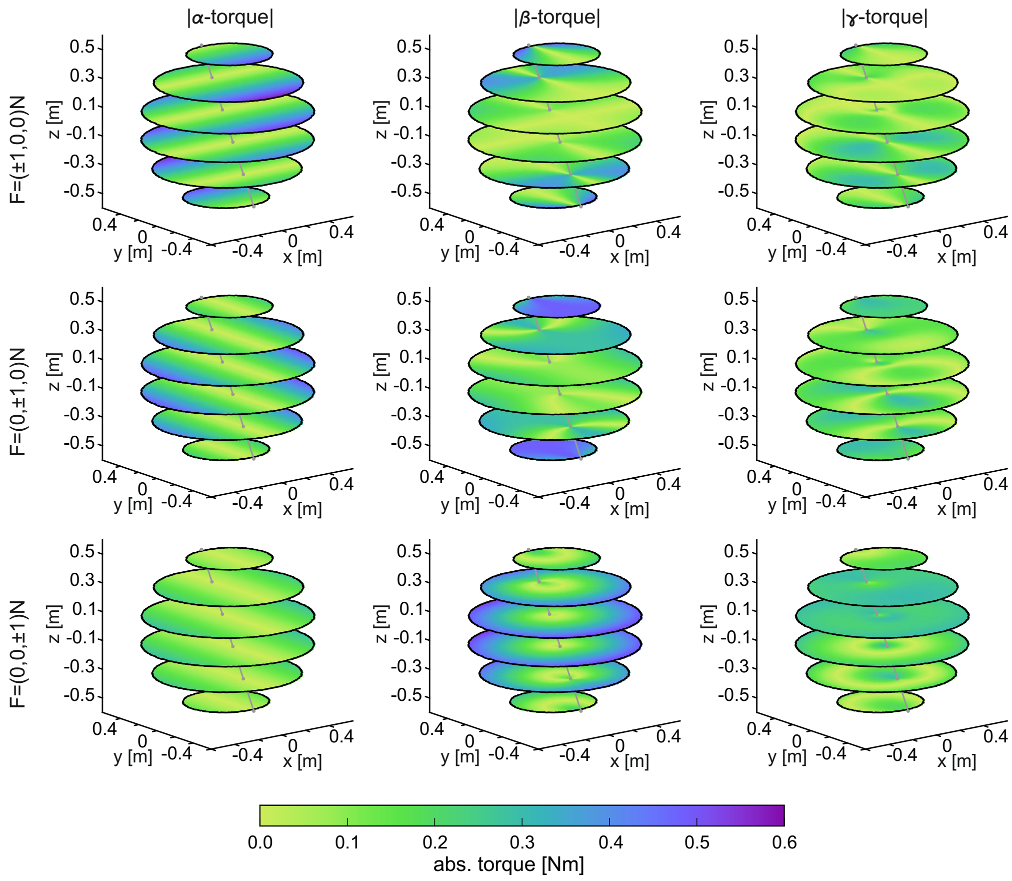


Figure 5.4.: Visualization of the absolute torques needed to create an end effector force of 1 N for the middle left leg. The three rows show the required torques to produce the force in x-, y-, and z-direction. The three columns show the torques for the α-, β-, and γ-joint. The α-axis is indicated by a slanted, gray line in all plots.

leg segment distance unrestrictedness In addition to the rather simple distance check of *leg tip distance unrestrictedness* that considers only the distance in x-direction between the tips of legs, another distance check was introduced to prohibit collisions between the segments of adjoining legs. For this purpose, the distances between all combinations of segments of two legs are computed.[3] For this calculation the leg segments are simplified as geometric primitives (see fig. 5.7). Both, femur and tibia are represented by capsules (cylinders with hemispherical ends) with suitable radii, r_{fem} and r_{tib}, which completely

[3] Collisions between β-joint drives of contralateral legs are not possible due to the angular limitation of the α-joints. Also, collisions between leg segments and the body housing or the housings of the α- or β-joint drives are prohibited by the angular joint limits. As a consequence, α- and β-joint drives are neglected during the distance calculation. For different robotic setups or joint angle limits, however, the inclusion of both drives might be required.

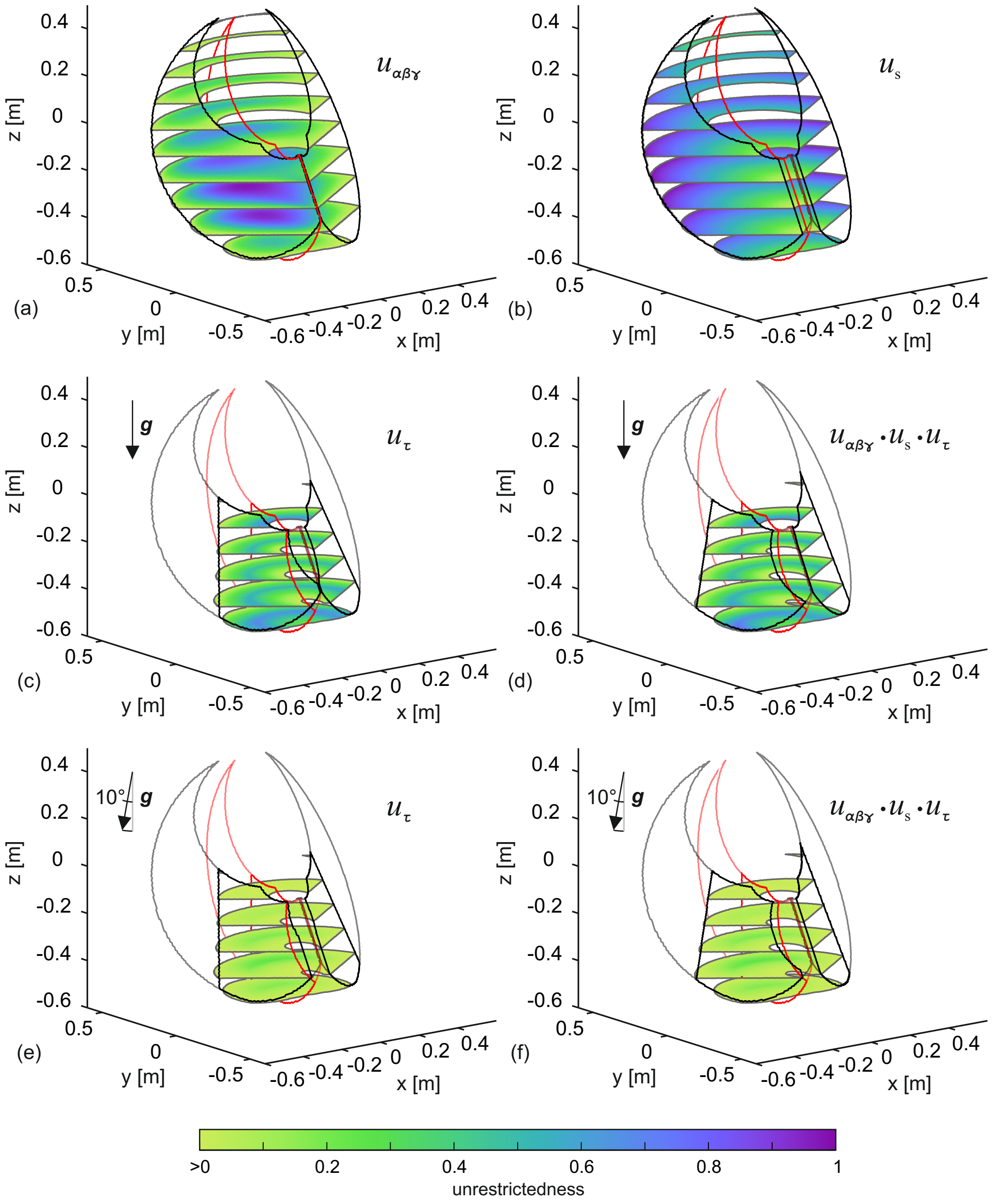


Figure 5.5.: Volumetric representations of different unrestrictednesses for the middle left leg within the workspace of the leg joints ($u_{\alpha\beta\gamma}$, shown in (a)). The slice planes are set at distances of 0.1 m in z-direction. The color denotes the unrestrictedness at the respective leg tip positions (only for positive, on-zero unrestrictedness). Therefore, the workspace of the leg is defined by the colored areas. In addition the contours of the unrestrictednesses are drawn as lines on the leg plane for α-angles of $\pm$ 1 rad ($\sim$57°) and at the neutral position with $\alpha = 0$ (in red; see fig. A.2 (g),(h) for leg orientations). For easier comparability of the workspaces, the contours of $u_{\alpha\beta\gamma}$ have been transferred to the other unrestrictednesses shown in (b)–(f) as semi-transparent lines. (a) joint angle unrestrictedness, (b) singularity unrestrictedness, (c) torque unrestrictedness for vertically directed vector of gravity, (d) combined unrestrictedness from (a-c), (e) torque unrestrictedness for walking on inclination with an angle of 10°, (f) combined unrestrictedness from (a, b, e).

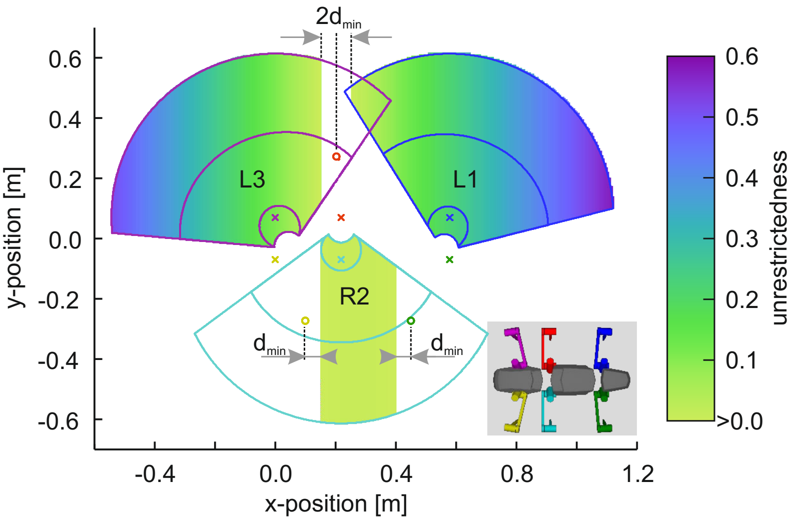


Figure 5.6.: Leg distance unrestrictedness for left front, right middle, and left hind legs based on the positions of the other legs. The origins of the () are marked by crosses in the colors denoted by the leg colors in the inset. The circles of corresponding color indicate the positions of the right front, left middle and right hind legs. Based on the positions of these legs the leg distance unrestrictednesses are plotted as color gradient (within the intersection of workspaces defined by the angular and singularity unrestrictednesses of the legs; outer contour). Additionally, the contours of the workspaces considering also the torque unrestrictedness (inner contour) are plotted.

envelop the rather complex geometries of the leg segments. The γ-joint drive is represented as sphere with radius r_γ since the geometrically more suitable representation as cylinder would be considerably more complex during the calculation of distances. For the definition of the corresponding unrestrictedness, the minimum distance between any two segments d_{seg} is used:

$$u_{\text{seg}} = \begin{cases} \frac{d_{\text{seg}} - d_{\text{min}}}{\left\|\boldsymbol{O}_{\text{LCS,a}} - \boldsymbol{O}_{\text{LCS,p}}\right\| - 2l_{\text{cox}} - 2r_{\text{fem}} - d_{\text{min}}} & \text{if } d_{\text{seg}} > d_{\text{min}} \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

(5.9)

As minimal allowed distance between the segments, d_{min} is used. For a posture, in which the legs are stretched in opposite directions, the minimal distance between them is spanned by the capsules that represent the femora (since the β-drives and the coxae are not considered in the distance calculation). In this case the unrestrictedness will evaluate to 1. For the case in which the distance between any two segments is reduced to d_{min} the function yields 0. For regular forward walking the *leg tip distance unrestrictedness*

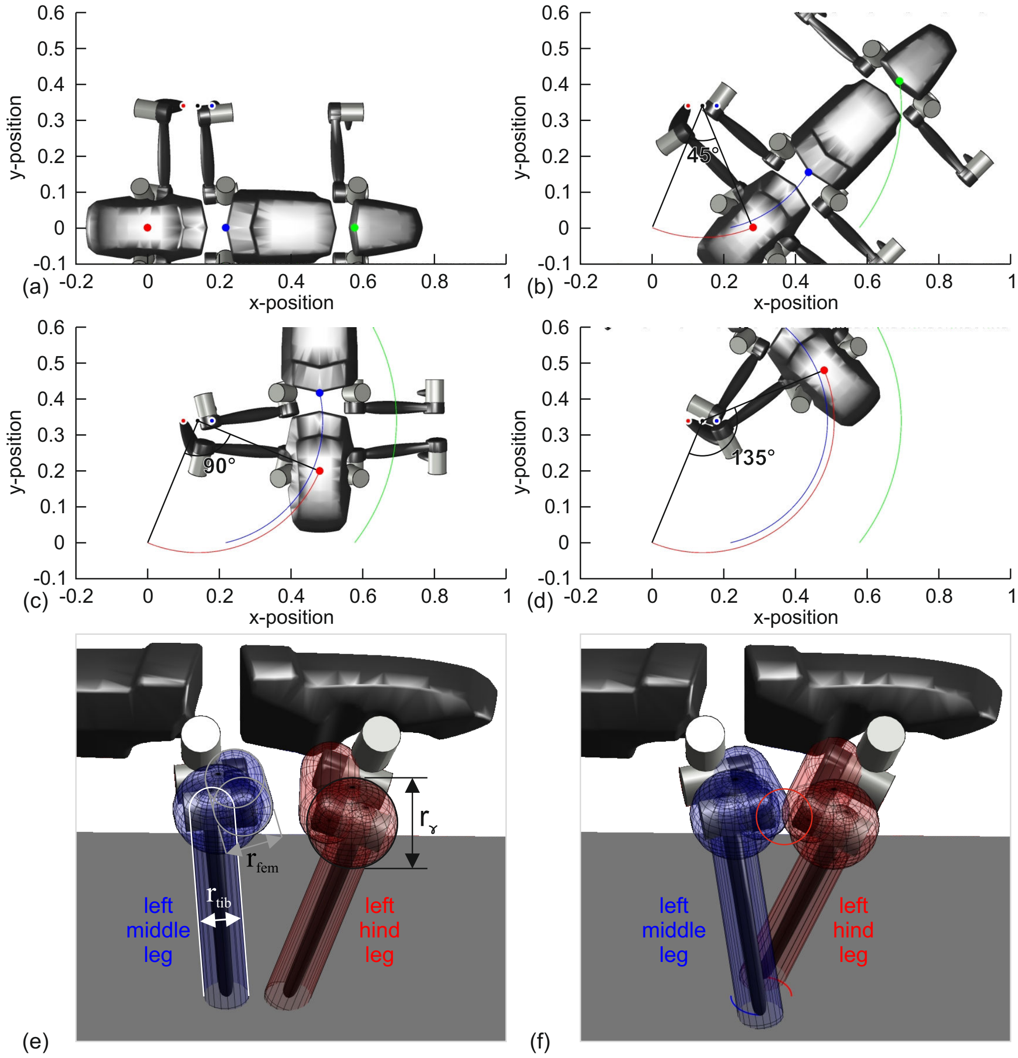


Figure 5.7.: Illustration of a movement that causes collisions between segments of adjoining legs. (a)–(d) show the movement of the robot on a circular trajectory at angles of 0°, 45°, 90°, and 135°, respectively. The center of rotation is marked by a black dot. Since the tips of the left middle and hind legs (marked by red and blue dots) are located close to the center of rotation they could stay in stance phase indefinitely without reaching the limits of their stance workspaces. Therefore, the swing phase is not initialized. In (d), at an angle of 135°the housings of the γ-drives collide with each other. To circumvent cases like these the distances between all segments of adjoining legs are computed. For this purpose the leg segments are represented by spheres and capsules that envelop the more complex leg segment housings. These geometric primitives are depicted as semi-transparent overlays in (e) and (f). (e) shows the situation that is already depicted in (a) for a turning angle of 0° from a slanted perspective. In (f), the posture of (c) is shown again from a slanted perspective. By checking the distance between the enveloping, geometric primitives, the collision between the leg segment housings is already forecasted at a rotational angle of 90°. The point of collision of the γ-spheres is marked by a red circle.

is sufficient to prohibit collisions. However, especially during curve walking with tight turning radii, collisions might occur between the segments of the internal legs if the leg tips are positioned close to the center of the turning circle. This scenario is depicted in fig. 5.7.

stability unrestrictedness In section 5.5 the concept of the stability module will be explained in detail. However, the core of the stability module is the definition of the ESM (Energy Stability Margin) by Messuri (1985). The ESM corresponds to the minimum energy that is required to let the robot tilt about any of the lines of its support polygon. In fig. 5.8, this is depicted for turning the COM from its original position at $\boldsymbol{p}_{\mathrm{COM}}$ to the statically instable position $\boldsymbol{p}'_{\mathrm{COM}}$. Postures that are potentially instable are prevented by the stability module (see section 5.6), e.g., by reduction of the walking speed or inhibition of stance-swing transitions. This, however, can result in deadlocks: If a leg is required for the maintenance of stability although it has almost reached the end of its workspace, the switch to swing phase is prevented by the stability module (see section 5.5.2). To prohibit the leg to leave its workspace the stance speed of the robot is reduced or even stopped (see section 5.6). After the robot comes to a halt none of the still unrestricted legs will be triggered to start its swing phase. Thus, the problematic posture cannot be resolved automatically.[4]

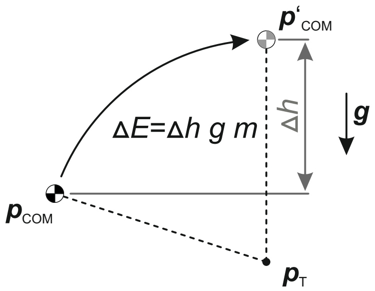


Figure 5.8.: Simplified depiction of the ESM. The COM of the robot is virtually rotated about the point $\boldsymbol{p}_T$ (located on the connecting line between two foot points) from its original position at $\boldsymbol{p}_{\mathrm{COM}}$ towards the statically instable position $\boldsymbol{p}'_{\mathrm{COM}}$. The ESM corresponds to the difference of potential energy between the two postures. For a detailed depiction see fig. 5.24.

During forward walking, this happens usually if the front legs are reaching the posterior end of their workspace with the middle legs having passed the x-position of the COM (see fig. 5.27). In this situation, both front legs are required to stay in stance phase to maintain stability. The forward movement must be stopped since the front legs are about to reach the limits of their workspaces. To prohibit this scenario, it must be ensured

[4] In section 5.7, the *deadlock-resolving influence* will be presented, which is meant to resolve these situations. However, the stability unrestrictedness that is presented here, is meant to prevent the occurrence of these situations in advance. Therefore, it serves to obviate the reduction of the stance speed.

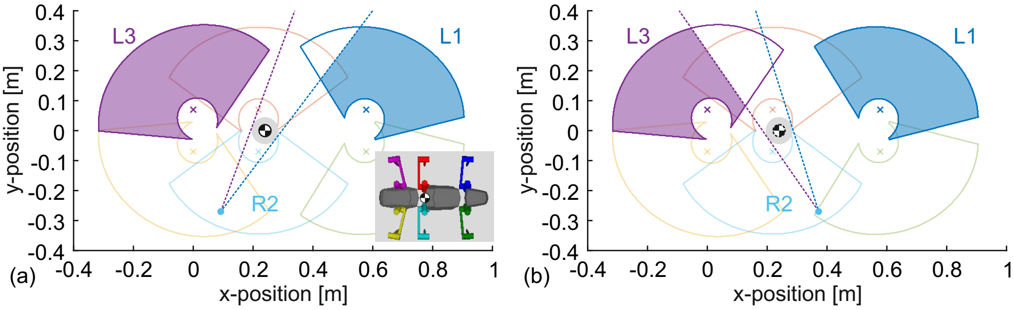


Figure 5.9.: Exemplary depiction of the stability unrestrictedness. The contours of the stance phase workspaces are drawn for the six legs in the colors indicated in the inset. The figure shows the additional constraints that must be considered for the positioning of the left front and hind leg based on the position of the right middle leg. The limitations are shown for two different positions of the right middle leg (indicated by circles). The tip of the left hind leg must be located behind the dashed/dotted magenta lines—depending on the position of the middle leg. The tip of the left front leg must be located in front of the blue lines, again depending on the position of the middle leg. The distance of the blue and magenta lines to the COM is defined by the minimal energy threshold ΔE_{min}.

that at least one of the front and middle leg combinations (left front and right middle leg or right front and left middle leg) is sufficient to maintain the stability together with the hind legs. For this purpose, the ESM is computed for both diagonal front and middle leg combinations separately. The energies $\Delta E_{i,j}$ that would be required to tilt the COM around the lines that are spanned by the leg tips of the leg i and j. The stability unrestrictedness is defined by:

$$u_{\mathrm{stab}} = \begin{cases} \frac{\Delta E_{i,j} - \Delta E_{\mathrm{min}}}{\|\boldsymbol{g}\|\, m\,(d_{\mathrm{CL}} + l_{\mathrm{cox}} + l_{\mathrm{fem}} + l_{\mathrm{tib}}) - \Delta E_{\mathrm{min}}} & \text{if } \Delta E_{i,j} > \Delta E_{\mathrm{min}} \\ 0 & \text{if } \Delta E_{i,j} \leq \Delta E_{\mathrm{min}} \end{cases} \tag{5.10}$$

with

$$\Delta E_{i,j} = -(\boldsymbol{p}'_{\mathrm{COM}} - \boldsymbol{p}_{\mathrm{COM}}) \cdot \boldsymbol{g}\, m \quad \text{and} \tag{5.11}$$

$$d_{\mathrm{CL}} = \max\left(\{\|\boldsymbol{O}_{\mathrm{LCS,n}} - \boldsymbol{O}_{\mathrm{COM}}\|\}_{n=\{i,j\}}\right) \tag{5.12}$$

$\Delta E_{i,j}$ is the minimal energy that is required to lift the center of gravity such that it would tumble around the line spanned by the tips of legs i and j. $\boldsymbol{g}$ is the vector of gravity, $\boldsymbol{p}_{\mathrm{COM}}$ is the current position of the center of gravity and $\boldsymbol{p}'_{\mathrm{COM}}$ is the virtual position of the center of gravity at maximal potential energy during the tilting movement. ΔE_{min} is a threshold that defines the minimal allowed stability margin. d_{CL} is the maximal distance between the COM and the leg onsets. The most stable posture would be achieved if the COM would hang vertically below the foothold of one leg. For this situation, u_{stab} would yield 1. For the other extreme posture with the COM directly above the respective support polygon line, u_{stab} returns 0. In fig. 5.9, the workspaces of the left front (relevant for forward walking) and left hind legs (relevant for backward walking) are shown for two different positions of the right middle leg.

5.2.2. Swing-, Stance-, and Swing-Target-Unrestrictedness

Due to the presented unrestrictednesses, the workspaces of the legs are defined. However, not all unrestrictedness components must be respected during swing and stance phase. For example, during stance phase, the distance between the tips of adjacent legs cannot be influenced since the legs must remain at fixed positions relative to each other. Therefore, the *leg tip distance unrestrictedness* is not considered while stancing. Correspondingly, separate combinations of the unrestrictednesses are used during swing and stance phases and for the selection of a target for the swing trajectory.

stance unrestrictedness During stance phase only the combination of joint angle, singularity, torque, and leg segment distance unrestrictednesses are taken into account to plan leg movements. Also for the determination of the stance-swing-transition, only these unrestrictednesses are used. The stance unrestrictedness is therefore defined by:

$$u_{\text{st}} = u_{\alpha\beta\gamma} \cdot u_s \cdot u_\mathcal{T} \cdot u_{\text{seg}} \tag{5.13}$$

Thus, the stance workspace is defined as $W_{\text{st}} = \{(x, y, z) | u_{\text{st}} > 0\}$.

swing unrestrictedness In swing phase, only the joint angle, singularity, and leg segment distance unrestrictednesses must be considered. The torque restriction of the drives typically does not affect the leg during swing since only the γ-joint drive must be lifted. The resulting swing workspace is $W_{\text{sw}} = \{(x, y, z) | u_{\text{sw}} > 0\}$ with

$$u_{\text{sw}} = u_{\alpha\beta\gamma} \cdot u_s \cdot u_{\text{seg}} \quad . \tag{5.14}$$

swing-target unrestrictedness The stance unrestrictedness includes all unrestrictednesses that are considered during swing phase ($W_{\text{sw}} \subseteq W_{\text{st}}$). Therefore, as long as the leg stays within the stance workspace, a switch to swing is possible at any time. However, while swinging, the leg might leave the stance workspace and thus a switch to stance might not be allowed. As depicted in fig. 5.1 for the classical leg controller, a module corresponding to the *target net* is needed that selects a proper position $\boldsymbol{p} \in (W_{\text{st}} \cap W_{\text{sw}})$ for a switch from swing to stance. Beside the stance unrestrictedness also the *leg tip distance* and the *stability unrestrictednesses* are included for the search of a suitable swing target.

The leg tip distance unrestrictedness acts between ipsilaterally neighboring legs and ensures that the swing target has a sufficiently large distance from foot positions of other legs. As will be shown in fig. 5.15, during fast forward walking the swing target is positioned at the anterior limit of the swing-target unrestrictedness. Since the corresponding swing-target workspace is constrained by the leg tip positions of neighboring legs, the target position will be close to the leg tip position of the anterior leg. This influence is therefore equivalent to the classical coordination rule 4, which postulated that (during forward walking) the swing target is located close to the foot position of the next anterior leg. Whereas the classical coordination rule 4 was meant as attractive function (see section 4.1), the unrestrictedness-based implementation acts as repulsive mechanism. During backward walking, swinging legs will target at the leg tip position of the posterior leg. Since the workspace definition prohibits the possibility for a leg to tread on a neighboring leg, rule 6 (treading-on-tarsus reflex; see fig. 4.1) does not need to be implemented.

The *stability unrestrictedness* is applied between the diagonal front and middle legs, and the diagonal middle and hind legs (L1↔R2, R1↔L2, L2↔R3, R2↔L3). The energy that would be required to turn the COM of the robot around the axis spanned by two of these foot points must always be bigger than $S_{\mathrm{ESM,min}}$. This ensures that pairs of diagonal legs can always support the center of gravity. For example, the right middle and left hind leg will always be positioned such that the right hind leg could switch to swing phase at any time.

The swing-target unrestrictedness is therefore

$$u_{\mathrm{swt}} = u_{\mathrm{st}} \cdot u_{\mathrm{tip}} \cdot u_{\mathrm{stab}} \quad . \tag{5.15}$$

The workspace, within which a target for the swing movement can be selected is defined by $W_{\mathrm{swt}} = \{(x, y, z) | u_{\mathrm{swt}} > 0\}$.

5.2.3. Unifying Unrestrictedness and Inter-Leg Coordination

In the original concept of Walknet, during the stance phase the current foot position of each leg is compared to the PEP-threshold to decide when to liftoff. If the leg is in front of the threshold it remains in stance phase. If it crosses the threshold the leg is lifted for swing. The PEP-threshold gets shifted forwards to shorten the stance phase or it is shifted backwards to prolong the stance phase in dependence of the coordination rule influences from other legs. Within the new unrestrictedness-based walking controller, in order to decide when to start the swing phase, a direct transfer of this concept could be to compare the current value of the stance unrestrictedness to a predefined threshold. The leg would switch to swing phase if the unrestrictedness drops below the threshold. Disregarding the exponential functions used for the definition of the restrictions, this corresponds to the concept proposed by Fielding and Dunlop (2004).

To incorporate the coordination rules, either the limits for the restrictions (e.g., the angular joint limits) or the threshold could be modified. Both approaches would yield similar results. Considering only the joint angle unrestrictedness, its limits can be imagined as a closed surface that envelops the permissible workspace of the leg. By modifying the joint angle limits, the workspace could be expanded or shrunk. However, since the transformation from joint angle space to Cartesian space is non-linear, this might not

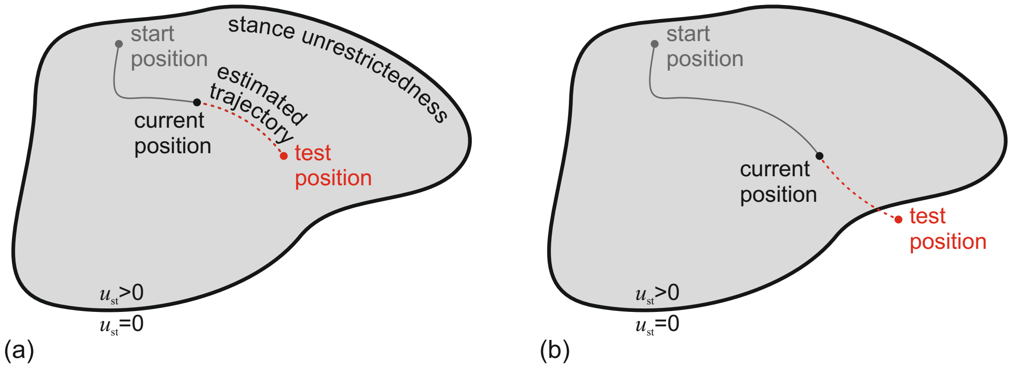


Figure 5.10.: Depiction of the unrestrictedness predetermination by the selector module during stance phase. (a) shows the course of the previous trajectory from the *start position* to the *current position* in gray. Assuming a constant walking direction, the future trajectory (red, dashed line) is estimated for the duration of the predetermination period. Since the *estimated trajectory* remains within the stance unrestrictedness (*test position* within stance unrestrictedness) the stance phase will be continued. (b) shows the stance movement at a later point in time (the current position has changed). Again, the future trajectory is estimated (red, dashed line). In this example the *estimated trajectory* would leave the stance unrestrictedness within the predetermination period (*test position* outside stance unrestrictedness). Therefore, the leg controller will switch to swing phase.

give the desired results: A small change of position close to the α-axis singularity will result in a much bigger change of unrestrictedness than the same movement at a more distal position. Also, for each unrestrictedness the function used to scale the limits in dependence of the coordination influences would need to be fine-tuned.

Therefore, to unify the different unrestrictednesses a conceptually easier approach is to compute the time until the limit of the workspace as defined by the stance unrestrictedness is reached. Since the future trajectory is not known a priori it must be assumed that the current movement will be continued (see fig. 5.10 (a); Espenschied and Quinn 1994; Görner and Hirzinger 2010). The remaining time can be compared to a time threshold just like the x-position was compared to a Cartesian threshold in the original concept of WALKNET. If the remaining time drops below this threshold the leg is supposed to switch to swing phase. To combine this concept with the coordination influences, the fixed time threshold Δt_{thr} can be increased or decreased by multiplying it with the outputs I of the coordination influences, thus shortening or prolonging the stance phase. The resulting threshold $\Delta t_{\text{pd}} = \Delta t_{\text{thr}} \cdot I_1 \cdot I_2 \cdot I_3$ is named *predetermination period*.[5]

[5] Note that multiple influences of the same type may act on a leg controller, e.g., influence 2 from the contralateral and one or two ipsilateral legs.

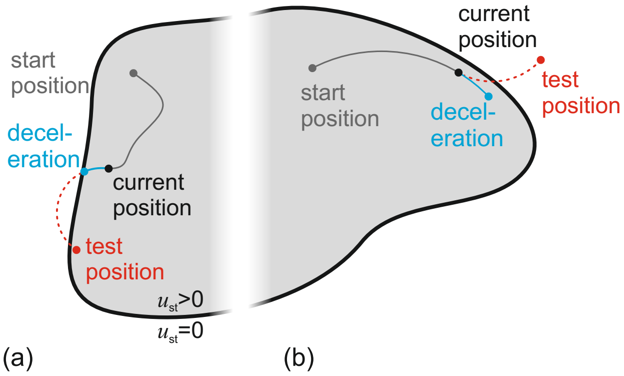


Figure 5.11.: Depiction of two rare situations that nevertheless need to be addressed in the control of the robot. (a) shows a case, in which the estimated stance trajectory leaves the stance unrestrictedness and enters it again. Since the *test position* is located within the unrestrictedness the stance phase would continue. However, for a shorter period, the estimated future trajectory is checked for each iteration (blue line). This allows for a gradual deceleration. (b) shows the case of a leg trajectory that is parallel to the limit of the unrestrictedness. Due to a sudden change of walking direction the *test position* lies outside the unrestrictedness. If the remaining time until the leg would leave its workspace is not sufficient for the deceleration, the previous walking direction is kept and the robot is decelerated (blue line).

To simplify the computation, it is sufficient to only check the unrestrictedness at the expected future position at *current time+predetermination period*. Therefore, if the leg would leave the workspace within the duration given by the *predetermination time* (assuming no change in walking direction), it should immediately switch to swing phase. If it would remain inside the workspace the stance phase may continue.

Since only a single future position is checked this involves the risk that the virtual extension of the trajectory leaves and reenters the workspace within this period (see fig. 5.11 (a)). Therefore, checking only a single future position, the leg would stay in stance although it ought to start its swing phase (conceptually, only the period until the leg leaves the workspace for the first time is relevant). Situations like these might also arise due to a change of walking direction: If the foot point moves at short distance parallel to the limit of the workspace, the time until it is expected to pass over the limit might be quite long. However, if the walking direction changes such that the foot point moves directly towards the workspace limit, it might reach it within the next iteration (see fig. 5.11 (b)). Although both scenarios occur rarely the controller must be able to cope with them, e.g., by an abrupt stop of the leg and body movements.

Due to the elasticity in the joints of the robot, a sudden stop does not harm its mechanics. Still, a controlled deceleration would be preferable. Based on the current velocity and the admissible rate of deceleration, the duration of the minimum deceleration period $\Delta t_{\text{min,dec}}$ can be defined. Thus, it is sufficient to make sure that the extension of the trajectory will not leave the workspace within this period. If the trajectory would reach the limit of the workspace at the end of the period, either the swing phase or—if this is not possible—the deceleration must be initiated (see fig. 5.11 (a)). If a change of direction would cause the trajectory to leave the defined workspace at an even shorter period the previous trajectory will be followed during the deceleration phase if a switch to swing phase is not possible (see fig. 5.11 (b)).

coordination influences To distinguish the coordinating mechanisms used in the new framework from the classical *coordination rules*, they will be named *coordination influences*. Following the classical coordination rules listed in section 4.1, the coordination influences used in conjunction with the unrestrictedness-approach will be numbered according to their respective classical equivalent (see also fig. 5.12):

- Influence 1 reduces the temporal threshold Δt_{pd} of the receiver leg by a factor f_1 if the sender leg is in swing phase (see fig. 5.12 (b)).
- Influence 2 increases the temporal threshold Δt_{pd} of the receiver leg by a factor f_2 for a short time after the sender leg touched down at the end of its swing phase (see fig. 5.12 (c)).
- Influence 3 increases the temporal threshold Δt_{pd} of the receiver leg by f_3 for a short time when the sender leg is as far away from the workspace limit as a swing phase would take (Δt_{r3}). Thus, the receiver leg should be able to finish its swing phase before the sender leg reaches its workspace limit (see fig. 5.12 (d)).

Whereas the original coordination rules 1 and 2 use de- and activation delays (see section 4.1), these are omitted in the new implementation. As reasons for the necessity of these delays the low conduction velocity of the insect nerves and the deferred ground contact detection due to the kinematic analysis have been indicated. However, since the new controller is explicitly designed for the application on a robotic system that uses internal sensors for ground contact detection both of these reasons do not apply.

For the development of WALKNET the directions of influence for the coordination rules were determined based on behavioral experiments on stick insects (see fig. 4.1). Usually, the quality of the deduced models was evaluated based on the similarity of the leg coordination to that of forward walking stick insects. For the application on a robotic system, however, other measures of walking quality might apply (reduction of energy consumption, body oscillations, etc.). Therefore, the directions of influence as found for stick insects do not necessarily need to be adapted for the control of a robot. However, to show the ability of this approach to reproduce the bioinspired gait patterns, the directions of influence will be retained during the walking optimizations presented in chapter 6.

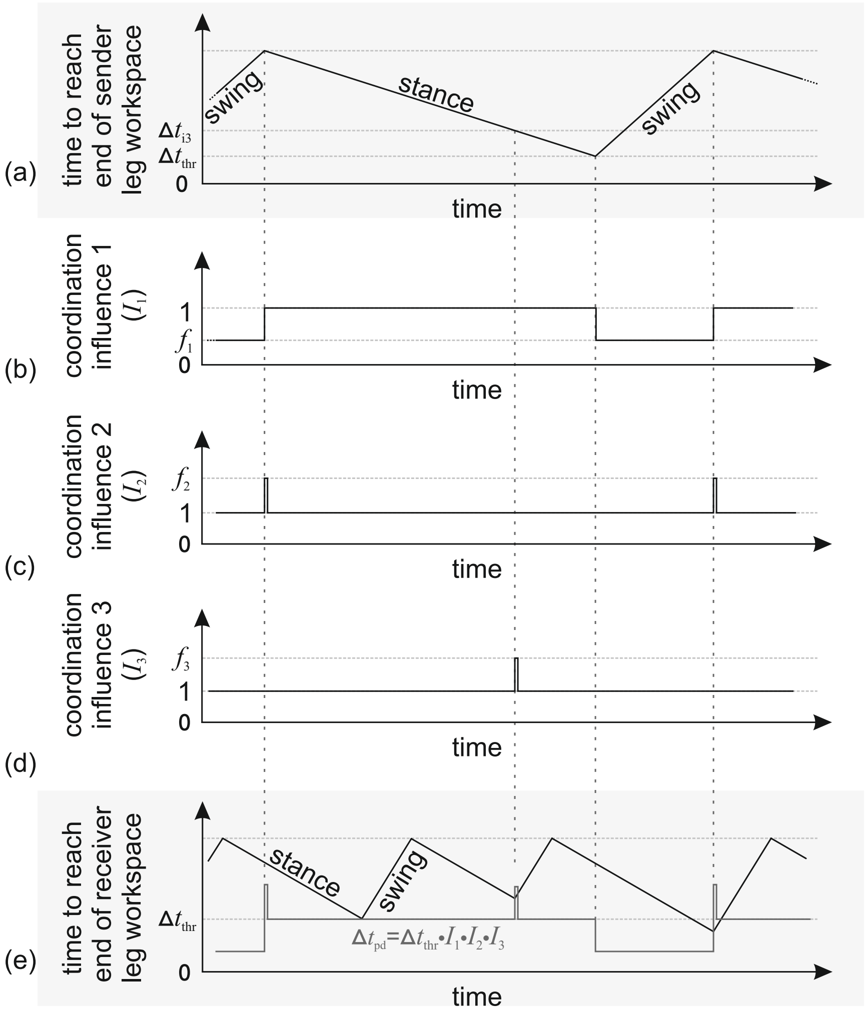


Figure 5.12.: Conceptual depiction of the coordination influences. In (a), the stance-swing-cycles of the sender leg are shown. Its state during the stance movement is represented by the time the leg would require to reach the limits of its stance workspace. During swing phase, this time increases as the leg moves away from the workspace limit that restricted the stance movement. (b)–(d) depict the conceptual effects of the coordination influences. (b) While the sender leg is in swing phase the coordination factor of influence 1 is decreased. (c) Directly after the sender leg switches to stance phase, the coordination factor of influence 2 is increased for a short period (one iteration of the walking controller). (d) When the sender leg crosses the threshold Δt_{r3}, influence factor 3 is increased for a short period (one iteration of the walking controller). In (e), the stance-swing-cycles of the receiver leg are shown. Whenever the time until the leg would reach the limit of its stance workspace drops below the predetermination period (product of Δt_{thr} and the three influences), a stance-swing-transition is initiated.

5.3. The Stance Module

As shown in fig. 5.1, in classical WALKNET, dedicated modules are responsible for movement generation during stance and swing phases. In contrast to the swing modules that control their respective legs independently, the movement of the stancing legs must be coordinated since the legs are mechanically coupled via the ground.

Therefore, also in the new walking controller one of the main tasks of the stance module is the synchronization of movements of the stancing legs to propel the body in the desired direction. During forward walking, synchronicity can be easily achieved since it is sufficient to move the feet straight backwards relative to the body. However, to enable omnidirectional walking (e.g., curve walking, turning) it is required to coordinate the leg movements during complex stance trajectories. Based on the concept of an internal body model by Schilling, Paskarbeit, Schmitz, et al. (2012), a similar interface was implemented that facilitates the manipulation of a virtual kinematic model of the robot (application of displacement vectors to the virtual body) to obtain synchronized leg trajectories. The details of this approach are described in section 5.3.2.

To allow for locomotion in rough terrain, also the stance heights of the legs must be coordinated. For example, if one of the legs steps on a raised object, without the stance height adaptation some of the legs in stance phase might be lifted off the ground. A solution for this problem is the reduction of the stance height of the leg that stands on the obstacle and the increase of the other legs' stance heights. This compensation is achieved using a system of virtual mass-spring-damper systems that represent the individual legs. This concept is presented in section 5.3.1.

5.3.1. Stance Height Adaptation

Using a virtual mass-spring-damper network (fig. 5.13 (b)) in combination with SVD (Singular Value Decomposition), a compromise between the desired and actual stance heights of the legs can be found. During each iteration of the walking controller three steps are processed to compute the desired posture for the next iteration. This process is schematically depicted in fig. 5.13. The details will be given in the next paragraphs. However, summarized, the leg tip positions are represented by masses that are connected via parallel springs and dampers with the robot plane (depicted by a dashed line in fig. 5.13 (b-f)). In the first step, the spring lengths are adjusted to the actual stance heights of the legs. Based on this state of the network, in the second step, its state for the next iteration (after a period of $1/f_{\text{contr}}$, the duration of one iteration of the walking controller) is computed. Since in this step the masses are not connected, each system will tend towards its individual resting state. This might change the distances between the positions of the masses. Since the spring lengths represent the legs' stance heights they cannot be used directly as desired leg positions for the next iteration. Therefore, to maintain the distances between the leg positions, SVD is used in the third step to find a transformation matrix that fits the new mass positions (from step 2) onto the actual leg positions. The obtained transformation matrix is then used to control the leg movements. During the next iteration of the walking controller, this cycle starts again with the first of these steps. Over time, the system will relax to an equilibrium in which the stretched and the compressed springs compensate. The details of this process are described in the following paragraphs:

adjustment of mass-spring-damper network to robot state As mentioned, a virtual network of mass-spring-damper systems is used to represent the leg positions $\boldsymbol{p}_{\mathrm{l},n} = \{x_n,\, y_n,\, z_n\}$ with $n = 1 \ldots 6$.[6] Each of the masses m_n is connected via a spring (with spring constant k_n and uncompressed length $z_{0,n}$) and a viscous damper (with damping coefficient c_n) to the horizontal plane of the RCS (x pointing to the front, y to the left, and z upwards; see fig. 5.13). The springs and dampers are aligned with the z-axis of the RCS. The lengths of the springs z_n (and therefore the distances of the masses from the horizontal plane) correspond to the current stance heights of the legs. This corresponds to step ① in fig. 5.13.

partial relaxation of mass-spring-damper network In this step, the virtual mass-spring-damper systems are evaluated independently of each other—therefore, the masses can move as if the corresponding legs did not form a closed kinematic chain. This corresponds to step ② in fig. 5.13. The movement of the masses is modeled by the second-order differential equation

$$m_n \frac{\mathrm{d}^2 z_n}{\mathrm{d}t^2} + k_n (z_n - z_{0,n}) + c_n \frac{\mathrm{d}z_n}{\mathrm{d}t} = 0 \quad . \tag{5.16}$$

Since oscillations are undesired, critical damping is assured by setting the damping constant $c_n = 2 m_n \sqrt{\frac{k_n}{m_n}}$.[7] Since the equation can be brought to a form in which the spring constant and the mass appear always as combination of $\sqrt{\frac{k_n}{m_n}}$, this ratio will be replaced by a factor $\omega_n = \sqrt{\frac{k_n}{m_n}}$ that corresponds to the undamped natural frequency of the mass-spring system. A higher value of ω_n (either due to a stiffer spring or a reduced mass) leads to a quicker reaction to disturbances.

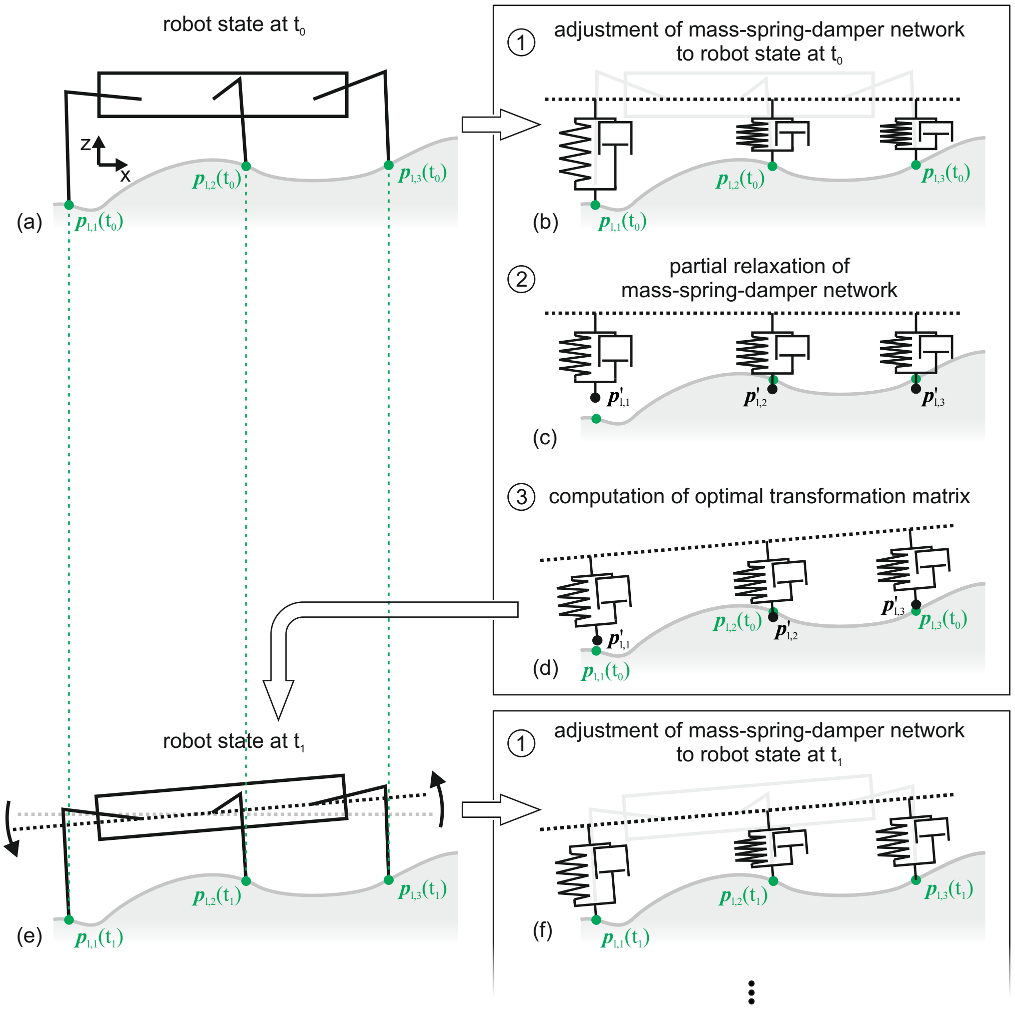
Solving eq. (5.16), the preliminary position z'_n of the mass at time t_{i+1} can be calculated based on its position at time t_i and the current velocity $v_n = \left(z_n(t_i) - z_n(t_{i-1})\right) / (t_i - t_{i-1})$ of the mass:[8]

$$\begin{split} z'_n(t_{i+1}) = e^{-\omega_n t_{i+1}} \Bigg(e^{\omega_n t_{i+1}} z_0 + e^{\omega_n t_i} \bigg(z(t_i) + \Big(v_n + \omega_n z(t_i) \Big) (t_{i+1} - t_i) \\ + z_0 \Big(1 - \omega_n (t_{i+1} - t_i) \Big) \bigg) \Bigg). \end{split} \tag{5.17}$$

[6] The subscript l in $\boldsymbol{p}_{\mathrm{l},n}$ is used to distinguish the leg tip positions from the *virtual pull points* that will be introduced in section 5.3.2

[7] Although the damping constant is chosen such that the system is critically damped, due to the iterative evaluation, numerical instabilities of the system cannot be precluded.

[8] For the first iteration of this step, the velocities of the masses are initialized to zero.



robot state at t_0
z
x
(a)
$p_{1,1}(t_0)$
$p_{1,2}(t_0)$
$p_{1,3}(t_0)$
① adjustment of mass-spring-damper network to robot state at t_0
(b)
② partial relaxation of mass-spring-damper network
$p'_{1,1}$
$p'_{1,2}$
$p'_{1,3}$
(c)
③ computation of optimal transformation matrix
(d)
robot state at t_1
(e)
$p_{1,1}(t_1)$
$p_{1,2}(t_1)$
$p_{1,3}(t_1)$
① adjustment of mass-spring-damper network to robot state at t_1
(f)

Figure 5.13.: Schematic depiction of a stance controller evaluation. (a) shows the state of the robot at time t_0. All feet are in contact with the ground, however, the stance heights vary. (b) shows the virtual mass-spring-damper systems between the horizontal plane of the RCS and the foot positions. (c) shows the state after the virtual mass-spring-damper systems have been evaluated independently. The left spring has decreased its length, whereas the previously compressed springs in the middle and on the right side have increased their length. Now, the positions of the masses do not match the actual positions of the feet anymore. (d) Using SVD a transformation matrix is computed that relocates the positions of the masses in a least-squares optimal way onto the actual foot positions. By applying this transformation matrix to the positions of the leg tips (relative to the RCS) and commanding the legs to move to these new positions, effectively, the body of the robot is moved. The new horizontal plane is slanted relative to the robot in the old posture. (e) shows again the robot's posture at time t_0 with the old and the new horizontal plane of it's coordinate system. (f) shows the robot's posture at time t_1 with the right side inclined. The differences in the stance heights of the legs have been reduced.

Using this equation, preliminary z-positions of the masses are obtained. Using these positions directly as z-components of the leg tip positions, the distances between the leg tips would change which would induce stress in the structure of the robot. To prevent this, the distances between the leg tips must be maintained. To do so, however, preliminary leg positions $\boldsymbol{p}'_{1,n}$ are composed of the preliminary z-positions of the masses and the actual x- and y- positions of the leg tips:

$$\boldsymbol{p}'_{1,n} = \begin{pmatrix} x_n \\ y_n \\ z'_n \end{pmatrix} \tag{5.18}$$

computation of an optimal transformation matrix To maintain the original distances between the leg tips, a 4x4 homogeneous transformation matrix $\boldsymbol{A}_h$ is required that rotates/translates the current leg positions $\boldsymbol{p}_{1,n}(t_i)$ (previously referred to as $\boldsymbol{p}_{1,n}$) to the next iteration's leg positions $\boldsymbol{p}_{1,n}(t_{i+1})$:

$$\begin{pmatrix} \boldsymbol{p}_{1,n}(t_{i+1}) \\ 1 \end{pmatrix} = \boldsymbol{A}_h \cdot \begin{pmatrix} \boldsymbol{p}_{1,n}(t_i) \\ 1 \end{pmatrix} \tag{5.19}$$

To consider the results of the evaluation of the mass-spring-damper network for this transformation, $\boldsymbol{A}_h$ is supposed to minimize the sum of squares of distances between the new leg positions $\boldsymbol{p}_{1,n}(t_{i+1})$ and the preliminary leg positions $\boldsymbol{p}'_{1,n}$. This will be approximated using SVD (Arun et al. 1987), depicted in fig. 5.13 as step ③. For this purpose, the centroids of the current and the preliminary leg positions

$$\boldsymbol{c} = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{p}_{1,j}(t_i) \quad , \tag{5.20}$$

$$\boldsymbol{c}' = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{p}'_{1,j} \tag{5.21}$$

and the covariance matrix

$$\boldsymbol{H} = \sum_{j=1}^{N} (\boldsymbol{p}_{1,j}(t_i) - \boldsymbol{c}) \cdot (\boldsymbol{p}'_{1,j} - \boldsymbol{c}')^T \tag{5.22}$$

must be computed. By applying the SVD to $\boldsymbol{H}$, it can be decomposed into the rotation matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, and the scaling matrix $\boldsymbol{S}$ such that $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$:

$$[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = \mathrm{SVD}(\boldsymbol{H}) \tag{5.23}$$

Based on this, the optimal rotation matrix can be computed by

$$\boldsymbol{R} = \boldsymbol{V}\,\boldsymbol{U}^T \quad . \tag{5.24}$$

For special cases, the SVD returns a *reflection matrix* that cannot be used directly for the rotation of the current leg tip positions. In order to check for this, the determinant of $\boldsymbol{R}$ must be computed. If $\det \boldsymbol{R} = 1$, $\boldsymbol{R}$ is a valid rotation matrix that can be used without further modifications. If $\det \boldsymbol{R} = -1$, $\boldsymbol{R}$ is a reflection matrix. In this case, the signs of the elements in the third column of $\boldsymbol{R}$ must be changed to obtain a valid rotation matrix.

For the translational part of the homogeneous transformation matrix, the required translation $\boldsymbol{d}$ can be computed by

$$\boldsymbol{d} = \boldsymbol{c}' - \boldsymbol{R} \cdot \boldsymbol{c} \quad . \tag{5.25}$$

Using the rotation matrix $\boldsymbol{R}$ and the translation $\boldsymbol{d}$, the homogeneous transformation matrix for stance height control $\boldsymbol{A}_h$ can be constructed:

$$\boldsymbol{A}_h = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{d} \\ 0 & 1 \end{pmatrix} \tag{5.26}$$

Based on this result, the leg tip positions $\boldsymbol{p}_{\mathrm{l},n}(t_{i+1})$ for the next iteration of the walking controller can be computed using eq. (5.19). During the next iteration of the walking controller the z-components of the newly computed foot positions $\boldsymbol{p}_{\mathrm{l},n}(t_{i+1})$ are used as mass positions z_n in step ①.

stance height control during swing phases In situations in which a leg that stood on an obstacle switches to swing phase, it might be problematic to ignore this leg in the stance height control. For example, if this leg has raised the stance height of the neighboring legs, the omission of this influence could result in an abrupt subsiding of stance height and consequentially in collisions of the main body and the obstacle. To prevent these sudden changes of stance height, the position of a previously stancing leg will still be considered by the controller as if the leg continued the stance phase. After touchdown, when the leg changed from swing to stance phase, the new foot position will be used. In order to achieve a smooth, continuous movement of the leg, the velocity of the mass is set to a value such as if the leg would have been in stance phase at the new position already during the previous iteration:

$$\begin{aligned} v_n(t_i) &= \frac{\Delta z}{t_i - t_{i-1}} \quad \text{with} & (5.27) \\ \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ 1 \end{pmatrix} &= \begin{pmatrix} \boldsymbol{p}_{\mathrm{l},n}(t_i) \\ 1 \end{pmatrix} - \boldsymbol{A}_{\mathrm{h}}^{-1} \cdot \begin{pmatrix} \boldsymbol{p}_{\mathrm{l},n}(t_i) \\ 1 \end{pmatrix} & (5.28) \end{aligned}$$

In this case, $\boldsymbol{p}_{\mathrm{l},n}(t_i)$ is the position of the leg tip after touchdown and $\boldsymbol{A}_{\mathrm{h}}$ is the transformation matrix for the transformation of the leg positions between t_{i-1} and t_i.

5.3.2. Direction Control

With the previously described mechanism, the stance heights of the legs can be adapted in unison such that no tension is introduced in the robot structure. However, to enable omnidirectional walking, beside these vertical movements, also horizontal leg movements must be realized by the walking controller to propel the body in the desired direction. As for the stance height controller, the relative distances between the leg tips must be maintained during this process. Therefore, a simple approach that moves the outer legs with a higher speed than the inner legs (similar to the differential steering used for bulldozers and tanks) would result in structural stress and high torques in the drives or slipping of the leg tips over the ground. Beside the wear of the material, this would also result in a higher energy consumption.

A perfect coordination of the leg trajectories is only possible using a centrally organized module that controls the movements of all legs during stance phase. However, if a certain level of stress in the mechanics is acceptable, also the stance trajectory can be realized decentrally using positive velocity feedback, as shown by Cruse, Bartling, and Kindermann (1995) and Schmitz, Schneider, et al. (2008). In another approach, Lévy and Cruse (2008) used decentral controllers that iteratively try to reduce the sum of torques in all joints to reduce the overall stress. However, for the implementation on Hector a central module was favored to reduce the torques and thus the energy consumption in the leg joints.

Beside the requirement to be compatible with the stance height control, the direction controller is supposed to offer an intuitive interface for manual control by an operator (e.g., by joystick) but also for algorithmic control of the desired walking direction (e.g., based on visual information; see section 7.2).

In the concept proposed by Schilling, Paskarbeit, Schmitz, et al. (2012), an internal body model based on an MMC (Mean of Multiple Computation) network is used to calculate the leg movements. Although excessively simplified, the underlying concept can be imagined as a main body with springs instead of legs. In this model, the springs connect the body with the actual foot points. By pulling at one point of the main body, its position is shifted relative to the foot points since the springs comply. For the control of the real robot the new foot points relative to the main body are then used as target positions for the leg movements. Based on this *virtual pulling* of the main body, a direction control concept can be created that interacts with the stance height control.

To control the walking direction *pull points* are defined at which the virtually body can be pulled in the desired direction. Internally, these pull vectors are transformed into a rotatory component ω and a translatory component $\boldsymbol{d}$ that describe the transformation of a point that is located in the middle between the two pull points (see fig. 5.14). Relative to a global coordinate system in which the leg tip positions are fixed, the rotation by ω around the vertical axis and the translation by $\boldsymbol{d}$ are sufficient to describe the movement of the body in two dimensions. Based on these values the required movements of the stancing legs can be computed that result in the desired body movement.

Figure 5.14 shows two examples for the movement of the feet as induced by pulling at the virtually body to illustrate the approach. The general setup is shown in (a) with the two pull points (between the front and hind leg onsets, respectively) visualized by blue circles. The current positions of the feet are marked by red dots. Although two pull points are depicted, they do not have to be used in conjunction. In the first example,

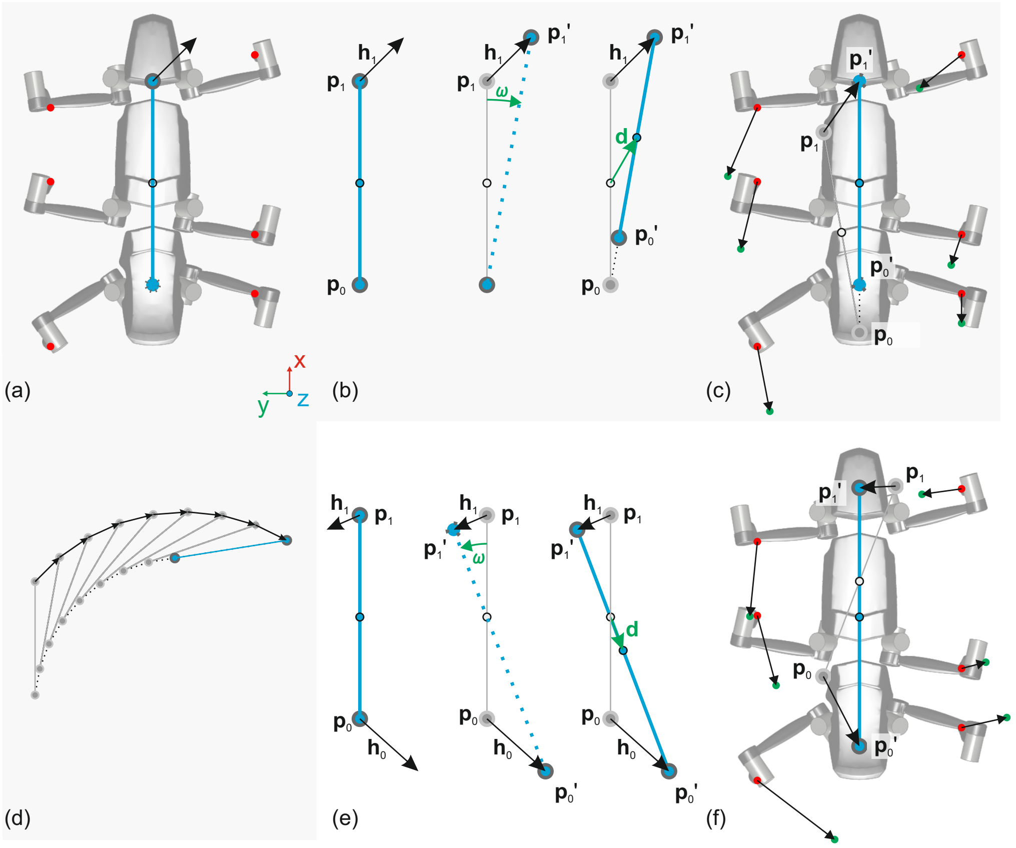


Figure 5.14.: Depiction of the trajectory generation for stancing legs for two cases, (a)–(d) and (e)–(f). In the first example, (a)–(d), only one of the pull points is used to direct the robot. The second example shows a rotational movement, for which both pull points must be used.
(a) shows the two pull points that are fixed to the virtual robot body, marked as blue points, and the positions of the leg tips as red dots. In (b), the concept for the computation of the rotation angle ω and the translation vector $\boldsymbol{d}$ is illustrated. Based on these two values a transformation matrix can be constructed (see eq. (5.34)). By applying the inverse of this transformation on the leg tip positions (relative to the RCS), the leg trajectories for the next time step can be calculated as depicted in (c). The green dots represent the desired foot positions for the next control cycle. (d) shows the resulting trajectory of the pull points (in world coordinates) for a subsequent application of these transformations.
In (e), the computation of ω and $\boldsymbol{d}$ is illustrated for a different example in which both of the pull points are shifted at the same time in different directions. (f) shows again the desired trajectories of the leg tips for the next time step.

visualized in (a)–(d), only the frontal point is pulled diagonally forwards. In (b), the pull points and the central axis are shown to illustrate the computation of $\boldsymbol{d}$ and ω. Based on the pull vector, the point $\boldsymbol{p}_1 = \{x_1, y_1\}$ is shifted by the heading vector $\boldsymbol{h}_1$ to its new position $\boldsymbol{p}'_1 = \{x'_1, y'_1\} = \boldsymbol{p}_1 + \boldsymbol{h}_1$. With the position of the rear point $\boldsymbol{p}_0 = \{x_0, y_0\}$, the angle ω can be computed:

$$\omega = \operatorname{atan2}\left(y'_1 - y'_0, x_1 - x_0\right) - \operatorname{atan2}\left(y_1 - y_0, x_1 - x_0\right) \tag{5.29}$$

As mentioned before, $\boldsymbol{d}$ refers to the movement of the point in the middle between the two pull points. In this case, however, the rear point is not shifted. Therefore, the movement of the middle point does not need to compromise between the two pull vectors. The desired movement (shifting $\boldsymbol{p}_1$ to $\boldsymbol{p}'_1$) can be achieved by assuming an implicit shift of $\boldsymbol{p}_0$ (depicted in the right pictogram of (b)) that maintains the distance between the pull points. For this example, $\boldsymbol{d}$ can then be computed by

$$\boldsymbol{d} = \boldsymbol{p}'_1 - \frac{\|\boldsymbol{p}_1 - \boldsymbol{p}_0\|}{2} \cdot \left(\boldsymbol{p}'_1 - \boldsymbol{p}_0\right) - \frac{\boldsymbol{p}_1 - \boldsymbol{p}_0}{2} \quad . \tag{5.30}$$

In the second example that is shown in fig. 5.14 (e,f), both pull points are shifted by the heading vectors $\boldsymbol{h}_0$ and $\boldsymbol{h}_1$ to manipulate the robot posture. In this case, the two variables $\boldsymbol{d}$ and ω are computed by

$$\boldsymbol{d} = \frac{\boldsymbol{p}'_1 + \boldsymbol{p}'_0}{2} - \frac{\boldsymbol{p}_1 + \boldsymbol{p}_0}{2} \quad \text{and} \tag{5.31}$$

$$\omega = \operatorname{atan2}\left(y'_1 - y'_0, x'_1 - x'_0\right) - \operatorname{atan2}\left(y_1 - y_0, x_1 - x_0\right) \quad \text{with} \tag{5.32}$$

$$\boldsymbol{p}'_1 = \boldsymbol{p}_1 + \boldsymbol{h}_1 \quad \text{and} \quad \boldsymbol{p}'_0 = \boldsymbol{p}_0 + \boldsymbol{h}_0 \quad . \tag{5.33}$$

The corresponding homogeneous transformation matrices (representing a horizontal translation and a rotation about the vertical axis) can be obtained by

$$\boldsymbol{A}_\mathrm{d} = \begin{pmatrix} 1 & 0 & 0 & -(x_0 + x_1)/2 \\ 0 & 1 & 0 & -(y_0 + y_1)/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\omega & -\sin\omega & 0 & d_x \\ \sin\omega & \cos\omega & 0 & d_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & (x_0 + x_1)/2 \\ 0 & 1 & 0 & (y_0 + y_1)/2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad . \tag{5.34}$$

This transformation describes the movement of the robot's body relative to a fixed coordinate system. However, since the translation of the foot points must be calculated relative to the local RCS, the inverse transformation must be applied to the current foot positions $\boldsymbol{p}_{\mathrm{l},n}(t_i)$ to obtain the goal positions $\boldsymbol{p}_{\mathrm{l},n}(t_{i+1})$ for the next iteration:

$$\begin{pmatrix} \boldsymbol{p}_{\mathrm{l},n}(t_{i+1}) \\ 1 \end{pmatrix} = \boldsymbol{A}_\mathrm{d}^{-1} \cdot \begin{pmatrix} \boldsymbol{p}_{\mathrm{l},n}(t_i) \\ 1 \end{pmatrix} \tag{5.35}$$

The resulting movement vectors of the stancing legs relative to the RCS are depicted in (c) and (d) for the two examples, respectively.

As mentioned before, a requirement of the direction control was the compatibility with the stance height controller. To combine the two controllers, the composition of the preliminary leg positions $\boldsymbol{p}'_{\mathrm{l},n}$ (see eq. (5.18)) can be modified to incorporate the direction control:

$$\begin{pmatrix} \boldsymbol{p}'_{\mathrm{l},n} \\ 1 \end{pmatrix} = \boldsymbol{A}_{\mathrm{d}}^{-1} \begin{pmatrix} x_n \\ y_n \\ z'_n \\ 1 \end{pmatrix} \tag{5.36}$$

This operation shifts the preliminary leg positions such that they account—beside the stance height adaptation—also for the desired walking direction. The following steps of the stance height controller do not have to be modified. As a result of this minor change, the homogeneous transformation matrix $\boldsymbol{A}_h$ (see eq. (5.26)) represents the stance height adaptions of the legs and the walking direction.

5.4. The Swing Module

As opposed to the stance phase, the movement of swinging legs does not need to be coordinated since they are not coupled via the ground. Therefore, the swing trajectories could be generated decentrally as, e.g., proposed for Walknet (Cruse, Bartling, Dreifert, et al. 1995). However, to choose an appropriate target position for the swing movement, the postures of adjacent legs must be considered to avoid collisions. Also, to realize omnidirectional walking, the current walking direction must be taken into account (the swing movement should oppose the walking direction to maximize the length of the subsequent stance trajectory). Therefore, the swing controllers are provided with information on the current state of their neighboring legs and the walking direction.

In Walknet, the task of swing trajectory generation is divided between the *target net* and the *swing net*. In this concept, the *target net* computes a position (based on the position of the next anterior leg) the swing movement should target at. The *swing net* then translates this target position to joint velocities that are used to control the leg joints. Thus, iteratively, a leg trajectory is generated that should ideally end at the provided target position. Within the framework of Walknet, *swing* and *target nets* were usually based on ANNs. The first implementation of a *swing net* could replicate the trajectory of insect legs during swing although the speed profile did not match the biological data (Cruse, Bartling, Dreifert, et al. 1995). The following variations (*swing nets* 2 and 3) included LPFs (Low Pass Filters) and were able to replicate also the speed profile (Schumm and Cruse 2006). Another advantage of the latter versions is the ability to generate swing trajectories for targets in all directions (e.g., also for backward walking). The disadvantage of the ANN-based approaches for *swing* and *target nets* is the need to train the networks prior to usage.

Another approach that has been successively used to generate end effector trajectories is based on vector fields (Khatib 1986). As shown by Cruse, Bläsing, et al. (2004) the output of the ANN-based *swing nets* can also be represented by a vector field. In combination with the *restrictedness*-approach, vector fields have also been used by Fielding (2002) to generate swing trajectories. Since the focus of this work is not the biomimetic implementation but rather the utilization of bioinspired concepts, the vector field based approach was favored as it is directly adaptable to the desired trajectories without the need of preceding training. The underlying concept for the trajectory generation is explained in section 5.4.2.

However, prior to every swing phase, a suitable target position must be selected. As mentioned before, this is closely coupled to the direction of walking. Therefore, the current transformation matrix of the stance module—or rather the inverse of it—is used for the swing target selection. This is described in section 5.4.1.

5.4.1. Selection of a Swing-Target

As pointed out in section 5.2.2, the swing target must be selected within the swing-target workspace (see section 5.2.2), which is the intersection of swing and stance workspaces. In addition, to facilitate smooth operation, following factors must be considered as well: Based on the current walking direction the swing target should be chosen to increase the potential length of the following stance phase (therefore until the limit of the stance workspace is reached during the subsequent stance movement). Also, target positions that might result in collisions with other legs must be avoided. Since both of these factors might vary during the course of a swing phase the selection of an optimal swing target would require considerable pre-planning which is beyond the scope of this work. Also, this level of pre-planning does not fit to the reactive, adaptive concept of decentral coordination. Therefore, the *direction* of swing is determined based on the state at liftoff. However, the actual *target position* of the swing movement along this direction is updated frequently, e.g., to prevent collisions with neighboring legs. Since smooth changes of walking direction are expected, the current walking direction is assumed to be maintained for the immediate future just as it is done during the process for the decision whether or not to start the swing phase within the selector module.

In fig. 5.15, the general concept for the selection of a swing target is depicted. As already proposed by Espenschied and Quinn (1994) and Fielding (2002) a fixed *home position* for each leg is defined that lies within the respective swing-target-workspace. This *home position* is the starting point in the process to determine a suitable target position: The swing target is chosen such that—assuming the current walking direction is maintained—the subsequent stance trajectory would pass through the *home position*. Potential candidates for the target position that suit this requirement (points along the red, dashed line in fig. 5.15 (b)) are computed by subsequent multiplication of the inverse stance transformation matrix $\boldsymbol{A}_h$ with the home position $\boldsymbol{p}_h$:

$$\boldsymbol{p}(n) = \boldsymbol{A}_h^{-n} \cdot \boldsymbol{p}_h \quad \text{with } n \geq 0. \tag{5.37}$$

$\boldsymbol{p}(n)$ therefore defines all points, from which a stance movement could be started such that it would pass $\boldsymbol{p}_h$ (assuming again the maintenance of the current walking direction).

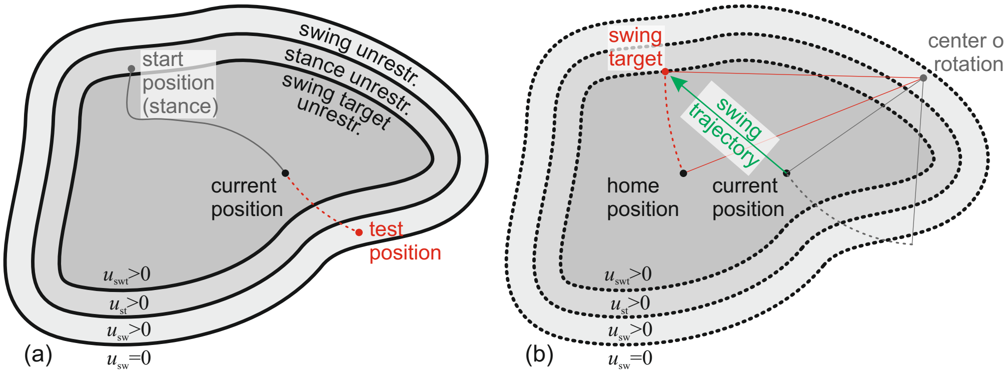


Figure 5.15.: Depiction of the process for swing target selection. (a) shows the course of the stance trajectory that has already been shown in fig. 5.10 (b). In addition to the previously shown limits of the stance unrestrictedness also the limits of the swing and swing target unrestrictednesses are depicted. The *test position* in the depicted example lies outside of the stance restrictedness. Therefore, the swing phase will be initiated. In (b), the selection of a swing target is shown. Starting from the *home position* the *reverse stance trajectory* (red, dashed line) is computed based on the center of rotation of the last stance movement (gray, dashed line). The furthest point along the *reverse stance trajectory* that remains within the swing target unrestrictedness is selected as target position for the swing movement that starts at the current position of the leg tip.

To include the requirement that the target position must lie within the swing-target-workspace, the furthest target position $\boldsymbol{p}_{\max}$ is determined by

$$\boldsymbol{p}_{\max} = \boldsymbol{p}(n_{\max}) \quad \text{with} \tag{5.38}$$

$$n_{\max} = \max\left(\{n|\boldsymbol{p}(n) \in W_{\mathrm{swt}}\}\right) \quad . \tag{5.39}$$

Since the swing target workspace W_{swt} changes continuously during walking due to the varying positions of the adjacent legs, $n_{\max}$ must be checked in each iteration of the walking controller. If $\boldsymbol{p}_{\max}$ is located outside the swing-target workspace, $n_{\max}$ must be reduced until the target position meets the requirement again. Although this check is performed in each iteration of the walking controller, eq. (5.37) is only evaluated once during each swing phase, at liftoff. Therefore, the path, along which the actual target is positioned, does not change throughout the swing phase. This measure was taken to prevent sudden changes of the swing direction in response to changes of walking direction. However, if, e.g., the walking direction is reversed during the swing movement, this will likely result in a direct re-liftoff right after touch-down of the swinging leg.

To reduce the tendency to start a new swing phase immediately upon touch-down the swing target position $\boldsymbol{p}_{\mathrm{swt}}$ can be chosen with a temporal distance Δt_{d} from the limits of the workspace:

$$\boldsymbol{p}_{\text{swt}} = \boldsymbol{p}\left(n_{\max} - \Delta t_{\text{d}} \cdot f_{\text{contr}}\right) \quad \text{with} \quad \Delta t_{\text{d}} < \frac{n_{\max}}{f_{\text{contr}}} \quad , \tag{5.40}$$

in which f_{contr} is the frequency of the walking controller.

To prevent an infinite loop for the evaluation of eq. (5.39) in situations in which the robot does not move at all (in this case, $\boldsymbol{A}_h$ will be an identity matrix and therefore the limits of the workspace cannot be reached), a maximum limit for $n_{\max}$ is defined. As a consequence of this limitation, if the robot does not move, the legs will swing to their home positions.

5.4.2. Trajectory Generation

Since the swinging legs are not coupled via the ground, the swing trajectories can be chosen independently. Therefore, no universally optimal swing trajectory can be defined. If the environment is known the trajectory can be optimized regarding the time or energy consumption for the complete swing phase (Erden 2011). In these cases the swing height would probably be reduced as much as possible without the risk of a collision.

On Hector, however, no mapping of the environment is planned that would allow these kinds of optimizations. Therefore, a robust trajectory generator is required that can deal with disturbances (e.g., collisions) during the movement. As already discussed, vector field based approaches have been successively applied for this task (Fielding 2002). Instead of planning a detailed trajectory that must be followed during the course of the swing phase, a mapping in form of a vector field between the current position of the leg tip and the desired leg tip velocity for the next iteration is defined. Thus, with a suitably designed vector field, perturbations from the anticipated trajectory are admissible without the need to replan the trajectory.

As Schumm and Cruse (2006) described, the swing trajectories of stick insects “can be roughly approximated by a section of a circle”. Since no other optimality criterion (e.g., energy optimality) is planned to be applied, the circular swing trajectories of the insects were used as starting point for the design of the vector field. However, a perfect circular trajectory might not be reasonable in all cases. For example, if the leg is close to its target position already during liftoff, the circle’s radius and thus the swing height will be too small to lift the leg due to the compliance and it might scrape over the ground. In these cases, e.g., an elliptical trajectory that lifts the leg to a certain pre-defined minimal swing height would be advantageous. To combine the circular swing trajectories of the insect model with the requirements of the technical system (minimal swing height), a vector field is designed that creates circular and (if required to achieve a minimal swing height) ellipsoid trajectories. Although, as mentioned, the strict following of a predefined trajectory is not essential, the target position (see section 5.4.1) should still be reached even if the leg deviates from the ideal trajectory. To realize this, a three-dimensional vector field is required that guides the leg towards its target position.

To define this three-dimensional vector field, first of all, the ideal circular/ellipsoid trajectory will be planned in two dimensions. In the following, this trajectory will be

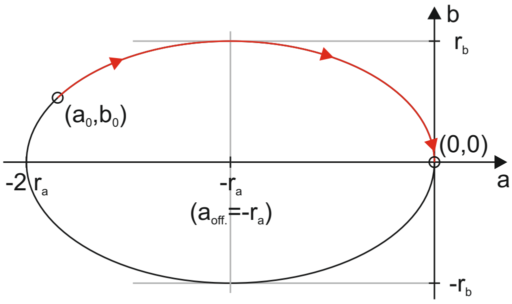


Figure 5.16.: Elliptical swing trajectory starting at point (a_0, b_0), targeting at point $(0, 0)$.

used for the definition of a two-dimensional vector field, which in turn will be expanded to three dimensions.

definition of the ideal trajectory As mentioned before, the trajectory that should emerge when following the vector field should be circular or elliptical. Since a circle is a special case of an ellipse with equal major and minor axes, the more general case of elliptical trajectories will be considered. The equation for an ellipse whose major and minor axes are aligned with the canonical coordinate axes a and b can be written in the form:

$$1 = \frac{a}{r_\mathrm{a}}^{2} + \frac{b}{r_\mathrm{b}}^{2} . \tag{5.41}$$

r_a and r_b are the semi-major and semi-minor axes. By introducing an offset in a-direction (a_off), the ellipse can be shifted laterally (see fig. 5.16):

$$1 = \frac{a - a_\mathrm{off}}{r_\mathrm{a}}^{2} + \frac{b}{r_\mathrm{b}}^{2} \tag{5.42}$$

Solving this equation for b yields:

$$b(a) = \pm\sqrt{\frac{\left((-a - a_\mathrm{off})^2 - r_\mathrm{a}^2\right) r_\mathrm{b}^2}{r_\mathrm{a}^2}} \quad . \tag{5.43}$$

Since the leg is supposed to be lifted upwards, only the positive signed solution is considered for the further analysis. Setting $a_\mathrm{off} = \pm r_\mathrm{a}$ will guarantee $b(0) = 0$. This ensures that the trajectory will always end at $(0, 0)$. For the swing trajectory it is assumed that the leg will always move from $a < 0$ towards the target position at $a = 0$. Therefore, the offset is set to $a_\mathrm{off} = -r_\mathrm{a}$. The ellipse is supposed to pass also through the point

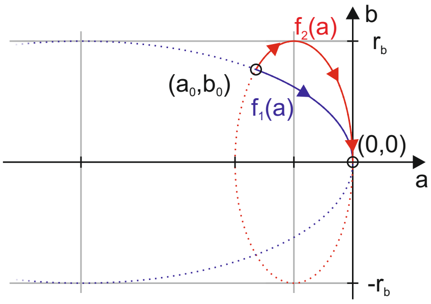


Figure 5.17.: Plot of $f_1(a)$ and $f_2(a)$ (eqs. (5.46) and (5.47)) showing two potential swing trajectories starting from (a_0, b_0), targeting at $(0, 0)$.

(a_0, b_0), at which the swing phase is initiated. Solving $b(a_0) = b_0$ for r_a yields

$$r_{\mathrm{a}} = \pm \frac{a_0 r_{\mathrm{b}} \left(\mp r_{\mathrm{b}} + \sqrt{r_{\mathrm{b}}^2 - b_0^2} \right)}{b_0^2} \quad \text{for } b_0 > 0 \text{ and} \tag{5.44}$$

$$r_{\mathrm{a}} = \frac{a_0}{2} \quad \text{for } b_0 = 0. \tag{5.45}$$

For $b_0 > 0$, two solutions with different signs exist. Applying $a_{\mathrm{off}} = -r_a$ and eq. (5.44) to eq. (5.43) results in two functions that each describe an elliptical curve from (a_0, b_0) to $(0, 0)$ (see fig. 5.17):

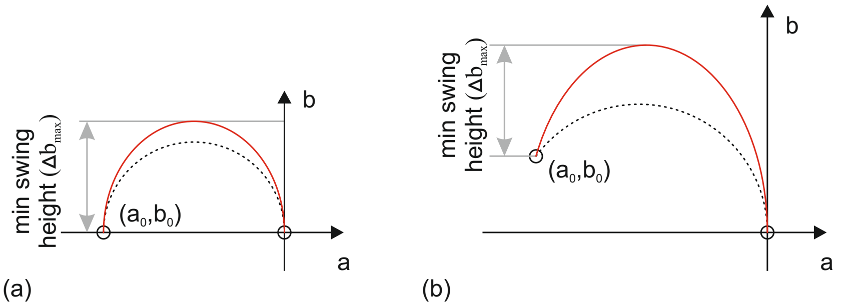
$$f_1(a) = \sqrt{-\frac{a\, b_0^2 \left(a\, b_0^2 - 2 a_0 r_{\mathrm{b}} \left(r_{\mathrm{b}} + \sqrt{r_{\mathrm{b}}^2 - b_0^2} \right) \right)}{a_0^2 \left(r_{\mathrm{b}} + \sqrt{r_{\mathrm{b}}^2 - b_0^2} \right)^2}} \quad \text{and} \tag{5.46}$$

$$f_2(a) = \sqrt{-\frac{a\, b_0^2 \left(a\, b_0^2 - 2 a_0 r_{\mathrm{b}} \left(r_{\mathrm{b}} - \sqrt{r_{\mathrm{b}}^2 - b_0^2} \right) \right)}{a_0^2 \left(r_{\mathrm{b}} - \sqrt{r_{\mathrm{b}}^2 - b_0^2} \right)^2}} \quad . \tag{5.47}$$

These functions depend only on the semi-minor axis r_{b} and the point (a_0, b_0), at which the swing phase starts. Starting at (a_0, b_0), one solution results in a trajectory that lowers the leg directly and the other solution gives a trajectory for which the leg is lifted to the desired swing height r_{b}. Ideally, the leg should be lifted after the stance-swing-transition. To identify the function that fulfills this requirement, the positions of the maxima are computed by solving $\mathrm{d}f_{1/2}(a)/\mathrm{d}a = 0$:

$$a_{\max,1} = \frac{a_0 r_{\mathrm{b}}^2 + a_0 r_{\mathrm{b}} \sqrt{r_{\mathrm{b}}^2 - b_0^2}}{b_0^2} \quad \text{for } f_1 \text{ and} \tag{5.48}$$

$$a_{\max,2} = \frac{a_0 r_{\mathrm{b}}^2 - a_0 r_{\mathrm{b}} \sqrt{r_{\mathrm{b}}^2 - b_0^2}}{b_0^2} \quad \text{for } f_2 \quad . \tag{5.49}$$



min swing height (Δb_{max})
b
(a_0,b_0)
a
(a)
min swing height (Δb_{max})
b
(a_0,b_0)
a
(b)

Figure 5.18.: Two cases for which the circular trajectory would not reach the defined minimum swing height. The dotted line represents the circular trajectory, the red line shows the ellipsoidal trajectory that satisfies the requirement for a minimum swing height relative to the starting point at (a_0, b_0).

To ensure that the leg is lifted after the stance-swing transition, $a_{\max}$ must be located between 0 and a_0, therefore $a_0 < a_{\max} < 0$. This requirement is only fulfilled by $f_2(a)$.

As stated before, the leg should swing on a circular trajectory if the resulting maximal swing height $\Delta b_{\max}$ is sufficient. For a circular trajectory, the semi-major and semi-minor axes must be equal, thus $r_{\mathrm{a}} = r_{\mathrm{b}}$. Applying this to eq. (5.43) and solving for the common radius $r_{\mathrm{ab}} = r_{\mathrm{a}} = r_{\mathrm{b}}$ yields

$$r_{\mathrm{ab}} = \frac{-a_0^2 - b_0^2}{2a_0} \quad . \tag{5.50}$$

For a circular trajectory to be possible, r_{ab} must be bigger than $b_0 + \Delta b_{\max}$, as depicted in fig. 5.18. In addition, $|a_0|$ must be smaller than r_{ab} to make sure the leg is lifted initially.

If a circular trajectory is not possible since the minimal swing height is not reached, r_{b} can be chosen such that it satisfies the requirement ($r_b = b_0 + \Delta b_{\max}$) and the corresponding r_{a} can be calculated using eq. (5.44). The resulting trajectory will then be ellipsoid.

At this point, the ideal circular/ellipsoid trajectory from (a_0, b_0) to $(0, 0)$ is known. Either the common radius r_{ab} can be used or a matching r_{b} with the corresponding r_a (computed by eq. (5.44)).

extension to two-dimensional vector field To generalize the ideal trajectory to a two-dimensional vector field as depicted in fig. 5.19, first of all, the previously determined ratio between the major and minor axes, r_{a} and r_{b}, is fixed. Therefore, it will be computed at liftoff (at the position (a_0, b_0)) and then remain unchanged during the course of the swing movement. With $k = \frac{r_{\mathrm{a}}}{r_{\mathrm{b}}}$, replacing r_{a} by $k\, r_{\mathrm{b}}$, eq. (5.43) yields:

$$b(a) = \sqrt{-\frac{a\,(a + 2k\, r_{\mathrm{b}})}{k^2}} \quad . \tag{5.51}$$

Using the previously computed r_b, this formula still describes the ellipsoid curve that passes through (a_0, b_0) and $(0, 0)$, and reaches at its apex a height of r_b. However, one requirement was the adaptability of the trajectory generation in case of deviations from this ideal curve. Therefore, if the leg tip is positioned at a point $(a_1, b_1$ that does not lie on this curve, the trajectory generator should still guide the leg towards the target position at $(0, 0)$. This can be accomplished by adapting r_b in eq. (5.51), thus creating a family of eccentric, ellipsoid curves (see fig. 5.19) that all pass the target position. Therefore, for every combination of coordinates (a_1, b_1), an r_b can be computed that results in a suitable curve (passing through (a_1, b_1) and $(0, 0)$). Demanding $b(a_1) = b_1$, eq. (5.51) must be solved for r_b:

$$r_\text{b} = \frac{-a_1^2 - k^2\, b_1^2}{2\, k\, a_1} \tag{5.52}$$

Inserting eq. (5.52) into eq. (5.51), this requirement is met:

$$b(a) = \sqrt{\frac{a\, b_1^2}{a_1} - \frac{a^2 - a\, a_1}{k^2}} \tag{5.53}$$

Equation (5.53) defines the trajectory of the swing movement from a point (a_1, b_1). The general shape of the trajectory, however, was previously defined based on the liftoff position (a_0, b_0) and the corresponding choice of k.

Equation (5.53) still maps the a-coordinate to the b-coordinate. For online trajectory generation, however, a mapping of $(a, b) \rightarrow (\dot{a}, \dot{b})$ is required, therefore, from the current position (a, b) to the desired leg tip velocity $\boldsymbol{v}_\text{ab} = (\dot{a}, \dot{b})$. To compute the direction of the movement based on the current position (a_1, b_1), eq. (5.53) is differentiated with respect to a to obtain the slope m:

$$m = \frac{\mathrm{d}b(a)}{\mathrm{d}a} = \frac{|b_1| \left(1 - \frac{a_1^2}{b_1^2 k^2}\right)}{2a_1} \tag{5.54}$$

The direction of movement that is required to follow the trajectory (eq. (5.53)) is therefore defined by

$$\boldsymbol{v}_\text{ab} = \frac{s}{\sqrt{1 + m^2}} \cdot (1, m) \tag{5.55}$$

with the desired speed of the swing movement $s = \|\boldsymbol{v}_{ab}\|$.

extension to three-dimensional vector field To apply the two dimensional vector field for the trajectory generation in three dimensional space, a conversion $(x, y, z) \rightarrow (a, b)$ is required to obtain $\boldsymbol{v}_\text{ab}$ based on eq. (5.55) and finally a conversion $(\dot{a}, \dot{b}) \rightarrow (\dot{x}, \dot{y}, \dot{z})$ is required to convert $\boldsymbol{v}_\text{ab}$ to a leg tip velocity $\boldsymbol{v}_\text{LCS}$ (relative to the LCS), which can be translated to angular velocities of the leg joint drives by the inverse kinematics (see section A.2).

Interpreting the a-b-coordinates as cylindrical coordinates with a being the radius and b the height relative to the target position results in a rotational symmetric vector field

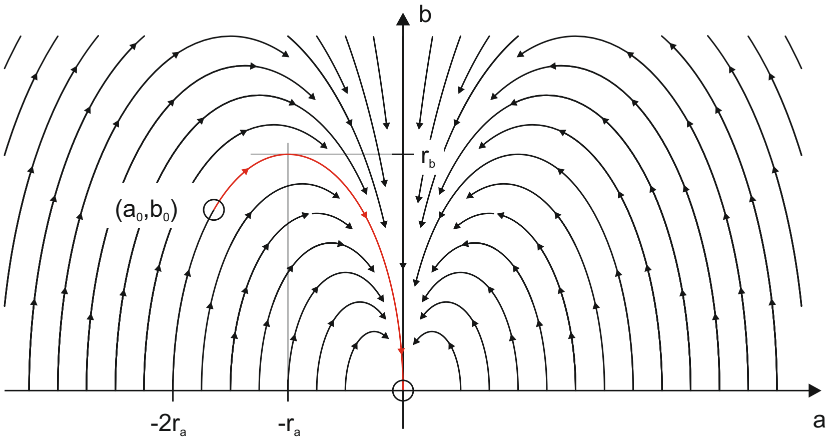


Figure 5.19.: Two dimensional vector field, created for the swing movement from (a_0, b_0) to $(0, 0)$, showing the alternative swing trajectories in case of deviation from the planned path.

in the x-y-z-coordinate system. When the leg reaches its target position at $a = b = 0$, it moves antiparallel to the b-axis. Thus, the orientation of the b-axis in the three dimensional coordinate system can be used to implement a desired touchdown vector v_{td} (see fig. 5.20). In three dimensions, the unit vector of the b-axis can be defined by

$$\boldsymbol{e}_{\mathrm{b}} = -\frac{\boldsymbol{v}_{\mathrm{td}}}{\|\boldsymbol{v}_{\mathrm{td}}\|} \quad . \tag{5.56}$$

Projecting the current position $\boldsymbol{p} = (x, y, z)$ onto the unit vector of the b-axis, the corresponding a-b-coordinates can be computed:

$$b = (\boldsymbol{p} - \boldsymbol{p}_{\mathrm{swt}}) \cdot \boldsymbol{e}_b \tag{5.57}$$

$$a = \|\boldsymbol{p} - (\boldsymbol{p}_{\mathrm{swt}} + b \cdot \boldsymbol{e}_b)\| \tag{5.58}$$

These coordinates can be used to setup a vector field or to compute the next movement during swing phase. Having computed the direction vector $\boldsymbol{v}_{\mathrm{ab}} = (v_{\mathrm{a}}, v_{\mathrm{b}})$ using eq. (5.55) for the next iteration in a-b-coordinates, it can be transformed back into a leg tip velocity $\boldsymbol{v}_{\mathrm{LCS}}$ in x-y-z-coordinates by

$$\boldsymbol{v}_{\mathrm{LCS}} = v_{\mathrm{b}}\, \boldsymbol{e}_{\mathrm{b}} + v_{\mathrm{a}} \frac{\boldsymbol{p}_{\mathrm{target}} - \boldsymbol{p}}{\|\boldsymbol{p}_{\mathrm{target}} - \boldsymbol{p}\|} \quad . \tag{5.59}$$

If the swing movement is started below the a-plane defined by $(\boldsymbol{p} - \boldsymbol{p}_{\mathrm{target}}) \cdot \boldsymbol{e}_{\mathrm{b}} = 0$, the leg is moved antiparallel to v_{td} until it is located above the plane (see fig. 5.20). If the leg does not detect ground contact until it reaches the target position, it will move downwards parallel to the touchdown vector.

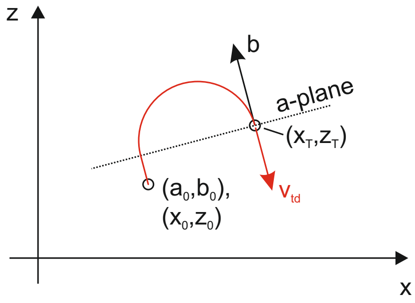


Figure 5.20.: A swing trajectory in the x-z-plane. The trajectory starts at (x_0, z_0) and ends at the target point $(x_\mathrm{T}, z_\mathrm{T})$. Since the starting point is below the a-plane, the trajectory starts with a “vertical” part (relative to the a-b-coordinate system) and changes to the circular trajectory above the a-plane.

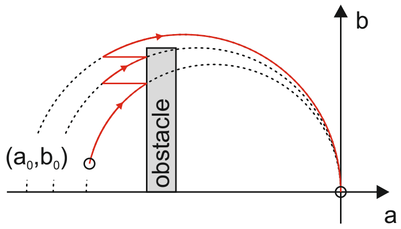


Figure 5.21.: Obstacle evasion during swing phase. If the leg hits an obstacle it is retracted and continues the swing phase along another ellipsoid trajectory as defined by the vector field.

5.4.3. Collision Detection and Evasion Movements

Using eqs. (A.23) and (2.1), the equivalent force $\boldsymbol{f}$ exerted by the tip of the leg onto the environment can be computed based on the torques sensed in the three actuators of the leg. This equivalent force $\boldsymbol{f}$ can be used to detect collisions with obstacles or the ground at the end of swing phase. A simple way to detect a collision would be to define a threshold for the force. If $\|\boldsymbol{f}\| > f_{\text{thr}}$, the leg has likely collided with either an obstacle or the ground. To distinguish between obstacle collisions and ground contact, the direction of force must be considered. Ground contact is assumed if the component of the force parallel to the direction of the touchdown vector v_{td} exceeds the force threshold $f_{\text{thr,gc}}$ (the leg pushes against the ground), therefore if $\boldsymbol{f}_{\|\boldsymbol{v}_{\text{td}}} > f_{\text{thr,gc}}$ with $f_{\|\boldsymbol{v}_{\text{td}}} = \frac{\boldsymbol{f}\cdot\boldsymbol{v}_{\text{td}}}{\|\boldsymbol{v}_{\text{td}}\|}$. An obstacle collision is detected if the norm of the force $\boldsymbol{f}_{\text{coll}}$ that acts perpendicular to the touch down vector $\boldsymbol{v}_{\text{td}}$ exceeds the threshold $\boldsymbol{f}_{\text{thr,coll}}$:

$$\boldsymbol{f}_{\text{coll}} = \boldsymbol{f} \cdot \frac{\boldsymbol{v}_{\perp v_{\text{td}}}}{\|\boldsymbol{v}_{\perp v_{\text{td}}}\|} \quad \text{with} \tag{5.60}$$

$$\boldsymbol{v}_{\perp v_{\text{td}}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} - \frac{\boldsymbol{v}\cdot\boldsymbol{v}_{\text{td}}}{\|\boldsymbol{v}_{\text{td}}\|^2}\boldsymbol{v}_{\text{td}} \tag{5.61}$$

If a collision was detected, the leg is retracted (see fig. 5.21). Due to the concept of the vector field, the swing phase can be continued on another elliptical trajectory that reaches a higher maximal swing height.

5.4.4. Dealing with Restrictions During Swing Phase

Using the presented approach based on ellipsoidal vector fields, for each point in space a velocity vector can be computed. However, the vector field does not incorporate the restrictions such as the joint angle limits. Therefore, if the leg hits a restriction during the swing movement it must cope with it by avoiding the restricted region. If the desired leg position gets too close to the singularity the leg position can be corrected by increasing the distance between the leg tip and the α-axis. If the leg tip leaves the leg tip/segment distance unrestrictedness, the leg can simply be moved away from the leg tip position of the adjacent leg until it is back inside the respective unrestrictedness.

For the joint angle unrestrictedness, however, a simple clipping of the joint angles at their limits does not always solve the problem. As described in section 2.1, the movements of the joint are coupled. Therefore, if the β-joint is at its limit for the upward movement of the *femur* and the trajectory generator still suggests to move the leg tip higher, this will not only affect the desired β-angle but also the desired α- and γ-angles. Figure 5.22 shows exemplarily why a clipping of the joint angles at their limits will not work. If the β-joint is already at its upper limit and the desired position for the next iteration requires a further increase of the β-angle, using this concept, only the α- and γ-joints will be moved. In some cases, this will result in a vicious circle since the movement of α- and γ-joints might even lower the position of the leg tip by shifting its desired position in- or outwards.

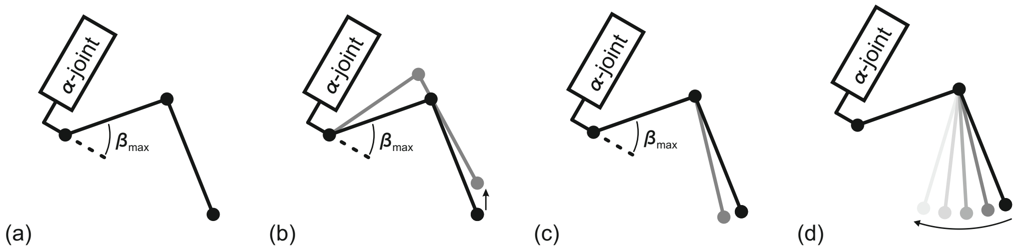


Figure 5.22.: Illustration of the problem regarding simple clipping of joint angles in case the β-joint reaches its limit. (a) shows the leg in a configuration in which the β-joint is already at its limit. In (b), the desired new leg configuration is shown in gray for a foot point that is shifted upwards. (c) Clipping the β-joint at its angular limit but leaving the γ-joint at the angle defined by the desired configuration shown in (b). The foot position is shifted inwards. (d) Movement of the foot position over the course of the next iterations using the clipping method. Although the foot position is supposed to be lifted, it moves continuously inwards until it would reach the limit of the γ-joint.

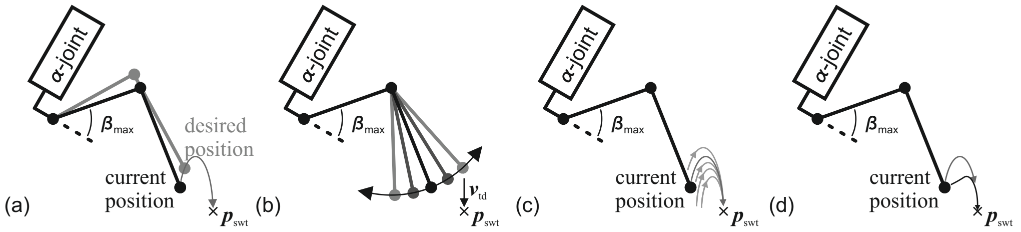


Figure 5.23.: The case already depicted in fig. 5.22 is addressed using a different strategy for the avoidance of the joint angle restriction. (a) shows the initial situation in which the leg tip is supposed to follow an ellipsoidal trajectory from the *current positions* towards the goal position $\boldsymbol{p}_{\text{swt}}$. (b) shows possible positions of the leg tip by moving only the γ-joint. (c) shows a family of different possible trajectories for the movement to the target position. (d) shows the actual swing trajectory in black and the originally desired trajectory in grey. The swing trajectory is obtained by rotating only the γ-joint towards the target position (with $\beta = \beta_{\text{max}}$) until the vector field - based trajectory generation yields a desired position with $\beta < \beta_{\text{max}}$. At this point, the vector field is followed again, thus lowering the leg towards the target position.

A solution of this problem is depicted in fig. 5.23. Using the unrestricted joints, the leg tip is moved closer to the line defined by $\boldsymbol{p}_{\text{swt}} + \delta \boldsymbol{v}_{\text{td}}$, therefore the line defined by the target position and the target velocity vector. Since the scope of action depends on the number of unrestricted joints, in the following, situations of one and two restricted joints are handled separately.

If only *one joint* is restricted at position $\boldsymbol{p}$, using the velocity vectors $\boldsymbol{v}_n$ and $\boldsymbol{v}_m$ (the respective columns n and m of the forward kinematics Jacobian matrix) for movements of the remaining free joints n and m, an intermediate target position $\boldsymbol{p}_{\text{interm}}$ can be found that respects the joint angle restrictions:

$$\begin{aligned} \boldsymbol{p}_{\text{interm}} &= \boldsymbol{p}_{\text{target}} + \delta\, \boldsymbol{v}_{\text{td}} \\ &= \boldsymbol{p} + \lambda_n \cdot \boldsymbol{v}_n + \lambda_m \cdot \boldsymbol{v}_m \end{aligned} \tag{5.62}$$

$\boldsymbol{p}_{\text{interm}}$ represents a position that respects the local restriction of a joint and that lies on the line defined by the target direction vector $\boldsymbol{v}_{\text{td}}$. To obtain a solution, eq. (5.62) must be solved for δ:

$$\delta = \frac{\Big(\boldsymbol{v}_m \times (\boldsymbol{p}_{\text{swt}} - \boldsymbol{p})\Big) \cdot \boldsymbol{v}_n}{(\boldsymbol{v}_m \times \boldsymbol{v}_n) \cdot \boldsymbol{v}_{\text{td}}} \tag{5.63}$$

If *two joints* are at their limits, the remaining free joint must be used to move the leg tip closer to the target. An intermediate target position can be constructed by

$$\begin{aligned} \boldsymbol{p}_{\text{interm}} &= \boldsymbol{p}_{\text{target}} + \delta\, \boldsymbol{v}_{\text{td}} + \varepsilon\, (\boldsymbol{v}_n \times \boldsymbol{v}_{\text{td}}) \\ &= \boldsymbol{p} + \lambda_n \cdot \boldsymbol{v}_n \quad . \end{aligned} \tag{5.64}$$

Solved for λ_n, this yields:

$$\lambda_n = \frac{\Big((\boldsymbol{p}_{\text{swt}} - \boldsymbol{p}) \times \boldsymbol{v}_{\text{td}}\Big) \cdot (\boldsymbol{v}_n \times \boldsymbol{v}_{\text{td}})}{\|\boldsymbol{v}_n \times \boldsymbol{v}_{\text{td}}\|^2} \tag{5.65}$$

The goal position for the next iteration of the walking controller is computed by

$$\boldsymbol{p}(t_1) = \frac{\boldsymbol{p}_{\text{interm}} - \boldsymbol{p}}{\|\boldsymbol{p}_{\text{interm}} - \boldsymbol{p}\|} \times s \tag{5.66}$$

with the desired speed s of the leg tip.

In combination with the evasion of the other restrictions (singularity, leg tip/segment, etc.), this approach might result in deadlocks. For example, if the desired leg position is located outside the leg tip distance unrestrictedness (therefore, too close to an neighboring leg), the desired position would be moved away from the adjacent leg. Due to this shift, one of the joint angle limits might be exceeded, which leads to the application of the correction mechanism described above. However, this might induce a shift of the desired position which moves it again out of the leg tip distance unrestrictedness. Although these cases are rare, they usually prohibit further locomotion since the affected leg does not reestablish ground contact. As countermeasure, in these cases, the target position $\boldsymbol{p}_{\mathrm{swt}}$ is successively moved towards the home position during multiple iterations of the walking controller until the leg tip reaches the target. Usually, by this measure, the problem is solved. However, an alternative solution for the generation of swing trajectories that directly incorporates the unrestrictednesses would be preferable.

5.5. The Stability Module

In comparison to robots most insects have the ability to cling to the ground using various techniques (Bässler 1983; Bullock et al. 2008). This, however, works only if the ground is sufficiently solid—on loose sand they cannot hold to the ground. But even if the insect looses contact, due to the small drop height the impact will not be impairing. Therefore, insects do not have to make sure at all times that static stability is maintained. The typical robot, however, must avoid crashes at all costs. Since only few walking robots are equipped with adhesive pads or claws (for counterexamples see Murphy et al. 2011; Parness et al. 2013), they must ensure dynamic or static stability irrespective of the substrate. As rather slow walker, Hector relies on maintenance of static stability. Therefore, a dedicated module is required to supervise the stability of the system. To quantify the stability, a stability measure is used that estimates the energy that would be required to let the robot tumble (see section 5.5.1). Based on this measure, during walking, the stability module interacts with the other modules of the walking controller, e.g., by preventing leg liftoffs that might result in instability (see section 5.5.2).

5.5.1. Measure for the Evaluation of Static Stability

Different measures of stability (stability margins) have been proposed to quantify the tendency of a system to topple over. The first stability margin was proposed by McGhee and Frank (1968) for a machine with constant velocity walking on flat, even terrain. In this concept, the stability margin equals the shortest distance between the vertical projection (along the vector of gravity) of the COM to any point of the support polygon. Therefore, a big distance between the projection of the COM and the support polygon implies a high stability. For the application on a robot that will change speed and direction during walking and furthermore is supposed to walk in rough terrain this stability measure is not adequate since it does not take the vertical position of the COM into account. The higher the COM is located above the support polygon, the more it resembles an inverted pendulum. Therefore, even slight disturbances can be sufficient to destabilize the robot. This dependence of the stability on the relative location of the COM is considered in the ESM (Messuri 1985, see below), which represents the minimum energy that is required to let a robot tilt around one of the lines spanning the support polygon. Although this stability measure was developed for stiff robots, due to the lack of an adequate stability measure that considers compliance, it will be used for the control of Hector. In the next paragraphs, first, the concepts of the ESM will be described, followed by a discussion of the limits of this stability margin for the application in a compliant system.

stiff robot (ESM) As mentioned, the ESM (Energy Stability Margin) represents the minimum energy that is needed to let a robot tilt around one of the support polygon lines. In fig. 5.24, an exemplary system is depicted that consists of the two points $\boldsymbol{p}_1$

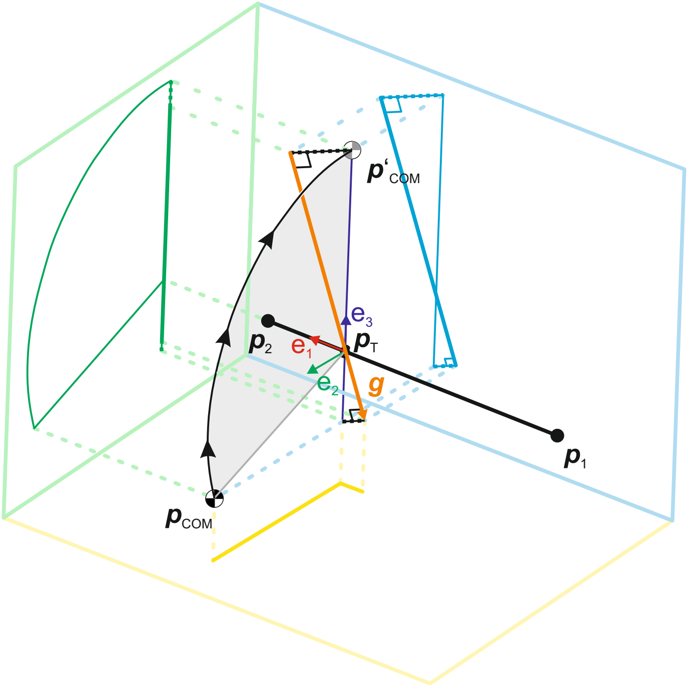


Figure 5.24.: Depiction of the computation of the stability margin. The points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ correspond to two foot points that span the support polygon. $\boldsymbol{p}_{\mathrm{COM}}$ is the actual position of the COM, for which the stability margin should be computed. $\boldsymbol{p}'_{\mathrm{COM}}$ represents the virtual position of the COM at the highest point during a rotation about the line between $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. The stability margin is proportional to the energy required to lift the COM from $\boldsymbol{p}_{\mathrm{COM}}$ to $\boldsymbol{p}'_{\mathrm{COM}}$.

and $\boldsymbol{p}_2$ of the support polygon and the COM that is located at $\boldsymbol{p}_{\mathrm{COM}}$. The minimal energy that is required to turn the mass around the axis $\overline{\boldsymbol{p}_1\boldsymbol{p}_2}$ equals the difference of the potential energy between the current state (mass at position $\boldsymbol{p}_{\mathrm{COM}}$) and the maximal potential energy the body could reach while the mass is rotated around the axis. The point at which the highest potential energy is reached, will be denoted $\boldsymbol{p}'_{\mathrm{COM}}$. To compute the position of $\boldsymbol{p}'_{\mathrm{COM}}$, the vector $\boldsymbol{e}_3$ (see fig. 5.24) must be determined that is part of an orthogonal system of unit vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$. $\boldsymbol{e}_3$ lies in the plane spanned by $\boldsymbol{e}_1 = (\boldsymbol{p}_2 - \boldsymbol{p}_1)/\|\boldsymbol{p}_2 - \boldsymbol{p}_1\|$ and the vector of gravity $\boldsymbol{e}_g = \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}$. It can be constructed by

$$\boldsymbol{e}_3 = \boldsymbol{e}_1 \times \boldsymbol{e}_2 \quad \text{with} \tag{5.67}$$

$$\boldsymbol{e}_2 = \frac{-\boldsymbol{e}_g \times \boldsymbol{e}_1}{|\boldsymbol{e}_g \times \boldsymbol{e}_1|} \quad . \tag{5.68}$$

The closest point on the axis $\overline{\boldsymbol{p}_1\boldsymbol{p}_2}$ around which the COM is revolved, is called $\boldsymbol{p}_\mathrm{T}$. It can be computed by an orthogonal projection of $\boldsymbol{p}_\mathrm{COM}$ onto $\overline{\boldsymbol{p}_1\boldsymbol{p}_2}$:

$$\boldsymbol{p}_\mathrm{T} = \frac{(\boldsymbol{p}_\mathrm{COM} - \boldsymbol{p}_1) \cdot (\boldsymbol{p}_2 - \boldsymbol{p}_1)}{\|\boldsymbol{p}_2 - \boldsymbol{p}_1\|}(\boldsymbol{p}_2 - \boldsymbol{p}_1) + \boldsymbol{p}_1 \tag{5.69}$$

The distance between $\boldsymbol{p}_\mathrm{T}$ and $\boldsymbol{p}_\mathrm{COM}$ is $d = \|\boldsymbol{p}_\mathrm{COM} - \boldsymbol{p}_\mathrm{T}\|$. Thus, the position of the point of highest potential energy is $\boldsymbol{p}'_\mathrm{COM} = \boldsymbol{p}_\mathrm{T} + \boldsymbol{e}_3 \cdot d$. The difference of the potential energy between the state in which the center of gravity is at $\boldsymbol{p}_\mathrm{COM}$ and the state where it is at $\boldsymbol{p}'_\mathrm{COM}$ is

$$\Delta E = -(\boldsymbol{p}'_\mathrm{COM} - \boldsymbol{p}_\mathrm{COM}) \cdot \boldsymbol{g}\, m \quad . \tag{5.70}$$

This is the minimal energy that is needed to let the robot topple over the connection $\overline{\boldsymbol{p}_1\boldsymbol{p}_2}$. As mentioned, the ESM is defined as the minimal energy that is needed to let the robot tumble about any of the support polygon lines:

$$S_\mathrm{ESM} = \min\left(\{\Delta E_i\}_{i=1\ldots n}\right) \tag{5.71}$$

with n being the number of lines of the support polygon and E_i being the energy that is needed to tilt the center of gravity around a particular line.

For static situations, when the robot is rigidly standing, $S_\mathrm{ESM} \geq 0$ denotes a stable and $S_\mathrm{ESM} < 0$ an instable posture. Although in principle only stable and instable situations must be distinguished, the stability margin is usually interpreted as a continuous measure of stability. This is only sensible if additional, unforeseeable factors must be taken into account. Therefore, a high stability margin means that the robot is more likely to maintain stability without intervention after an impact (e.g., a researcher kicking the robot). In general, the energy E_imp that is introduced into the robotic system due to the impact must be smaller than S_ESM to guarantee stability. Besides external disturbances, self-induced accelerations of the robot must be considered. Therefore, if the robot needs to stop immediately, e.g., due to an obstacle, the inertia of the robot might be sufficient to overcome S_ESM, letting the robot topple over. Thus, if sudden stops must be expected, the kinetic energy of the robot E_k should be considered as the lower boundary of stability ($E_\mathrm{k} < S_\mathrm{ESM}$).

compliant robot (not solved) The computation of the energies required to tilt the robot about the lines of the support polygon is based on the assumption of a rigid robot. Hector, however, is equipped with compliant joints that would result in lowering of the COM during the tilting motion. This poses a problem that has not yet been solved. In fig. 5.25, the subsiding is depicted for standing and tilted postures. The potential energy is reduced in both cases for the compliant robot since the COM is lowered compared to the stiff robot. As a consequence, the potential energies of the springs in the joints would need to be considered as well to quantify the change of potential energy due to the tilting:

$$\Delta E_\mathrm{comp} = -(\boldsymbol{p}'_\mathrm{COM} - \boldsymbol{p}_\mathrm{COM}) \cdot \boldsymbol{g}\, m + \left(\sum_{i=1}^{N} E'_{\mathrm{spr},i} - \sum_{i=1}^{N} E_{\mathrm{spr},i}\right) \tag{5.72}$$

$E_{\text{spr},i}$ and $E'_{\text{spr},i}$ represent the potential energy of the i-th spring in the standing and the tilted postures, respectively.

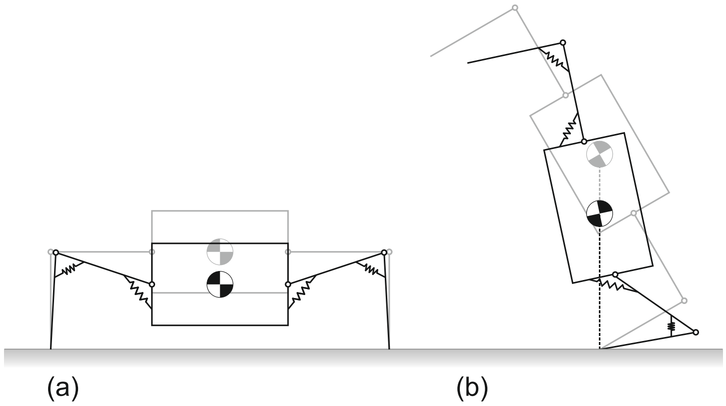


Figure 5.25.: Schematic comparison of stiff (gray) and compliant (black) robots for two different postures. (a) shows robots during normal standing. The central body of the compliant robot is lowered due to the weight of the central body which loads the springs. (b) shows the robots in a critical posture, in which the COM is situated vertically above the foot point. Again, for the compliant robot the position of the COM is lowered as compared to the stiff robot and the springs are loaded.

For the compliant joints used in HECTOR, the potential energy due to a torsion θ of the elastomer coupling can be approximated based on eq. (2.1), which results in:

$$E_{spr}(\theta) = \int_{\omega=0}^{\theta} \tau \mathrm{d}\omega \tag{5.73}$$

$$= 6162.59\theta^6 + 32.025\theta^2 \tag{5.74}$$

With the sensor data of the joint drives, the total potential energy stored within the elastomer couplings can be estimated. In the tilted posture all but two legs can be assumed to be unloaded. Therefore, the energy previously stored in the couplings may have facilitated the tilting of the robot. The missing values to compute ΔE_{comp} (according to eq. (5.72)) are the position of the COM in the tilted posture and the corresponding torsions of the elastomer couplings in the loaded legs. The estimation of these values, however, would require extensive simulations. Nagy (1992) suggested a stability measure that incorporates the compliance of the terrain. Since it assumes a single spring-like compliance between the leg tips and the ground it cannot adequately represent the distributed compliance in the legs of HECTOR. A stability margin to quantify the static stability of compliant robots is therefore lacking.

For this reason an experimentally determined minimal threshold ΔE_{min} is used in combination with the ESM for the control of HECTOR. The stability margin is computed under the assumption of stiff legs (see eq. (5.70)), however, based on the actual, subsided posture of the compliant robot. This measure is used for the computation of the stability unrestrictedness (see section 5.2.1) as well as for the mechanisms that will be introduced in section 5.5.2. During the performed experiments the robot remained stable insofar that it did not topple. Still, due to the simplifications general stability cannot be guaranteed for the compliant system.

5.5.2. Interaction During Walking

In the original concept of WALKNET, the legs inevitably switched to swing phase if they reached their PEP. However, as shown in chapter 4, this might compromise the stability of the robot. To facilitate stable walking the stability module interacts with the walking controller in two ways, the *pre-liftoff stability check* and the *stability maintenance*.

pre-liftoff stability check Before the transition of a leg from stance to swing phase is performed, the prospective posture is checked by the stability module. For two cases, this is depicted in fig. 5.26. Assuming the leg would be released from stance phase, the stability margin is computed under consideration of the remaining stancing legs. If this prospective stability margin lies below a given threshold the leg controller is forced to remain in stance mode. Thus, as long as the leg is essentially required to maintain stability, it is appointed to stance. This mechanism can be regarded as an additional, artificial coordination influence since it prolongs the stance phase of individual legs.

Without the intervention of the pre-liftoff stability check, the legs would change to swing phase at the latest when they reach the end of the stance unrestrictedness. However, since this mechanism keeps the legs in stance phase, they might leave their workspace. To circumvent this, the velocity of the robot must be reduced if a leg gets close to its workspace limit. This mechanism is explained in more detail in section 5.6.

stability maintenance In addition to the pre-liftoff stability check, the stability margin is continuously checked to ensure stability. If the COM moves towards one of the lines spanning the support polygon, the movement must be stopped before the stability threshold is underrun. As previously mentioned, a gradual deceleration is preferable. The mechanism that predicts potential problems in advance and then reduces the speed of the robot is described in section 5.6.

5.6. Speed-Reducing Measures

As mentioned before, different constraints must be considered during movement. For example, the legs must remain within their workspaces and the static stability of the robot must be maintained. Due to interactions of various mechanisms, however, problematic situations might arise. In these cases the walking velocity of the robot is gradually reduced.

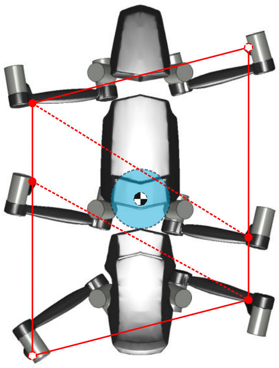


Figure 5.26.: Example for the pre-liftoff stability check. The positions of the leg tips are marked by dots, the support polygon is represented by red lines. Based on the coordination influences, the front right and the hind left leg (open red circles) are supposed to switch to swing phase. If they were allowed to switch to swing phase, the support polygon would be reduced to the central region, bordered by the dashed lines. For visualization, the minimum stability threshold is depicted as blue area around the COM. In the case of liftoff of the front right leg the minimal distance between the COM and the closest line of the support polygon would be smaller than the stability margin threshold. Therefore, the switch to swing phase would be prohibited for this leg. However, a liftoff of the hind left leg would maintain the required stability and would therefore be allowed.

For example, as described in section 4.1, in the original concept of Walknet the leg controllers switch to swing phase as soon as the leg tip crosses the PEP. Applying this concept to the unrestrictedness-based coordination influences presented in section 5.2.3, the legs would switch to swing phase at the latest when they reach the limit of their workspaces. However, due to the previously described pre-liftoff stability check (see section 5.5.2), a leg that reaches the end of its workspace might be forced to remain in stance phase if it is required to maintain static stability. To prevent the leg from leaving its workspace, the movement of the whole robot must be stopped. Since an abrupt stop might be harmful for the leg joint actuators (especially if they do not feature an inherent compliance), the robot must be gradually decelerated. This, however, necessitates the early detection of potential problems. To forecast problems, as done before, a constant walking direction is assumed. For for a given *prediction period* (although conceptually similar, not equal to the *predetermination periods* of the selector module), the trajectories of the stancing legs are extrapolated. Within this period, problematic postures must be detected. The less time remains between the current and the problematic state, the stronger must the robot be decelerated. The following three situations are checked for each future iteration of the walking controller within the duration of the prediction period:

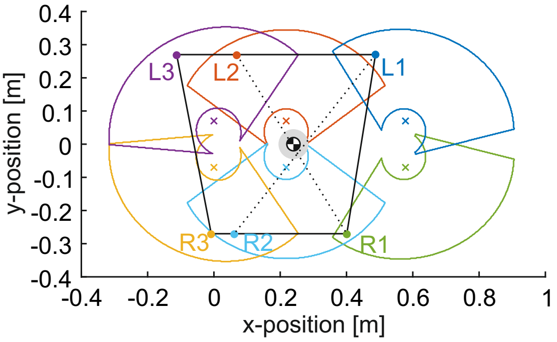


Figure 5.27.: Exemplary illustration of a state that must be prevented by the stuck check. For each leg the corresponding workspace is marked. The front legs are close to the limits of their workspaces, thus triggering a switch to swing phase, whereas the middle legs are still sufficiently afar from their workspace limits to remain in stance phase. However, due to the pre-liftoff stability check, the front legs are not allowed to lift since they are both essential for the support of the COM.

stability check During each iteration the estimated leg tip positions of all currently stancing legs are computed. Based on these positions the stability margin is evaluated, upon which the decision is made whether the particular posture is sufficiently stable or not. It is only considered whether or not the margin lies above or below the predefined threshold. Legs that are in swing phase will not be considered—even if they are expected to reach ground contact in the near future as the ground contact cannot be guaranteed in advance.

workspace check To assure free movement within the workspace of each leg, the unrestrictedness of each leg tip position during the *prediction period* is computed. If one of the legs would leave its workspace during the prediction period, the velocity must be reduced to impede this.

stuck check Using only the described mechanisms, situations may occur during which neither of the front legs is allowed to liftoff since this would endanger the stability (see fig. 5.27). Although the robot is statically stable since both front legs are in ground contact, no combination of front and diagonally opposing middle leg is able to support the COM sufficiently to allow a liftoff of the other front leg. If the middle legs are still far enough from the limits of their workspace, they will not lift off and swing to a position that might enable the front legs to lift. Thus, the robot might get stuck.

To prohibit situations like these during forward walking, the two stability margins for the diagonal connections of the front to the respective, contralateral middle legs are calculated. If one of the stability margins is already below the threshold and the other is about to fall below the threshold as well, the speed of the robot is reduced such that the second pair of front and middle legs will never cross the threshold. This guarantees that at least one of the front legs can be lifted. For backward walking the same check is performed for diagonal pairs of hind and middle legs.

To decelerate the robot, the heading vectors $\boldsymbol{h}_0$ and $\boldsymbol{h}_1$ as used within the direction controller (see section 5.3.2) are scaled by a speed factor $f_{\text{sp}} \in [0, 1]$. Thus, the walking velocity of the robot can be gradually reduced. If a problematic situation that led to a speed reduction is resolved (e.g., if the stability is increased by a leg that switched from swing to stance phase) f_{sp} is gradually increased up to 1.

5.7. Beyond Biology — Further Coordination Influences

With the previously described checks and the consequent reduction of speed it can be assured that the robot will remain operational. However, conditions might occur during which the robot is decelerated to a complete stop without a chance to restore regular coordination. For example, if the stuck check stops the robot with the front legs in a position that would allow them—in principle—to start a swing phase but too far away from their workspace limits to actually trigger the stance-swing transition, the robot will remain in this posture indefinitely. To resolve these states, artificial coordination influences have been introduced that will trigger stance-swing-switches in all or only selected legs.

This manipulation of the coordination is realized by modification of the predetermination period Δt_{pd} (see section 5.2.3). Therefore, as done for the bioinspired coordination influences, additional factors are multiplied:

$$\Delta t_{\text{pd}} = \Delta t_{\text{thr}} \prod_{i=1}^{N} I_i \cdot \prod_{i=1}^{N} T_i \tag{5.75}$$

with the bioinspired coordination influences I_i and the additional, technical influences T_i.

restricted-neighbor-induced swing If a switch to swing is prohibited by the pre-liftoff stability check although the leg is close to the limits of its workspace, the walking speed of the robot is gradually reduced (see section 5.6). Ideally, the problematic situation is resolved on its own, e.g., by a swinging leg that touches down and increases the stability sufficiently for a liftoff of the previously impeded leg. However, if no other leg is in swing phase (or is triggered to start swing until the end of the deceleration), the robot would come to a halt without a way to resolve this deadlock. The restricted leg is indispensable for stability maintenance mainly because its neighboring legs are at positions that do not suffice to support the robot.

If such a problem is encountered during the predetermination period, a switch to swing of the neighboring legs is facilitated. As depicted in fig. 5.28, for this purpose, T_1

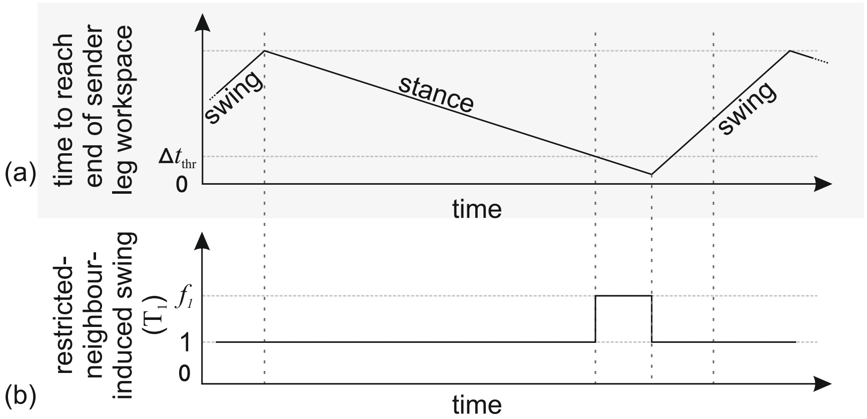

(a)
time to reach end of sender leg workspace
Δt_thr
0
swing
stance
swing
time
(b)
restricted-neighbour-induced swing (T1)
f_l
1
0
time

Figure 5.28.: Illustration of the *restricted-neighbor-induced swing* mechanism. Whenever the sending leg (a) gets close to the limit of its workspace a switch to swing of the neighboring leg (b) is facilitated by increasing T_1.

is increased to a factor of f_1, thus increasing the predetermination period Δt_{pd} (according to eq. (5.75)) of the neighboring legs. At the end of the thereby induced swing phases, the neighboring legs might establish ground contact at positions that are better suited to support the robot, thus enabling the previously restricted leg to switch to swing phase.

stuck-induced swing In situations, in which it can be expected that both front legs will get stuck—therefore, if the stuck check (see section 5.6) detects a problem—a switch to swing phase is facilitated for all middle and hind legs to resolve the problem. Since the target positions of the middle legs are selected such that they can support the robot together with the respective, contralateral front leg, the problem is likely to be resolved at the end of their swing phases. Since the selection of swing targets for the middle legs depends also on the positions of the hind legs, these are also encouraged to swing in order to increase the range of possible swing targets of the middle legs.

Correspondingly, if the stuck check detects a problematic situation for the hind legs, a switch to swing phase is facilitated for both front and middle legs. In fig. 5.29, the case for forward walking is depicted (considering only the front and middle legs). If the energies that would be required to tilt the robot about the lines L1/R2 and R1/L2 fall below a threshold of ΔE_{thr}, a switch to swing phase of both middle and hind legs is facilitated by increasing the influence of T_2.

deadlock-resolving influence If the biological and the two previous technical coordination influences are not sufficient to resolve a deadlock, all legs are facilitated to switch to swing phase—irregardless of their current positions and the cause of the deadlock. As indicator for such a situation, the speed factor f_{sp} (see section 5.6) is compared to a threshold value f_{thr}. If the speed factor falls below this threshold, the tendency of

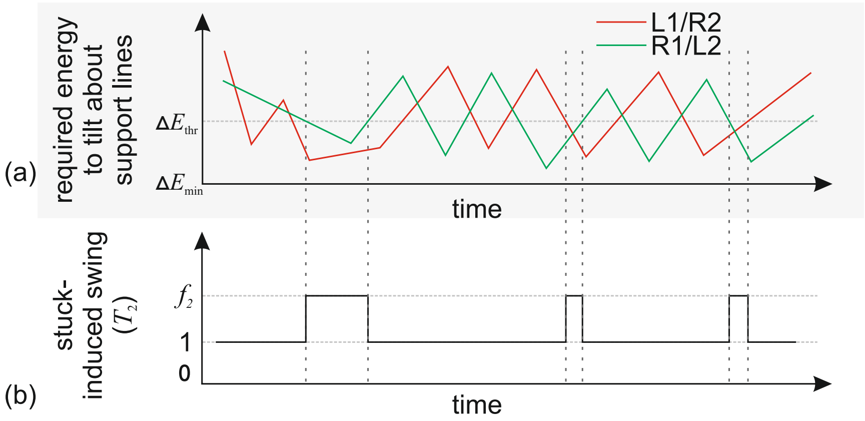


Figure 5.29.: Illustration of the *stuck-induced swing* mechanism. If the energies that would be required to tilt the robot about the diagonal lines connecting the front with the contralateral middle legs (left front leg (L1) to right middle leg (R2) and right front leg (R1) to left middle leg (L2)) both drop below a defined threshold ΔE_{thr} (a), a switch from stance to swing is facilitated in both middle and hind legs (b).

the legs to switch to swing is gradually increased, as depicted in fig. 5.30. Although this mechanism encourages all legs to start swing phases, due to the pre-liftoff stability check, instabilities are prohibited.

Since the selection of a swing target depends on the walking speed (see section 5.4.1), the slower the robot moves, the closer is the swing target to the home position. Therefore, in the case of a deadlock that leads to a stop of the robot, the legs will swing to their home positions until the deadlock is resolved.

5.8. Sequence of Operation

Due to the abilities and requirements of the robot that differ from those of the stick insect (listed in the introduction of this section), the bioinspired walking controller was modified to comply. The most relevant change is the extension of the bioinspired concept to omnidirectional walking using an unrestrictedness-based workspace definition. This allows for an adaption of the swing and stance trajectories to the desired direction of walking. Figure 5.31 shows the sequence of subtasks performed during each iteration of the controller. Each control cycle starts with the retrieval of data from leg joint actuators and additional sensors. Based on this data, newly established ground contacts of the swinging legs are identified and the corresponding leg controllers are switched to stance phase (see section 5.4.3). On average, this advanced processing of the swinging legs and the corresponding transitions of the leg controllers to stance phase increases the number of stancing legs and therefore the stability margin. This is beneficial in the later steps

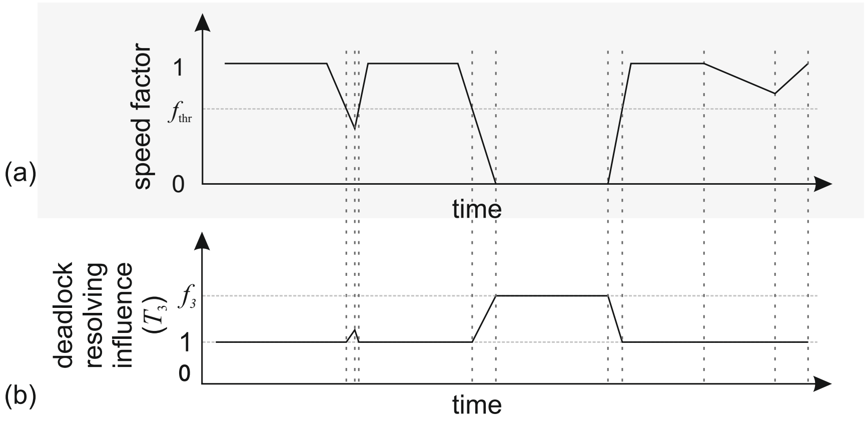


Figure 5.30.: Illustration of the *deadlock-resolving influence* mechanism. If the speed factor (*actual speed=speed factor* × *desired speed*) drops below a defined threshold f_{thr} a switch to swing phase is facilitated in all legs. The lower the speed factor gets, the higher is the influence to the legs.

when, e.g., the *pre-liftoff stability* check decides about whether a leg is allowed to switch to swing phase. With a higher overall stability margin, this check is more likely to allow a liftoff.

In the next step, the external direction commands are processed. These might originate from user input via a controller or sensor input (e.g., visual stimulus (Meyer et al. 2016); see section 7.2). Based on the received commands, the body trajectory for the current controller iteration is computed (see section 5.3). Considering the desired body trajectory, the future leg trajectories are estimated under the assumption of a continuous movement. These estimated, future leg trajectories are used, e.g., for the predetermination of potentially problematic situations. For this purpose, the workspace, stuck and stability checks (see section 5.6) are performed on the prospective trajectories to ensure an unhindered operation. In case any of the checks fails, an appropriate counteraction is activated. Independent of the source of problem, the robot is slowed down. In case the speed is already reduced below a given threshold, the deadlock-resolving mechanism is triggered that arbitrarily activates swing phases in the legs to attain a posture that allows continuation of movement (see section 5.7). If none of these problems is anticipated the controller proceeds directly to the execution of the leg controllers. If, however, a potential problem was detected in a previous control cycle and the walking speed was reduced, it is gradually increased back to the desired speed.

Before the leg controllers are handled, the *sequence of legs* must be assessed for sequential execution. With a fixed sequence (e.g., front left, front right,. . . ,hind right leg) the leading legs would be preferred for stance-swing switches. Since legs that have been in stance for a long time are more likely to reach the limits of their workspace, the durations of their current stance phases are used as sorting criteria.

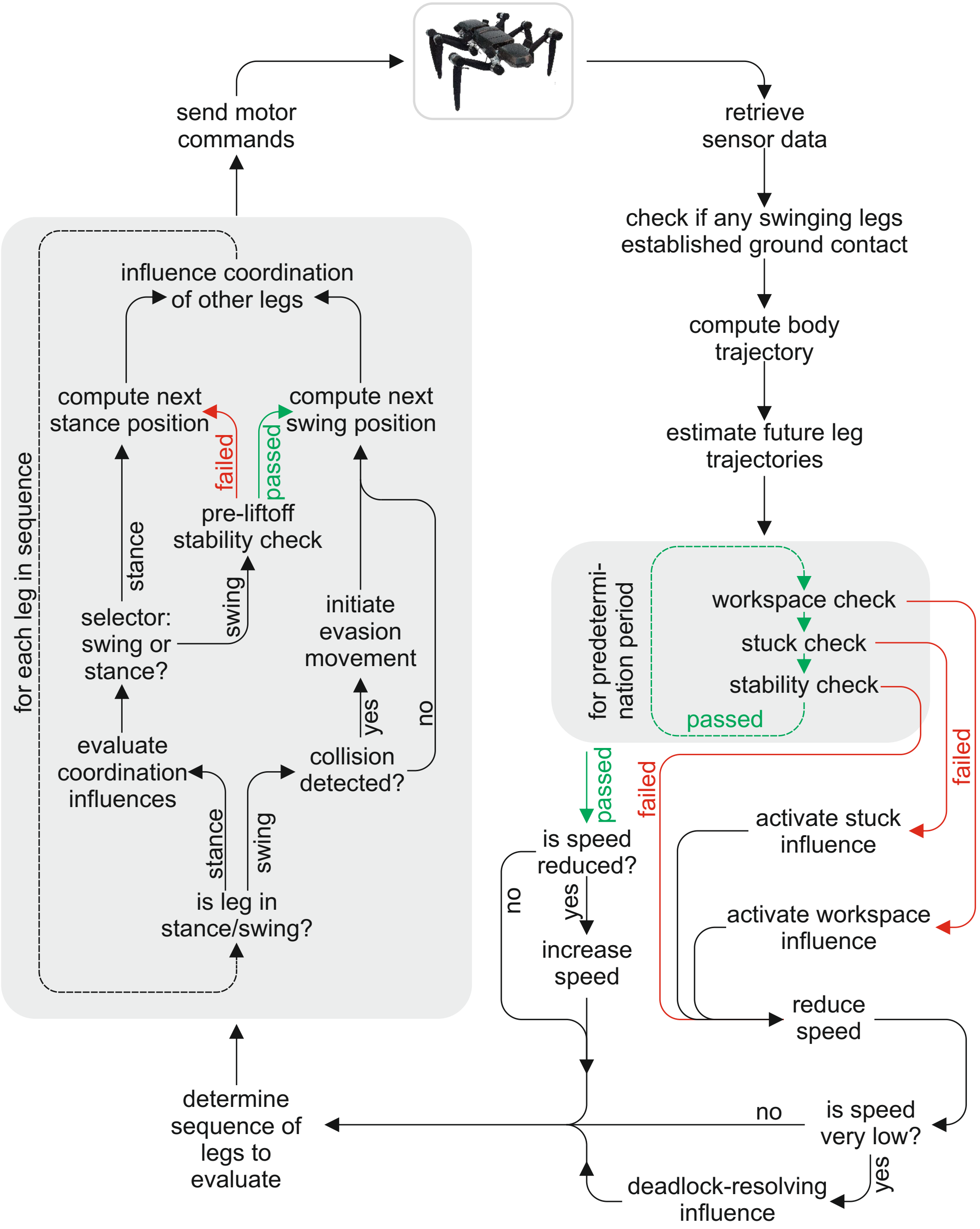

send motor commands
retrieve sensor data
check if any swinging legs established ground contact
compute body trajectory
estimate future leg trajectories
for predetermination period
workspace check
stuck check
stability check
passed
failed
passed
failed
activate stuck influence
activate workspace influence
reduce speed
is speed reduced?
yes
no
increase speed
is speed very low?
no
yes
deadlock-resolving influence
determine sequence of legs to evaluate
for each leg in sequence
influence coordination of other legs
compute next stance position
compute next swing position
failed
passed
pre-liftoff stability check
stance
swing
selector: swing or stance?
initiate evasion movement
yes
no
evaluate coordination influences
collision detected?
stance
swing
is leg in stance/swing?

Figure 5.31.: Flow chart showing the tasks performed during each cycle of the walking controller.

Within each *leg controller*, different paths exist for swinging and stancing legs. If the leg is in *swing phase*, it will remain in swing until it detects ground contact. As this ground contact detection is already performed earlier in the control cycle, the leg will inevitably remain in swing at this point of execution. Therefore, it must merely be tested if the leg collided with an obstacle. In this case, an evasion movement will be initiated (see section 5.4.3). At the end of the swing phase handling, the next position of the swing trajectory must be computed. Also, if the leg is in swing phase, it will influence the coordination of neighboring legs via influence 1 (see section 5.2.3).

For a leg in *stance phase*, the selector module (section 5.2) must decide whether the leg should remain in stance or if it should start to swing. This decision is based on the current position of the leg, the states of the neighboring legs and the outcomes of the predetermination checks that were performed earlier. In case the selector advocates the start of swing, the *pre-liftoff stability check* is performed to ensure stability. If the test is passed, the leg switches to swing state. If the leg fails the check, it will remain in stance phase. The final tasks in the handling of each stancing leg is the computation of the goal position for the next iteration of the walking controller and the calculation of the coordination influences for the neighboring legs.

This procedure is performed for each leg of the previously determined sequence. As last step during the control cycle the motor commands are sent to the joint drives to move the legs to the next goal positions on their stance/swing trajectories.

Summary

In chapter 4, Walknet, a direct implementation of a bioinspired walking controller, was tested in simulation for the application on Hector. Since the bioinspired controller does not include a dedicated module for monitoring of stability, the robot encounters occasional instabilities. For the application on the real robot, however, maintenance of static stability is a fundamental requirement. Therefore, a new controller was developed that combines the adaptability of the bioinspired concept with the technical requirements of the robot (e.g., static stability). Also, the new controller extends the leg coordination mechanisms that previously supported only forward walking to facilitate omnidirectional locomotion.

Structurally similar to Walknet, the new controller consists of multiple modules that are responsible for dedicated tasks. Whereas in Walknet, these modules have been implemented as ANNs, in the new framework, they have been explicitly formulated. However, the conceptual functionality of both approaches is similar.

As in Walknet, swing and stance modules are responsible for the generation of leg trajectories during swing and stance phases. The transitions between these two leg states are controlled by the *selector module*. To realize omnidirectional walking, the workspaces of the legs are defined based on multiple constraints, e.g., the angular limits of the joints. For the definition of these constraints, *unrestrictedness* functions have been defined. These functions constrain the workspaces of the legs. The decision, whether a leg should switch from stance to swing phase, is based on the estimation of the remaining time until this leg would reach the limit of its workspace. If the remaining time is bigger than a threshold, the leg remains in stance. If it is smaller, the leg switches to swing phase. By modifying the threshold the stance and swing phases of the legs can be coordinated.

For leg coordination, the first three coordination rules of Walknet were adapted to this new concept. The fourth coordination rule, which lets swinging legs target at their anterior neighboring leg, was implemented implicitly due to the process by which swing targets are selected.

The leg movements during *stance phase* are synchronized to reduce tension in the robot structure, thus lowering the energy consumption. Based on a concept by Schilling, Paskarbeit, Schmitz, et al. (2012), the movement of the real body can be controlled by virtually pulling an internal model of the robot in the desired direction. To enable the robot to walk in uneven terrain, also a stance height adaption is implemented that compromises between the desired stance heights of the legs. Since this is based on iterative evaluations of a virtual network of springs and dampers, oscillations due to numerical instabilities could occur. During the tests, however, these were not observed.

Compared to the generation of stance trajectories, *swing* trajectories are generated rather independently for each leg since swinging legs are not mechanically coupled via the ground. Still, the direction of walking and the postures of adjacent legs are considered. For example, during fast forward/backward walking, the legs will target at the foremost/hindmost position within their workspaces. Since the workspaces are restricted by the leg tip positions of neighboring legs, swinging legs will target at positions close to these. This corresponds to the targeting behavior (coordination rule 4) of the classical Walknet. The actual swing movements towards the target positions are controlled based on vector fields, which yield ellipsoidal trajectories. Since the technical restrictions of possible leg tip positions (e.g., due to the joint angle limits) are not incorporated in the setup of the vector field, the trajectory generator may command the leg to leave its workspace. To prohibit these transgressions, the trajectory is altered such that the leg tip remains within the workspace. These modifications, however, might interrupt the smooth swing movement. Therefore, a swing trajectory generation that inherently includes the workspace limits would be beneficial.

As mentioned, *static stability* of the robot must be preserved during walking to prevent tilting. To quantify the stability, the ESM (Energy Stability Margin) was selected, which represents the minimum energy that is required to rotate the robot about one of the lines spanning the support polygon. Since the distributed compliance in the legs (due to the elastomer couplings in the joint drives) cannot be incorporated into this stability margin, the computation is based on the assumption of a stiff robot. As a consequence, the static stability of the system cannot be guaranteed although during the performed experiments the robot did not topple. However, the development of an adequate stability margin that considers the compliance of the joints would be eligible.

As technical complement to the bioinspired coordination influences, further influences were introduced that facilitate an undisturbed locomotion. Together with a speed-reduction that is utilized whenever a potential problem is estimated, these additional coordination influences incorporate the technical requirements of the robot (e.g., static stability) into the bioinspired concept of Walknet.

The ability of this new concept to reproduce the adaptive gait patterns observed in stick insects is shown in chapter 6. In chapter 7, an implementation of the controller is tested on Hector in different walking scenarios.

6. Optimization of Walking and Relative Assessment Measures

Abstract: Although the introduced stability criteria facilitate statically stable walking, this does not necessarily result in recurring gait patterns as observed for the classic Walknet*. With irregular gait patterns, however, frequent speed reductions are more likely, e.g., if the swing phases of the hind legs are unsynchronized. Therefore, to facilitate unhindered operation, parameter sets for the walking controller must be identified that generate recurring gait patterns, ideally alike the classic* Walknet*. For automatic gait characterization, three specific assessment measures will be defined to rate the regularity of the emerging gaits. Using these measures, parameter sets can be identified to produce the continuous spectrum of gaits that has been identified in stick insects.*

In chapter 5, various mechanisms were presented that interact with each other to allow statically stable walking. For example, due to the measures presented in section 5.6, the walking speed is reduced to allow for a recoordination of the legs if a potential problem is detected (e.g., the reduction of the stability margin). As a consequence of these mechanisms, the bioinspired coordination influences do not affect the stability of the system as they did within the framework of the classical Walknet. In the new system even for a bad choice of parameters the stability will be maintained.[1] However, to prohibit instabilities the mechanisms of section 5.6 can only reduce the speed factor and thus the walking speed. The technical coordination rules (see section 5.7) are not designed to establish regular gait patterns but to resolve problematic situations. Therefore, in the new controller, the bioinspired coordination rules still serve the purpose to coordinate the legs such that a regular gait pattern emerges. This can help to reduce interventions in form of speed reductions. Therefore, whereas in Walknet the bioinspired coordination rules were responsible to maintain stability, in the new controller they help to maximize the walking speed. As a consequence, the average speed factor (ratio of actual over desired speed) produced by a set of coordination parameters can be used as a first indicator of the quality of coordination.

Using the average speed factor as sole quality measure would yield technically sufficient walking in the sense of continuous locomotion but it would likely miss the adaptability of gait patterns observed in walking stick insects. For example, by choosing a high value for coordination influence 2, whenever a leg touches down at the end of its swing phase neighboring legs would be obliged to start their swing phases. After a short interval of

[1] Within the abilities of the stability module (see section 5.5.1).

unstructured leg coordination this would lead to a speed-independent tripod gait. Since a fundamental concept of the bioinspired leg coordination is the constant, high swing velocity only the stance velocity is adapted to the walking speed. For slow walking with this speed-independent tripod this would therefore result in very small steps at a high frequency. Although this “fixed” gait would yield continuous locomotion it does not feature any of the advantages of the adaptive gait generation.

The goal of the walking optimization should therefore be to obtain a parameter set that yields truly adaptable gaits—meaning wave gait at low speeds, tetrapod at medium, and tripod gait at high speeds. As mentioned before, these idealized gaits represent only points in a continuum of gaits. However, the occurrence of these distinct gait patterns at different walking speeds indicates the adaptability of leg coordination and can be used as a quality measure for the evaluation of the walking controller. A characteristic of the mentioned gait patterns is the regularity in the sequence of the legs’ stance-swing-cycles. For example, during wave gait a metachronal wave of protractions travels from the back of the robot to the front. Therefore, the independent sequences of liftoffs on both sides of the robot start with the hind legs, followed by the middle and at last the front legs. Thus, measures of the regularity of the gaits can be used for the evaluation of walking quality.

6.1. Gait Regularity as Assessment Measure

Technically, it does not matter whether the order in which the legs are lifted is recurring. However, in many cases, the regularity of a gait will correlate with other advantageous features such as reduced oscillations of the main body and an overall higher speed factor. Eventually, also a reduction of the energy consumption could be observable in comparison to a fixed tri- or tetrapod gait since the number of legs that are in swing phase is reduced for slower walking speeds and therefore less swinging legs must be supported by the stancing legs.[2]

To measure the regularity of a gait, a table can be used that lists the likeliness of subsequent liftoffs for each combination of two legs. For this purpose the gait pattern must be analyzed regarding liftoff sequences. In the simplest case, only individual, successive liftoffs are considered as it can be found for slow walking. Therefore, gait patterns that show concurrent liftoffs of multiple legs, for which no sequence of individual liftoffs can be determined, are excluded for now. In this context, *concurrent liftoffs* means that switches from stance to swing phase occur for multiple legs during the same iteration of the walking controller. This, however, does not exclude gait patterns in which the swing phases of multiple legs overlap.

Each cell of this table represents a pair of successive swing initiations. The rows of the table denominate the legs that switched to swing phase and the columns denominate

[2] Since the α- and β-drives are located close to the body, their masses must be supported independently of the leg state. During the stance phase, however, the γ-drive is structurally supported by the tibia, which reduces the required energy compared to a swinging leg.

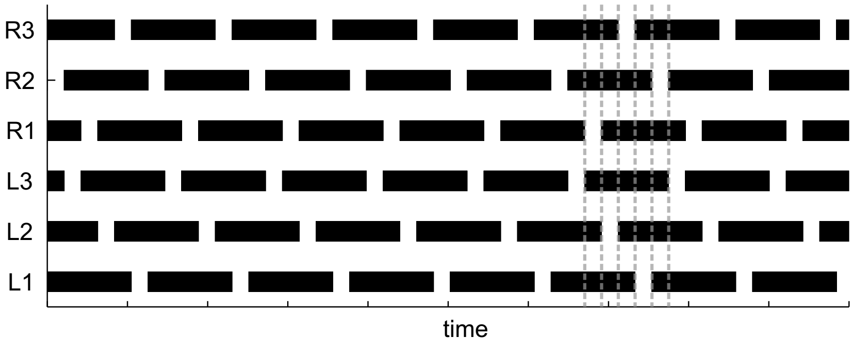


Figure 6.1.: Gait pattern showing an ideal wave gait. The black bars indicate stance phases. Each dashed vertical line indicates exemplarily the end of a stance phase. A characteristic of the ideal wave gait is the equidistance of these lines.

after prior	L1	R1	L2	R2	L3	R3
L1	0	0	0	7	0	0
R1	0	0	8	0	0	0
L2	0	0	0	0	0	8
R2	0	0	0	0	8	0
L3	0	8	0	0	0	0
R3	8	0	0	0	0	0

Table 6.1.: Switch table for the ideal wave gait depicted in fig. 6.1.

the legs that start their swing phase next. Therefore, each entry in the matrix assesses the number of times that liftoffs of the corresponding legs happened successively.

For the idealized wave gait shown in fig. 6.1, table 6.1 can be gained. Each row and each column has only one entry. Therefore, it is explicit which leg starts its swing phase after a certain leg started its swing phase. The sequence of swing phase initiations is always ...,L1,R2,L3,R1,L2,R3,.... For a perfect tripod, however, in which the triplets are lifted alternately with all legs of a triplet lifting in the same iteration the relationship is not distinct anymore. To deal with this problem, groups of legs can be considered rather than single legs. In the example of the perfect tripod, two groups consisting of three legs each can be identified that lift alternately. Here, the sequence of swing starts is ...,(L1,R2,L3),(R1,L2,R3),.... The corresponding switch table is listed in table 6.2.

For experimental data, e.g., of walking stick insects, this concept can be expanded to consider a certain time interval, the *concurrence interval*, within which legs must start to swing in order to consider them a liftoff group. As it was done for the ideal tripod, the sequence of liftoffs of the groups rather than individual legs is listed in the switch table. To group the single stance-swing transitions into liftoff groups, the

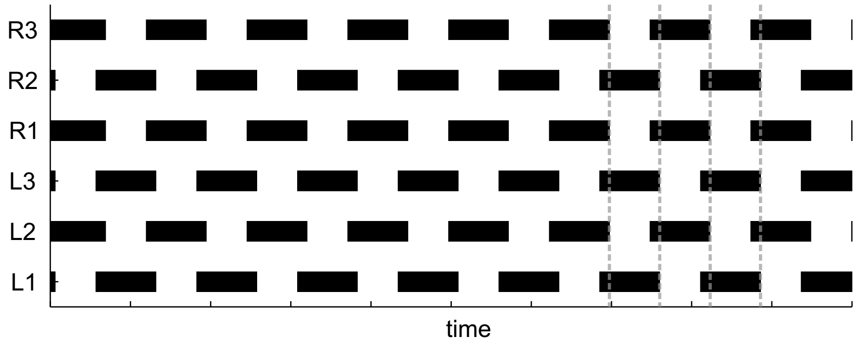


Figure 6.2.: Gait pattern showing an ideal tripod gait. The black bars indicate the stance phases. The dashed vertical lines mark the liftoffs of each leg triplet.

prior \ after	(L1,R2,L3)	(R1,L2,R3)
(L1,R2,L3)	0	8
(R1,L2,R3)	7	0

Table 6.2.: Switch table for the ideal tripod depicted in fig. 6.2.

gait pattern of all legs is progressively scanned for a given concurrence interval: In the first iteration, the algorithm searches the gait patterns for concurrent liftoffs of six legs. Under normal conditions, this case should never emerge since—without any legs in stance phase to support the body—the robot would obviously tumble. This kind of instability is supposed to be prevented by the pre-liftoff check (see section 5.5.2). However, if a sufficiently long concurrence interval is chosen, this case may be detected even though the swing phases do not overlap at all. Thus, the detection of six concurrent liftoffs indicates an inappropriately chosen concurrence interval. The same operation is performed during the next iterations of the grouping algorithm for concurrent liftoffs of five and four legs. As for groups of six legs, the detection of these liftoff groups indicates a too long concurrence interval. Meaningful results can be obtained during the next iterations of the algorithm, in which groups of three, two, and single leg liftoffs are detected. To prevent the assignment of a liftoff event to multiple liftoff groups, each liftoff can only be grouped once. Since the number of legs per liftoff group decreases with every iteration of the algorithm, bigger groups are therefore favored over smaller groups. Thus, given the chosen concurrence interval, the algorithm merges as many stance-swing transitions as possible in each liftoff group.

As explained, the grouping of leg liftoffs depends on the choice of the concurrence interval. Therefore, also the resulting switch tables and the regularity measures for tripod, tetrapod, and wave gait, which will be explained in the following, depend on the concurrence interval. To account for this, the gait patterns are iteratively processed for

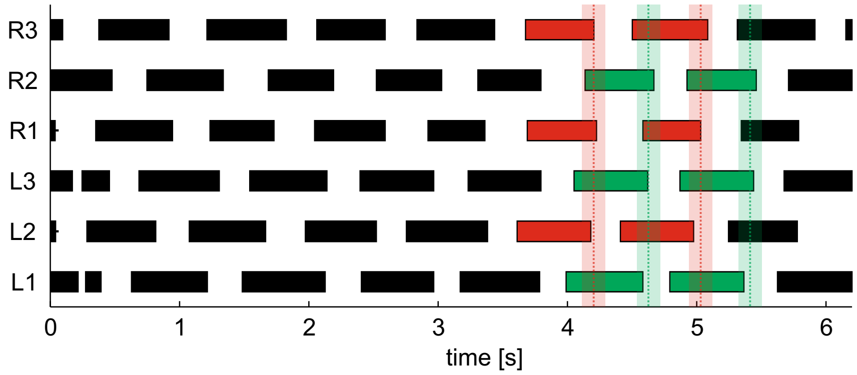


Figure 6.3.: Gait pattern based on experimentally obtained data from *Carausius morosus* (data was recorded by Theunissen, Hertrich, et al. (2014); "Animal12_110415_00_32"). The black bars indicate stance phases. Of the standard gaits, this sample shows the closest resemblance to a tripod gait. For two consecutive stance-swing cycles, the bars indicating the stance phases of each triplet have been colored. The light colored areas depict the concurrence intervals, for which the legs are grouped.

prior \ after	(L1,R2,L3)	(R1,L2,R3)	(L1,R1,L2,L3,R3)
(L1,R2,L3)	0	7	0
(R1,L2,R3)	6	0	0
(L1,R1,L2,L3,R3)	1	0	0

Table 6.3.: Switch table for the gait pattern depicted in fig. 6.3.

all possible concurrence intervals within a sensible range (e.g, 0 s up to 0.7 s; see fig. 6.4). For the classification of the gait, the maximum regularity measures (for tripod, tetrapod, and wave gait) are used that can be obtained for any concurrence interval within the given range.

In the following example that is based on a walking stick insect (the data was recorded by Theunissen, Hertrich, et al. (2014)), a concurrence interval of 0.18 s was chosen to group the lifting legs. The gait pattern is depicted in fig. 6.3. For two step cycles, the bars that indicate stance phases have been colored red and green to mark the affiliation to a liftoff group. The concurrence intervals are shown as vertical, semitransparent bars. The centers of these intervals are marked by dashed lines of the respective color. For this case, table 6.3 can be obtained. The grouping of (L1,R1,L2,L3,R3) is due to a rather unlikely situation at the beginning of the recording in the interval [0.22, 0.245] s (see fig. 6.3). According to the data, for a brief moment, only the right middle leg was in stance phase.

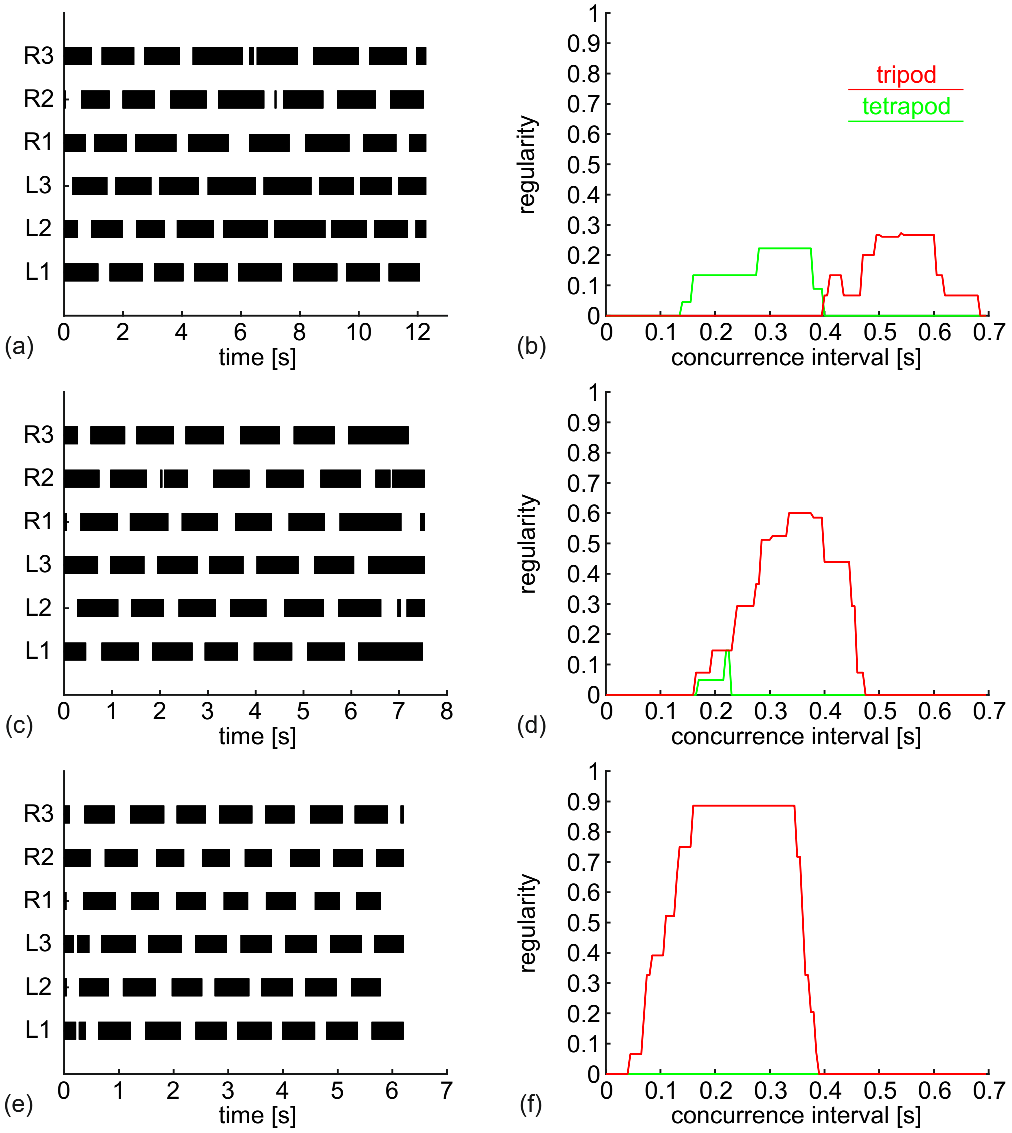


Figure 6.4.: (a,c,e) Gait patterns based on experimentally obtained data from *Carausius morosus* (data was recorded by Theunissen, Hertrich, et al. (2014)). The black bars indicate stance phases. (b,d,f) show the corresponding regularity measures for varying concurrence intervals. Only the regularity measures for tetrapod and tripod gaits are shown since in all trials, on average, more than one leg is in swing phase, thus failing the criterion for wave gait. For concurrence intervals over 0.7 s, no increase of the regularities could be detected since the number of legs in the liftoff groups increases above the limits of 2 legs for applicability of the tetrapod regularity measure and 3 legs for the tripod regularity measure.

As stated before, a characteristic of stick insect gait patterns is their speed dependency. For the evaluation of gaits at different speeds a classification would be beneficial as this allows the evaluation of the adaptiveness of the controller to different walking speeds. Based on the concept of the switch tables, measures for the different gaits can be deduced as shown below.

assessment measure for tripod regularity In tripod, the relevant two groups are g_1={L1,R2,L3} and g_2={R1,L2,R3}. As a measure of the tripodness of a gait, the ratio of the sum of the transitions between $g_1 \to g_2$ and $g_2 \to g_1$ over the sum of all transitions can be used:

$$r_{\text{tri}} = \frac{3 \cdot \left(n_{(g_1 \to g_2)} + n_{(g_2 \to g_1)}\right)}{\sum\limits_{i=0}^{N} \sum\limits_{j=0}^{N} \#g_j \cdot n_{(g_i \to g_j)}} \tag{6.1}$$

Therefore, transitions involving other groups (e.g., {L1,L3}→{R1,L2,R3}) are neglected. The factor of 3 in the numerator weighs the number of group switches by the number of legs in each group just as the $\#g_j$ does in the denominator. Since the quantization is based on a switch matrix, it requires the preparatory grouping of stance-swing transitions. During the ideal tripod depicted in fig. 6.2, the legs of each group lift off simultaneously. As a consequence, the length of the concurrence interval does not change the grouping and hence does not influence the regularity measure. For experimental data, e.g., the pattern depicted in fig. 6.3, the concurrence interval changes the grouping and therefore influences also the regularity measure. Based on three gait patterns of *Carausius morosus*, in fig. 6.4 (b,d,f), the tripod regularity is plotted in red for different concurrence intervals.

assessment measure for tetrapod regularity Correspondingly, the measure for tetrapodness can be defined under consideration of the two naturally occurring partitions of the legs: g_1={L1,R2}, g_2={L2,R3}, g_3={L3,R1}, and g_1={R1,L2}, g_2={R2,L3}, g_3={R3,L1} (Wilson 1966). In this work, all transitions from one group to another (within each partition) will be considered, therefore $\ldots g_1 \to g_2 \to g_3 \ldots$ as well as $\ldots g_1 \to g_3 \to g_2 \ldots$.

$$r_{\text{tetra}} = \frac{2 \cdot \max\left(\left\{ \left(n_{(g_1 \to g_2)} + n_{(g_2 \to g_3)} + n_{(g_3 \to g_1)}\right), \left(n_{(g_1 \to g_3)} + n_{(g_3 \to g_2)} + n_{(g_2 \to g_1)}\right) \right\}\right)}{\sum\limits_{i=0}^{N} \sum\limits_{j=0}^{N} \#g_j \cdot n_{(g_i \to g_j)}} \tag{6.2}$$

As for the tripod, also for the tetrapod regularity measure the stance-swing transitions must be grouped first. Therefore, it also depends on the concurrence interval. The dependency of the tetrapod regularity measure from the choice of the concurrence interval is plotted in fig. 6.4 (b,d,f) for three exemplary gait patterns in green.

assessment measure for wave gait regularity Following the usual definitions (Schilling, Hoinville, et al. 2013; Wosnitza, Bockemühl, et al. 2013), wave gait is the slowest of these idealized gaits. The main characteristic of wave gait is that on each side of the body an anteriorly directed metachronal wave of protractions can be observed (. . . R3→R2→R1. . . and . . . L3→L2→L1. . .).[3] Therefore, switch tables are compiled separately for each side of the robot, considering only the legs on the corresponding side. The wave gait regularity is then computed by

$$r_{\text{wave}} = \frac{n_{(L3\to L2)} + n_{(L2\to L1)} + n_{(L1\to L3)} + n_{(R3\to R2)} + n_{(R2\to R1)} + n_{(R1\to R3)}}{\sum\limits_{i=0}^{N}\sum\limits_{j=0}^{N} \#g_j \cdot n_{(g_i\to g_j)}} \quad . \tag{6.3}$$

Since the transitions between wave gait and tetrapod are fluent, as demarcation, on average only one leg is allowed in swing phase for a gait pattern to be classified as wave gait. As compared to the measures for tetrapod and tripod gaits, the wave gait regularity measure does not rely on liftoff groups. Therefore, it does not depend on the choice of the concurrence interval.

6.2. Selection of Walking Parameters for Optimization

Just as for the classical implementation of the coordination rules, the number of tunable parameters is also very large for the new coordination influences. Moreover, since the new implementation is supposed to support omnidirectional walking the strengths and directions of the coordination influences cannot be derived from studies on stick insects. Therefore, in principle the coordination influences might act between all directly neighboring legs with different strengths. Assuming symmetry of the coordination strengths between the left and the right side of the robot, seven parameters[4] could be tuned for each coordination influence. This sums up to 21 parameters for the coordination influences plus the additional time threshold of influence 3 for which also seven different values could be used. In addition, the technical aspects of the walking controller that influence coordination (described in section 5.7) must be harmonized with the other parameters. With the simulation running roughly in real-time and the multitude of different walking scenarios the controller is supposed to master, the required computational time for an optimization would considerably exceed the available resources.

Thus, the number of parameters must be reduced as much as admissible. Just as the insects, most of the time the robot will presumably walk straight forwards with only slight curves. Therefore, it seems reasonable to optimize the system for this specific

[3] The anteriorly directed metachronal waves of protractions are only characteristic for forwards walking. During backwards walking, also posteriorly directed waves of protraction have been reported (Graham and Epstein 1985). To evaluate the resemblance of robotic gaits during backwards walking this limitation can be omitted.

[4] Three parameters for influences between the contralateral leg pairs, two parameters for anteriorly directed influences between ipsilateral leg pairs and two parameters for ipsilateral, posteriorly directed influences.

scenario. However, the resultant set of controller parameters should also support the various other walking scenarios. For now, the directions of the coordination influences will be maintained as defined in the classical WALKNET concept (e.g., influence 1 acting solely forwards, see fig. 4.1). Different strengths between individual leg pairs will also not be considered in the optimization. However, since the workspace of the front, middle and hind legs differ, the time thresholds at which the legs would switch to swing phase (if no coordination influence acts on them) and the home positions will be tuned individually for each pair of contralateral legs. According to Dürr (2005), the coordination strengths for rules 2 and 3 of the classical WALKNET differ significantly between ipsilateral and contralateral leg pairs. Therefore, the ipsi- and contralateral strengths of influences 2 and 3 will be handled individually. In addition, the time threshold of influence 3 will be tuned globally. The strengths of the technical coordination mechanisms will also be set for all legs identically. The 14 parameters that will be tuned, the effective directions of influence, and the limits are given in table 6.4.

6.3. Optimization Process for Walking

An ideal set of parameters would combine the advantages of the bioinspired control with the requirements of the technical system. Therefore, it would reproduce the continuous spectrum of gaits that is observed in stick insects for different walking speeds, maintain the stability of the system such that the desired speed can be achieved at all times, and be independent of the starting posture and the walking direction. However, as already indicated in chapters 3 and 4, these demands might not be compatible. To get an overview of the obtainable results, in a first test, 80,000 different, randomly selected parameter sets were tested for a single starting posture (shown in fig. 6.7 (a)) and eight desired walking speeds (-0.05 m/s, 0.05 m/s, 0.10 m/s, 0.15 m/s, 0.20 m/s, 0.25 m/s, 0.30 m/s, 0.35 m/s). The application of an optimization algorithm such as simulated annealing or differential evolution was rejected since the definition of a quality function would be complex if it were to incorporate all requirements mentioned above and trade them off against each other. Also, optimization algorithms might prematurely concentrate on small regions of the parameter space, trying to find the optimum, whereas a global overview of the parameter space would be advantageous at this stage of optimization to estimate the limits of the control approach. Therefore, the undoubtedly more structured search strategies were precluded in favor of data generation based on random parameter sets combined with a subsequent offline analysis. For the slow walking speeds of ±0.05 m/s, the robot was commanded to walk straight ahead for 60 s to achieve a reasonable number of stance-swing cycles. For the faster walking speeds the test duration was limited to 30 s. To allow the robot to establish a temporally stable leg coordination the first 10 s were excluded from each run for the further analysis.

Since the home positions were among the optimization parameters that were chosen at random, prior to every test trial the basic reasonability of these positions were checked regarding three requirements: As *first requirement*, the home positions of anterior legs must be positioned in front of the home positions of posterior legs.

name	**direction** (if applicable)	**limits**
influence 1 ipsilateral	L2→L1, R2→R1 L3→L2, R3→R2	[0,1]
influence 2 ipsilateral	L2→L1, R2→R1 L3→L2, R3→R2	[1,5]
influence 2 contralateral	L1↔R1, L2↔R2 L3↔R3	[1,5]
influence 3 ipsilateral	L1→L2, R1→R2 L2→L3, R2→R3	[1,5]
influence 3 contralateral	L1↔R1, L2↔R2 L3↔R3	[1,5]
influence 3 time threshold		[0,5]

restricted-
neighbor-induced
swing
	L1→{R1,L2},L2→{L1,R2}
	L2→{L1,R2,L3},R2→{R1,L2,R3}
	L3→{L2,R3},R3→{R2,L3}
		[1,5]

stuck-induced
swing
	{L1/R1}→{L1,R1,L2,R2}
	{L3/R3}→{L2,R2,L3,R3}
		[1,5]

rel. x-position of home front legs [m]		[-0.3,0.3]
rel. x-position of home middle legs [m]		[-0.3,0.3]
rel. x-position of home hind legs [m]		[-0.3,0.3]
time threshold front legs [s]		[0,3]
time threshold middle legs [s]		[0,3]
time threshold hind legs [s]		[0,3]

Table 6.4.: List of parameters, effective directions for coordination influences, and corresponding limits that were used during the walking optimization.

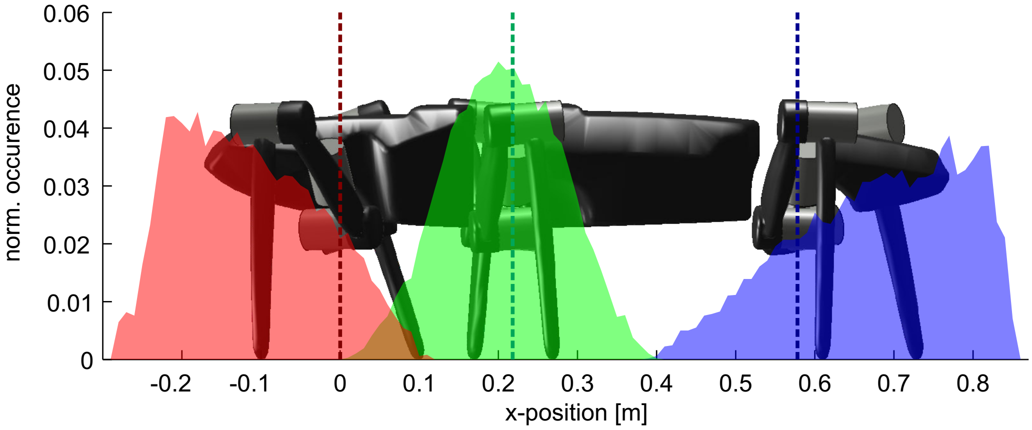


Figure 6.5.: Distribution of valid home positions for the three leg pairs. The distributions are shown as semi-transparent, color-filled plots with the x-positions of the leg onsets depicted by vertical, dashed lines of the respective color. Only those sets of home positions are considered that enable static stability and a minimum speed factor of 0.1 for all tested speeds.

The *second requirement* is that if the legs are positioned at their home positions, the minimum stability must be ensured to allow further movement. To test this case, the stability was evaluated for the postures with all legs at their respective home positions. If the stability margin (see section 5.5.1) was below the selected threshold of 2.6 J, the parameter set was neglected.

The *third requirement* demands that the average speed factor (the mean ratio between actual and desired speed) remains above 0.1. Therefore, during the test trials, if the average speed factor dropped below 0.1 for any of the desired speeds, the test was aborted and the parameter set discarded. This measure was taken to preclude improper data sets and to reduce the time required to obtain results for suitable parameter sets.

These three requirements therefore introduce a dependency in the choice of home positions of neighboring leg pairs. Figure 6.5 shows the distribution of valid home positions for the three leg pairs. The peaks of valid home positions for the front and hind legs are shifted by 0.2 m, and -0.2 m, respectively. The distribution of middle leg positions is comparably symmetrical with a peak at -0.05 m.

A broad overview of the results is shown in fig. 6.6 for different desired walking speeds. In all subfigures, the step length is plotted over the desired speed. The regularities of the gaits are shown as gradients of blue for wave gait, green for tetrapod, and red for tripod gaits. Darker points depict a parameter set that achieves a higher regularity. For ±0.05 m/s, of the three distinct gaits wave gait is predominant. Although most of the depicted data sets achieve only a low speed factor using small steps, some parameter sets achieve the desired walking speed with speed factors of approximately 1. The step lengths for these results differ over a range of ∼2 cm up to ∼40 cm. The step lengths

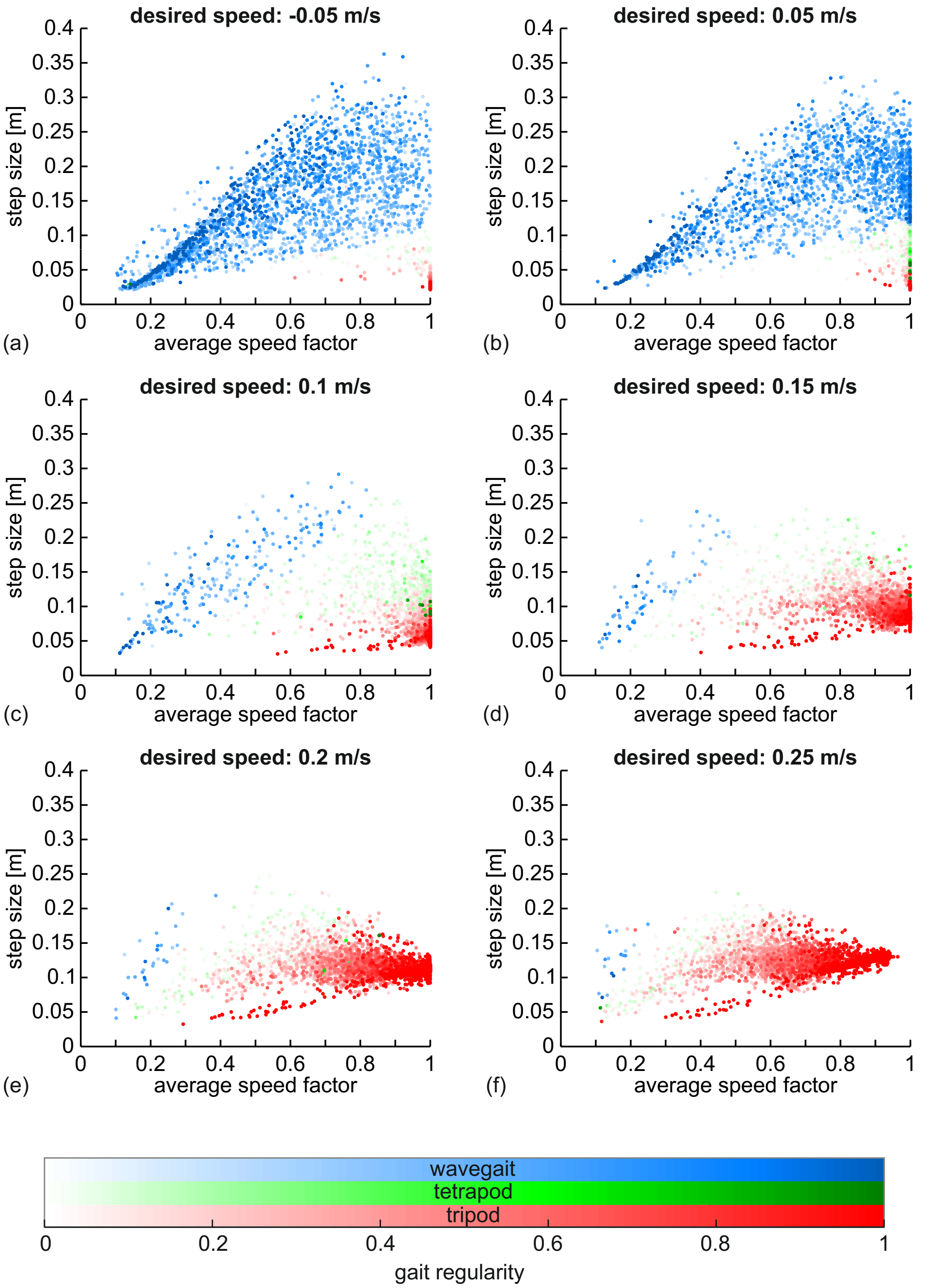


Figure 6.6.: The subplots show the results of 25,000 parameter sets that achieve at least a speed factor of 0.1 for different desired walking speeds between -0.05 m/s and 0.25 m/s. The achieved speed for each trial is the product of the desired speed and the average speed factor (x-axis). In each plot the step size is plotted over the average speed factors. The regularity of the gaits is depicted by gradients of blue for wave gait, green for tetrapod, and red for tripod gait.

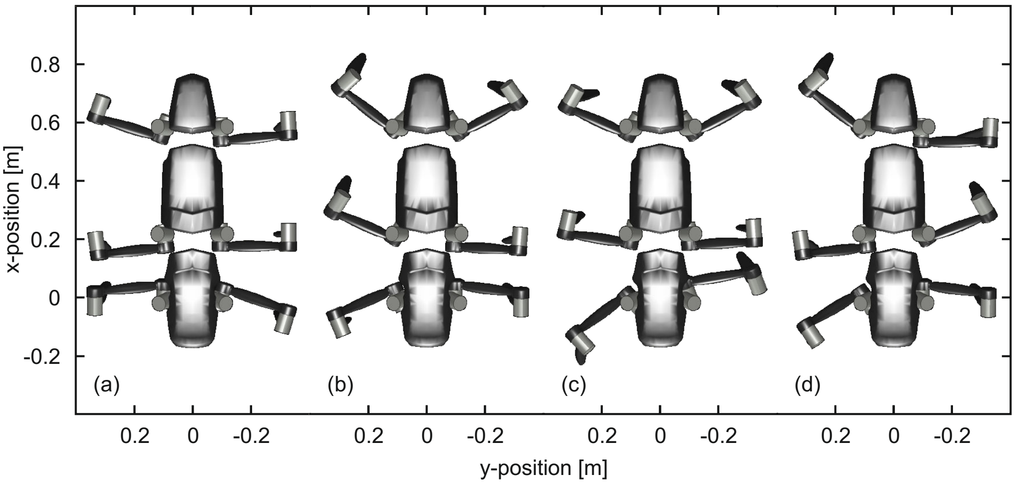


Figure 6.7.: (a)–(d) Four exemplary starting postures that have been used during the walking optimization to achieve posture-independent gait characteristics.

of stick insects relative to their body size is usually much longer than the relative step lengths of hexapod robots. Therefore, to match the technical to the biological system as far as possible, longer step lengths are preferable. As can be seen from the green and red data points at high speed factors and comparatively small step lengths, also tetra- and tripod gaits were detected at a speed of 0.05 m/s. For medium walking speeds of 0.1 m/s and 0.15 m/s, the cluster of tetrapod gaits shifts upwards towards bigger step lengths. Although multiple data points exist for wave gaits at these speeds, no parameter set achieves the full desired speed. For these speeds as well, often tripod gaits were detected. At faster desired walking speeds of 0.2 m/s and 0.25 m/s only few occurrences of tetrapod gait were detected. In these cases, only tripod gaits achieve high speed factors.

The dominance of wave and tripod gaits in fig. 6.6 over tetrapod gaits is apparent, comparing the total number of parameter sets that evoke the respective gaits. A reason for this mismatch might be that both wave and tripod gait are easier to enforce using the coordination influences. By increasing solely influence 2 (enforcing the start of a swing phase if a neighboring leg finished its swing phase) these gaits will likely emerge on their own. However, for tetrapod to occur, the effects of the coordination influences must be harmonized.

As mentioned, during the initial data acquisition, the test runs were aborted if one of the following three requirements were not met:

- The leg tips of anterior legs must be positioned in front of posterior legs.
- If all legs are in stance phase and positioned at their home positions, the minimum stability margin must be achieved.

- The average speed factors for all runs must be above 0.1.

For the further analysis the requirements for suitable parameter sets were increased. In addition to the previously mentioned requirements only those parameter sets were considered for evaluation that achieved the following goals:

- The minimum speed factor for any of the desired walking speeds up to 0.2 m/s should not fall below 0.4.
- At the lowest forward speed (0.05 m/s) a regular wave gait with a regularity of at least 0.5 must be observable.
- At different desired walking speeds a tetrapod and a tripod gait must each be detectable with a regularity of more than 0.5.
- Up to the desired speed at which the peak of tetrapod regularity is detected the tripod regularity must not rise above 0.1.

The limits in these requirements were chosen based on a first evaluation of the obtained data that showed, e.g., that the speed factors were reduced for all parameter sets for desired speeds above ∼0.2 m/s.

Based on these requirements, 75 parameter sets were filtered from the original data. For these parameter sets additional tests were conducted: For each parameter set the robot was commanded to walk at speeds in the range [-0.35, 0.35] m/s with incremental steps of 0.005 m/s. To exclude parameter sets that achieve good results only for specific starting postures, 13 different starting postures[5] were used for each combination of parameter set and desired speed (exemplary starting postures are shown in fig. 6.7; initial leg tip positions are listed in table B.2). These starting postures were randomly generated but then used unchanged for the test runs of all parameter sets.

Applying the same filter criteria as before, the number of valid parameter sets could be reduced to 6. For reference, the parameter sets will be numbered $PS_1,\ldots,PS_6$. The corresponding values are listed in table B.1. In fig. 6.8, the gait regularity results for different starting postures are shown as overlays for the 6 parameter sets in subfigures (a) to (f). In each subfigure, the combined results for 13 different starting postures are depicted. The mean speed factor for all postures is shown as black line with the minimum and maximum speed factors as gray lines. The mean step length is shown as yellow line. The regularity measures for wave gait (blue), tetrapod (green), and tripod (red) are shown as overlays of semi-transparent filled plots with the respective maximum regularity as dashed lines. Therefore, light areas mean that only few of the starting postures result in an explicit gait at the given walking speed, whereas darker areas show that the respective gait is achieved for most/all of the starting postures. Due to the filtering, the common feature is the wave gait at slow, the tetrapod at medium, and the tripod gait at high walking speeds. Although the actual speed degrades for higher desired speeds, typically, the tripod gait is maintained. For negative walking

[5] This is the maximum number of robot simulations that can be run in parallel on the available computer.

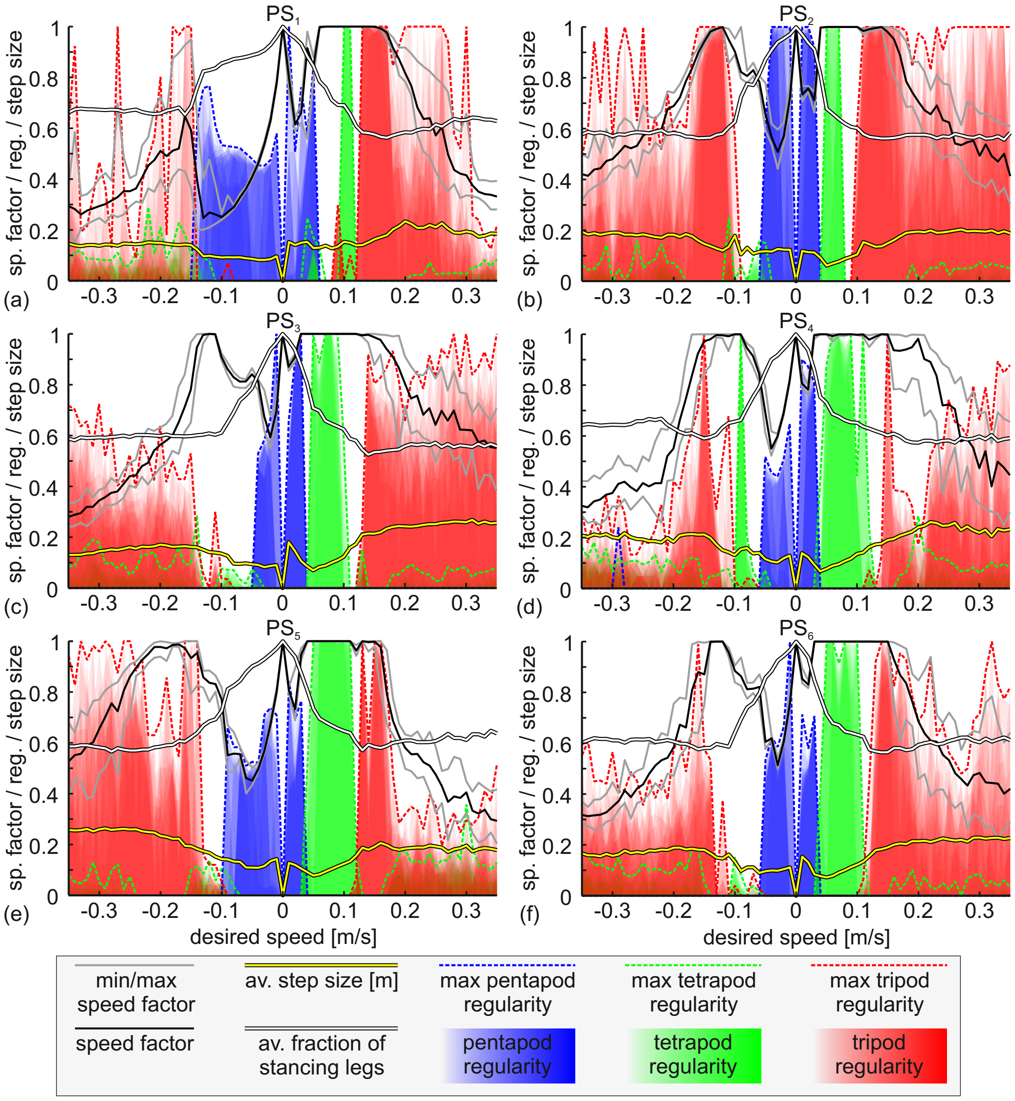


Figure 6.8.: Speed spectra for six valid parameter sets (a)–(f). The subfigures show the regularity of distinct gaits as semi-transparent, color-filled plots. The data for 13 different starting postures are shown as overlays in each subfigure. Due to the semi-transparency of the color-filled regularity plots, light areas indicate that only few of the starting postures lead to the given gait regularity whereas darker areas show the independency of the gait regularity from the starting postures. The respective maximum regularities of the gaits are plotted as colored, dashed lines. The mean speed factor is shown as black line, enclosed by two gray lines that depict the minimum and maximum speed factor achieved for all starting postures. The mean step length is plotted as yellow line.

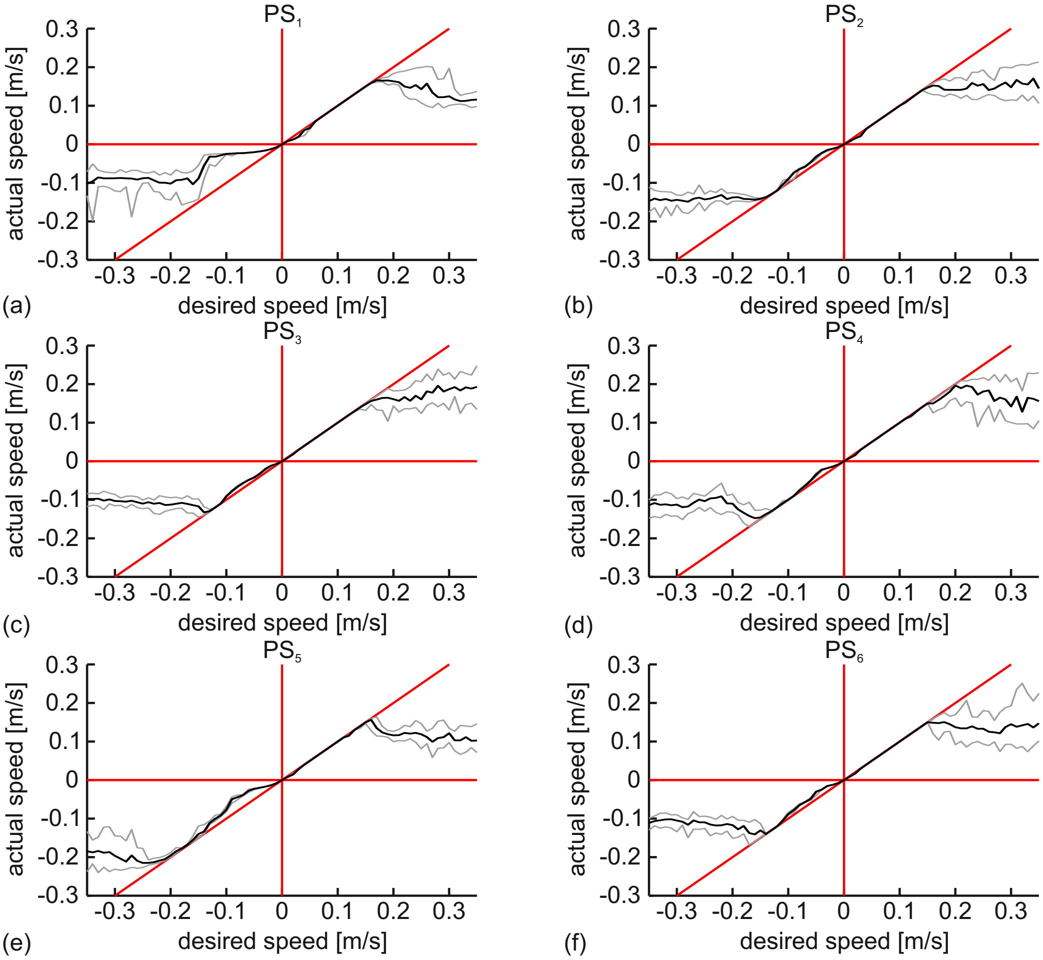


Figure 6.9.: Plots of actual speed over desired speed for the six chosen parameter sets (a-f). The mean actual speed is plotted as black line with the maximum and minimum speeds over the different starting postures as encompassing gray lines. For reference, the bisecting line and the horizontal and vertical lines passing the point of origin are shown in red. The difference between the desired and the actual speed is due to the speed reduction mechanisms, e.g., to maintain stability.

speeds no parameter set achieves the same sequence of gaits. Many parameter sets show a regular wave gait at low negative speeds and tripod gaits at high negative speeds, whereas tetrapod gaits are usually not observable. As stated before, the direction of the coordination influences was adopted from the Walknet concept that is grounded on observation of forward walking stick insects. Although the strengths of the influences were modified, the directions were maintained. In addition, the quality of backward

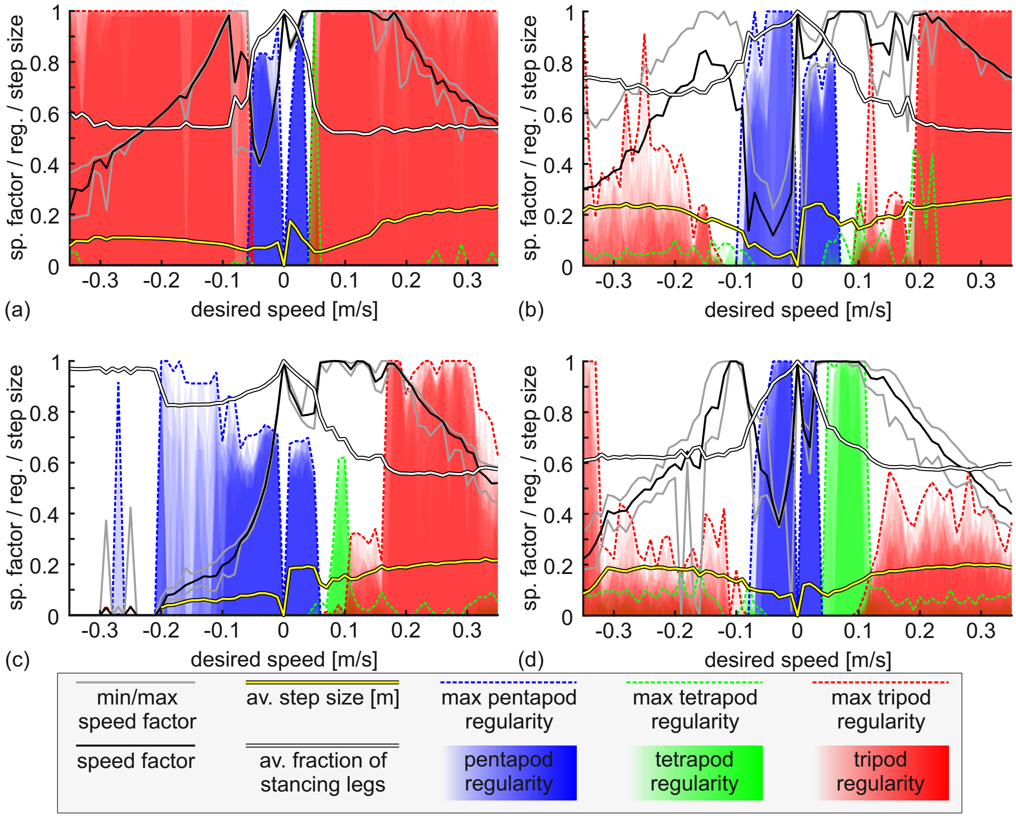


Figure 6.10.: Speed spectra for four parameter sets that show various deficits. The subfigures (a) to (d) show the regularity of distinct gaits as semi-transparent, color-filled plots. The data for 13 different starting postures are shown as overlays in each subfigure. Therefore, light areas show that only few of the starting postures lead to the given gait regularity whereas darker areas show the independency of the gait regularity from the starting postures. The respective maximum regularities of the gaits are plotted as colored, dashed lines. The mean speed factor is shown as black line, enclosed by two gray lines that depict the minimum and maximum speed factor achieved for all starting postures. The mean step length is plotted as yellow line.

walking was not considered during the preselection of parameter sets. Still, the adaptive concept enables locomotion even in these unaccounted scenarios.

For reference, fig. 6.9 shows the actual speed vs. the desired speed for the same parameter sets already depicted in fig. 6.8. During forward walking, the different parameter sets achieve roughly the same maximum speed. Up to this limit, the actual speed follows the desired speed rather precisely. For higher desired speeds, the speed factor decreases and thus the actual speed does not increase substantially—in some cases the actual speed even decreases.

As examples for improper walking behaviors, fig. 6.10 shows four examples of gait spectra for different parameter sets that do not meet the defined requirements. (a) shows a parameter set for which, after a short range of wave and tetrapod gaits, tripod gait is dominant throughout the whole speed range. A high speed factor is achieved up to a speed of 0.2 m/s. As can be seen from the progress of the step length, for the lowest speeds very small step lengths occur. For faster walking, the step length gradually increases. The adaptiveness of this parameter set is limited to the change of the step length for different speeds. Therefore, the reproduction of biological walking is not achieved. (b) shows an example in which only wave gait and tripod reach sufficient regularity levels, whereas tetrapod is barely detectable. (c) was excluded since wave gait did not appear for all starting postures with the required regularity. Finally, in (d), a rare example is shown for which a temporally stable tetrapod is achieved but the tripod regularity does not exceed 0.5 for all starting postures. Ideally, the robot should change gradually from wave gait over tetra- to tripod gait with increasing speed. Contrary to the previously shown parameter sets, none of these four examples fulfills this requirement.

6.4. Results for Different Locomotion Tasks

To verify the functionality of the presented approach, the robot was virtually tested in different scenarios. Beside the already partially presented forward and backward walking also curve and sidewards walking were tested. From the parameter sets that fulfill the requirements, PS_1 was chosen for the further experiments since it shows the expected gait transitions at nearly constant step lengths—at least for forward walking. For an informed selection of the best parameter set for a specific application, the side conditions must be defined and eventually tested using the different parameter sets. Although the subsequent results will be all based on PS_1 the other parameter sets are expected to achieve qualitatively comparable results.

6.4.1. Forward Walking

As examples for the speed-dependency of the gait, fig. 6.11 shows gait patterns at three different desired speeds for parameter set PS_1. Starting from one of the default starting postures the robot achieves a recurring coordination pattern after a short initial transient phase.

In fig. 6.11 (a), a gait pattern corresponding to wave gait is shown for a desired speed of 0.01 m/s. As can be seen below the gait pattern, the speed factor is reduced periodically. A notable peculiarity of this pattern are the overlapping swing phases of the middle legs (L2 and R2), a feature that can be observed also in the gait of stick insects (Dürr 2005). If simultaneous swing phases of contralateral legs are undesirable, the contralateral effects of coordination influences 1 and 2 can be increased to reduce the likeliness of this occurrence.

As shown in fig. 6.11 (b), during tetrapod walking at 0.11 m/s, the speed factor is not

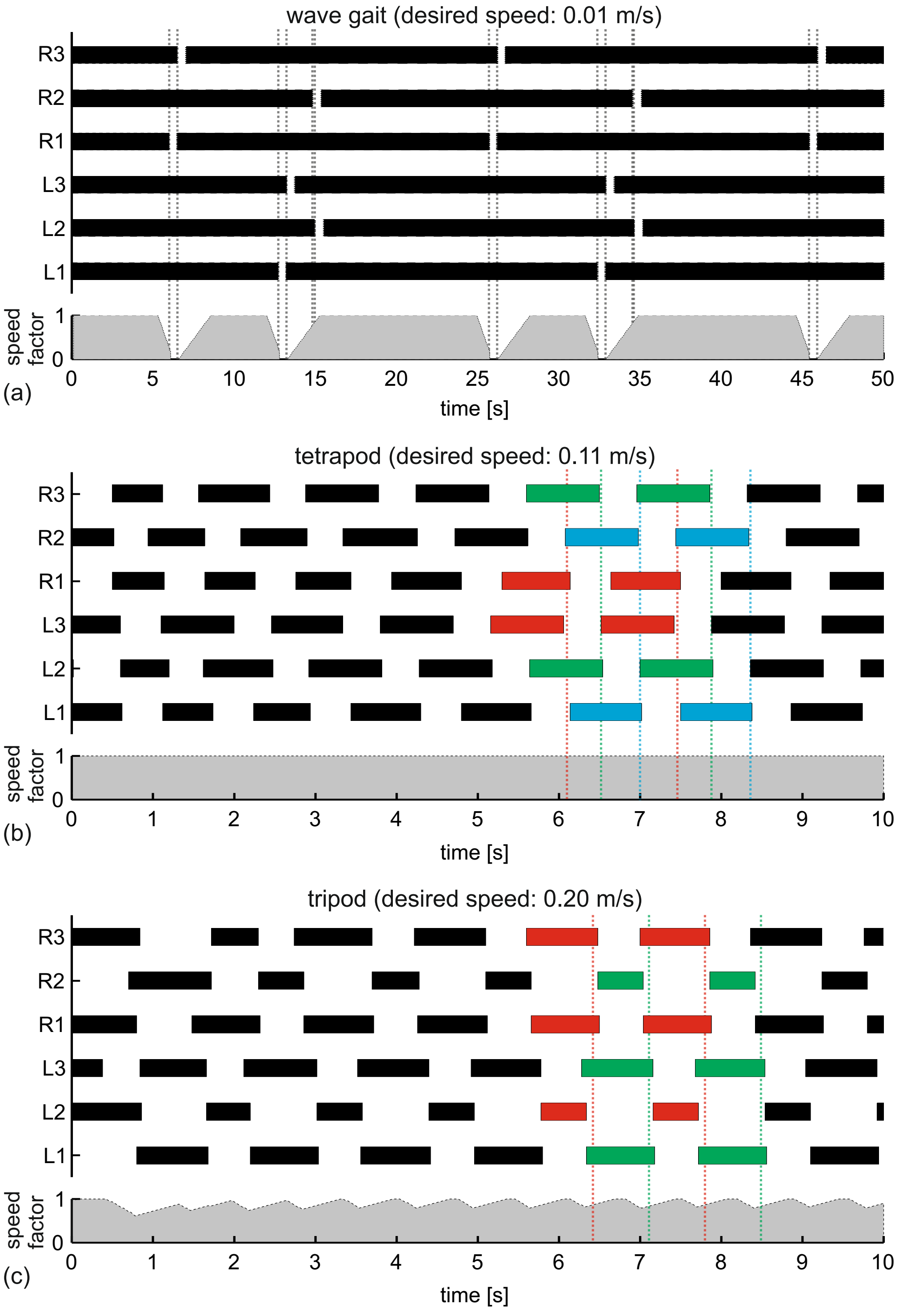


Figure 6.11.: Gait patterns and speed factors for parameter set PS_1 at different speeds for straight forward walking. Bars indicate stance phase. The three examples are shown for desired speeds of 0.01 m/s for wave gait in (a), 0.11 m/s for a tetrapod gait in (b), and 0.2 m/s for tripod gait in (c). In (a), vertical dashed lines exemplarily mark the time of liftoff of the individual legs. In (b) and (c), for two consecutive step cycles, the stance bars of legs groups are colored. The vertical, dashed lines indicate the middle of the liftoff period of the respective leg group.

reduced. The legs of the respective groups {R3, L2}, {R2, L1}, and {R1, L3} start their swing phases almost concurrently and the phase shift between the leg groups is even.

Tripod gait is depicted in fig. 6.11 (c) for a desired speed of 0.2 m/s. The periodic speed factor reductions lead to an actual speed of 0.183 m/s for this case.

6.4.2. Curve Walking

To test the curve walking characteristics of controller and parameter set PS_1, the robot was given heading vectors slanted relative to the robot axis for different desired walking speeds. As shown in section 5.3.2, within the stance module points can be defined at

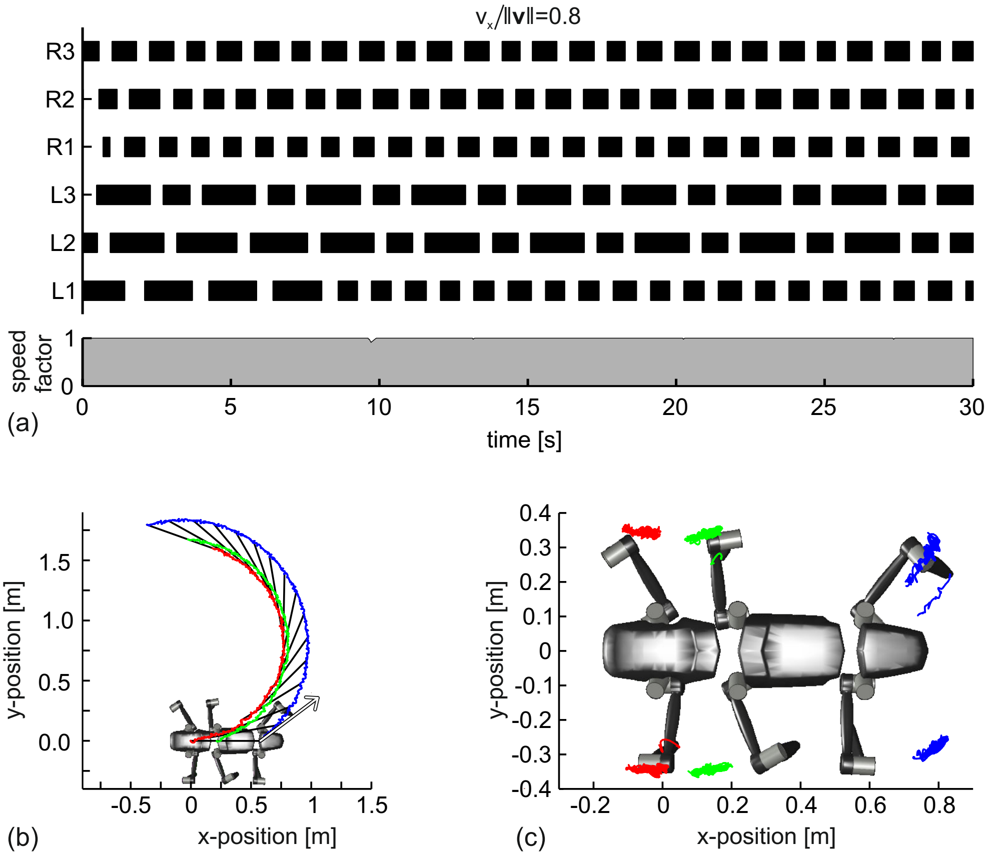


Figure 6.12.: Exemplary results for curve walking using the pull vector that is depicted as white arrow in (b). (a) shows the gait pattern and the speed factor for a walking period of 30 s. (b) visualizes the positions of the body axis as black lines at intervals of 2 s. The trajectories of the midpoints between the front, middle, and hind legs are plotted in blue, green, and red, respectively. (c) shows the leg trajectories relative to the local RCS.

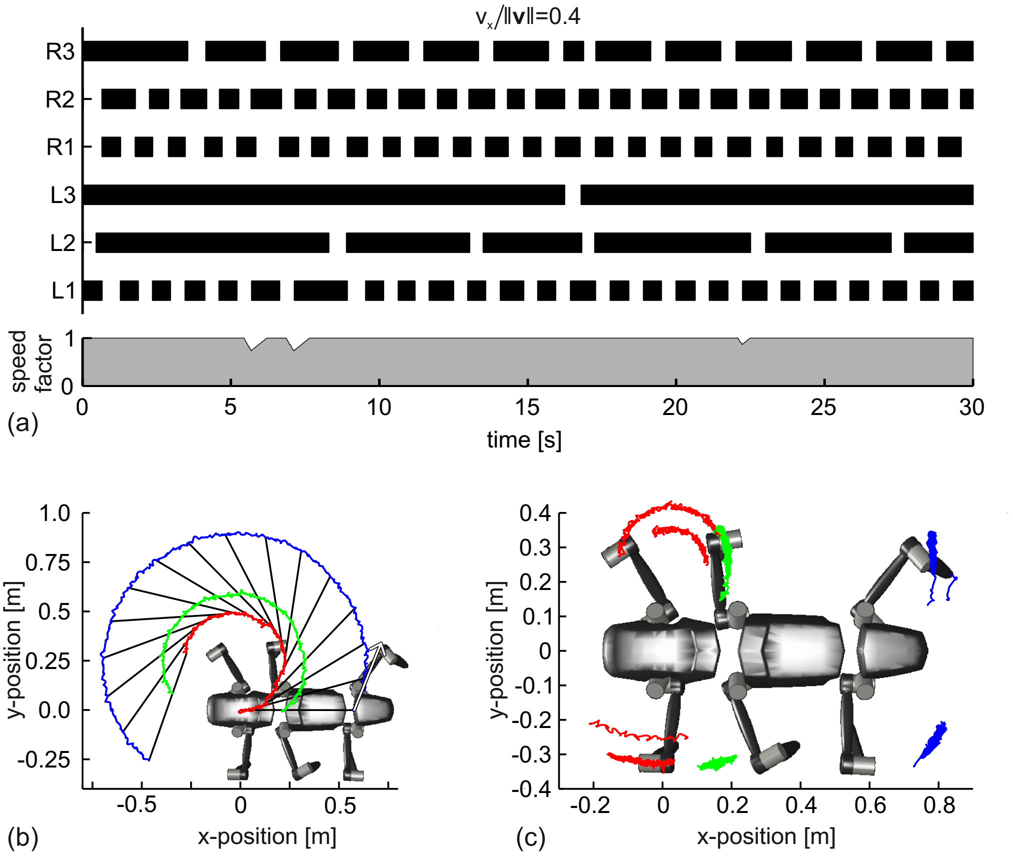


Figure 6.13.: Exemplary results for curve walking using the pull vector that is depicted as white arrow in (b). (a) shows the gait pattern and the speed factor for a walking period of 30 s. (b) visualizes the positions of the body axis as black lines at intervals of 2 s. The trajectories of the midpoints between the front, middle, and hind legs are plotted in blue, green, and red, respectively. (c) shows the leg trajectories relative to the local RCS.

which the robot can be virtually "pulled" to a given direction. In this case, only the frontal point (located between the onsets of the front legs) is used to set the direction, corresponding to the case depicted in fig. 5.14 (a)–(d). The direction in which this point is pulled is given as ratio between the x-component of the velocity vector v_x (along the body axis) and its norm $\|\boldsymbol{v}\|$. Therefore, a ratio of 1 corresponds to straight forward walking, whereas a ratio of 0 indicates sharp turning about the center between the hind leg onsets.[6] Speed spectra alike those shown for different parameter sets during forward walking are depicted in fig. 6.15 for just one parameter set (PS_1) but different

[6] The direction, in which the robot is pulled is visualized for three examples in subfigures (b) of figs. 6.12 to 6.14 (b) as white arrows

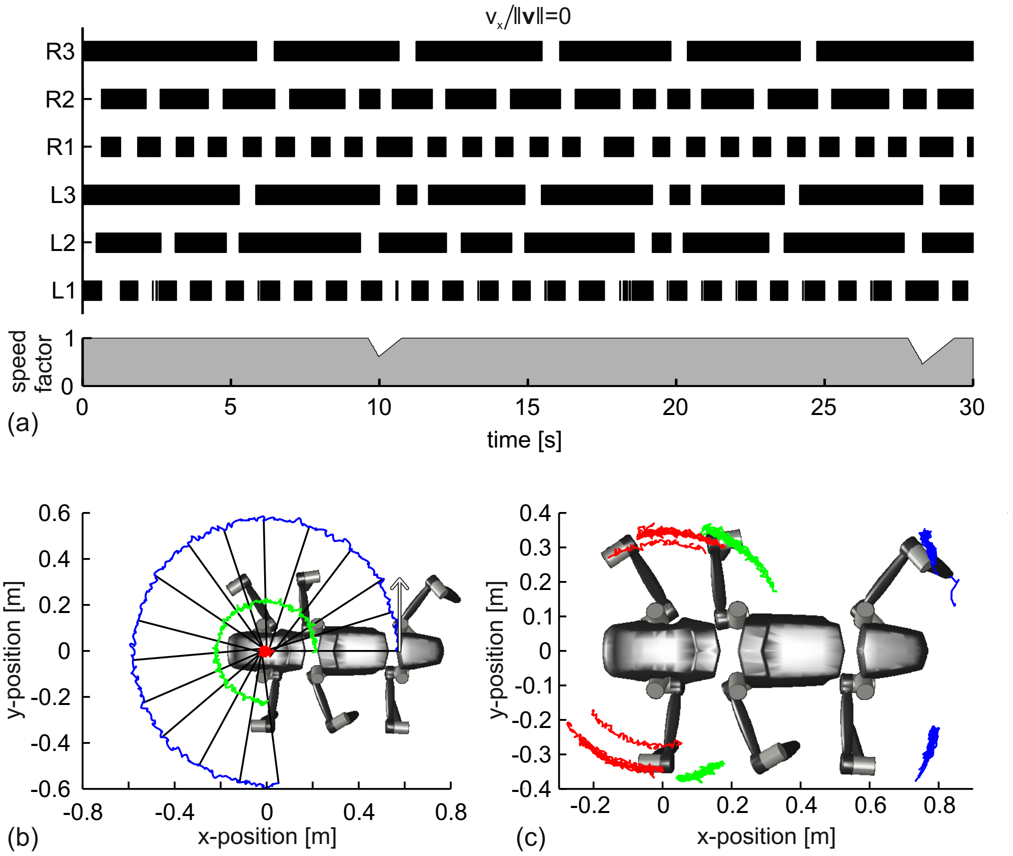


Figure 6.14.: Exemplary results for curve walking using the pull vector that is depicted as white arrow in (b). (a) shows the gait pattern and the speed factor for a walking period of 30 s. (b) visualizes the positions of the body axis as black lines at intervals of 2 s. The trajectories of the midpoints between the front, middle, and hind legs are plotted in blue, green, and red, respectively. (c) shows the leg trajectories relative to the local RCS.

direction vectors. Subfigure (f), displaying the case for straight forward walking, is identical to fig. 6.8 (a) and is shown here only as reference. In subfigures (a)–(f), the ratio is continuously increased, resulting in increasing curve radii. Figure 6.15 (e) shows the speed spectrum for curve walking at a direction ratio of $v_x/\|\boldsymbol{v}\| = 0.8$. Already for this case the regularity of tripod gait for higher walking speeds is considerable reduced. Tetrapod gait, however, is still detectable although only as sharp peak at a speed of 0.09 m/s. For even tighter curve walking, this peak vanishes as well, leaving wave gait as only detectable distinct gait.

Figures 6.12 to 6.14 show the gait patterns, the trajectory of the central robot axis, and the leg trajectories for three different direction ratios and desired speeds of 0.1 m/s. The

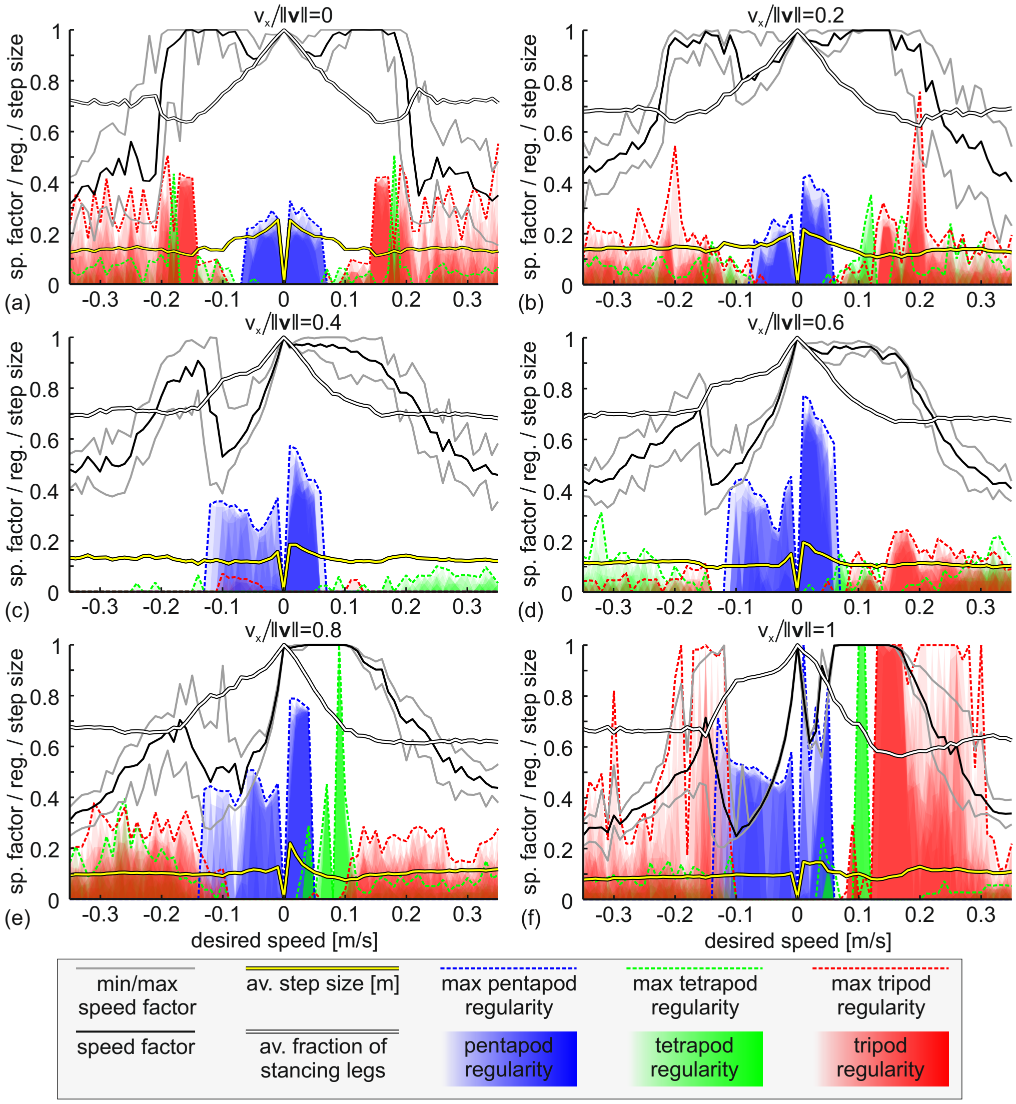


Figure 6.15.: Speed spectra of curve walking. Instead of fixed curve radii, the direction in which the frontal point is virtually pulled is changed for the six walking scenarios shown in (a) to (f). The direction is given as fraction of the x-component over the norm of the pull vector (relative to the local RCS). Therefore, a value of 1 results in pure forward walking (identical to fig. 6.8 (a)) whereas a value of 0 corresponds to an orthogonal pull direction. For exemplary trials, pull vectors are shown in figs. 6.12 to 6.14.

subfigures (a) in each of the figures show the gait patterns and the speed factors. Already at a direction ratio of 0.8, the pattern features a higher stance-swing frequency for the right legs that are on the outer side of the circular body trajectory (see fig. 6.12 (b)). The left middle and hind legs exhibit the lowest stance-swing frequency as they are located closer to the center of the turning circle. For tighter turns at a direction ratio of 0.4, depicted in fig. 6.13, this becomes even more pronounced. For the rotation about the point between the hind leg onsets that is shown in fig. 6.14, the effect is reduced again as the left hind leg is further away from the center of turning. In this case, the stance direction of this leg is reversed since it stances from back to front.

The body trajectories are depicted in subfigures (b). In each case the central axis of the body—connecting the points between the hind legs' and the front legs' onsets—is plotted as black line at intervals of 2 s. The trajectories of the central points between the respective contralateral leg pairs are shown in blue (front), green (middle), and red (hind). The initial posture of the robot is shown as reference with a white arrow indicating the direction in which the frontal point is virtually pulled (see fig. 5.14). Since the point at which the robot is virtually pulled coincides with the central point between the front legs' onsets, the pull vector is oriented tangential to the circle that is implied by the blue, frontal trajectory. In the subfigures (c) the leg trajectories are plotted relative to the local RCS. In this representation the circularity of the leg trajectories can be seen. The initial posture of the robot is shown again in each figure. Starting from the initial positions the trajectories of the stancing legs during their first stance phase do not pass the respective home positions. Thus, for those legs that start with the stance phase, different, concentric trajectories are observable for the first and the latter stance movements.

6.4.3. Sidewards Walking and Tight Turning

Although the robot is based on the model of *Carausius morosus*, also walking scenarios that are untypical for the insect can be tackled using the omnidirectional walking controller. Examples for these cases are sidewards walking as it can be observed in crabs (Evoy and Fourtner 1973) and turning motions on the spot. Compared to the previously shown examples that use only the stance module's frontal *pull point*, in these cases also the rear *pull point* must be utilized (see fig. 5.14). For sidewards walking the pull directions for both points are set to be equal and parallel (see fig. 6.16 (b)). For the turning movement the direction at the rear pull point is reversed, thus creating an asymmetrical situation (see fig. 6.17 (b)). The analyses of the gait patterns for parameter set PS_1 are shown in fig. 6.18.

In both cases, the results are nearly symmetrical for positive and negative pull directions. Due to the symmetric robot setup, this is expectable. Still, it underlines again the starting posture independent operation of the walking controller. For sidewards walking a gait spectrum similar to those observed for forward walking can be demonstrated. The tetrapod gait regularity is not as prominent, however, peak values of 0.5 can be detected. Up to speeds of 0.15 m/s high speed factors are achieved. Also for the turning movements constantly high speed factors are achieved in a similar range of desired speeds. In

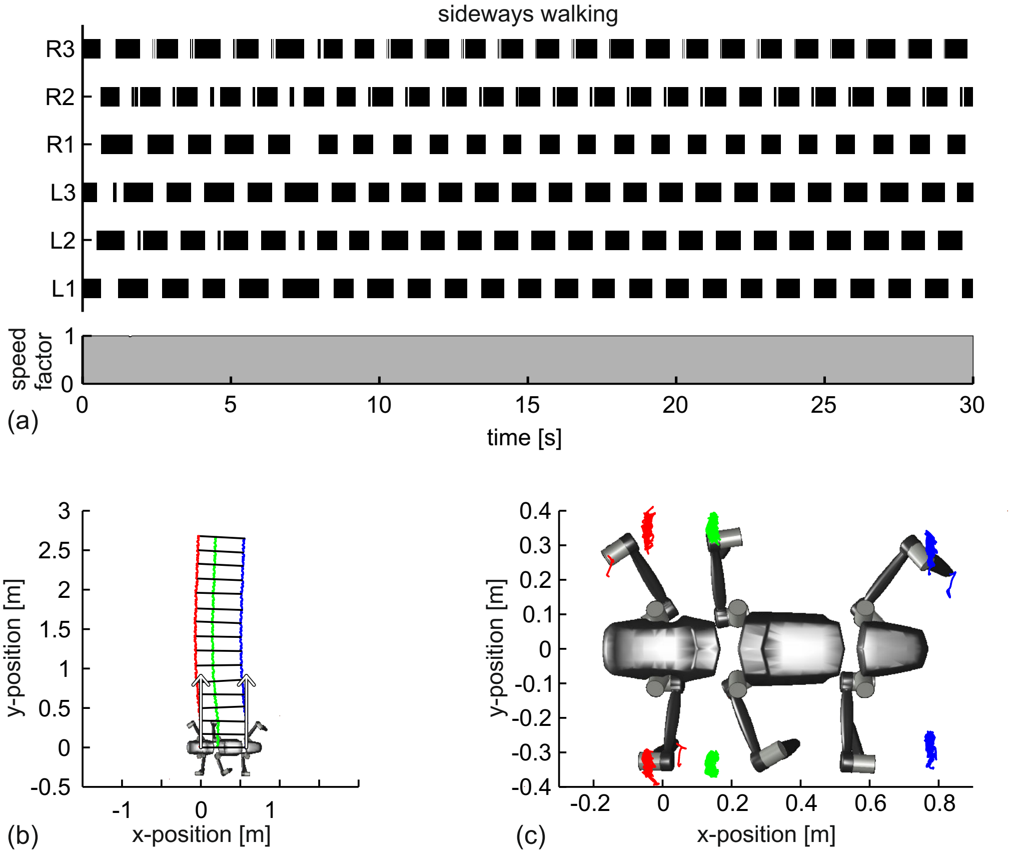


Figure 6.16.: Exemplary results for sidewards walking using the pull vectors that are depicted as white arrows in (b). (a) shows the gait pattern and the speed factor for the walking period of 30 s. (b) visualizes the positions of the body axis as black lines at intervals of 2 s. The trajectories of the midpoints between the front, middle, and hind legs are plotted in blue, green, and red, respectively. (c) shows the leg trajectories relative to the local RCS.

contrast to tight curve walking (see fig. 6.15 (f)) the turning movement exhibits a high tripod regularity for high speeds.

Gait patterns, body and leg trajectories for these two cases at speeds of 0.1 m/s are depicted in figs. 6.16 and 6.17. Especially during sidewards walking a high regularity of the gait pattern is apparent in the gait pattern. For the legs on the right side, irregular, short interruptions of the stance phase occur if the respective contralateral leg touches down. In these cases, influence 2 triggered the start of a swing phase. In some cases, even two of these interruptions can be observed for the right front leg during the period of a regular stance movement. These second interruptions are also triggered by influence 2, in this case originating from the ipsilateral neighbor, the right middle leg. The step

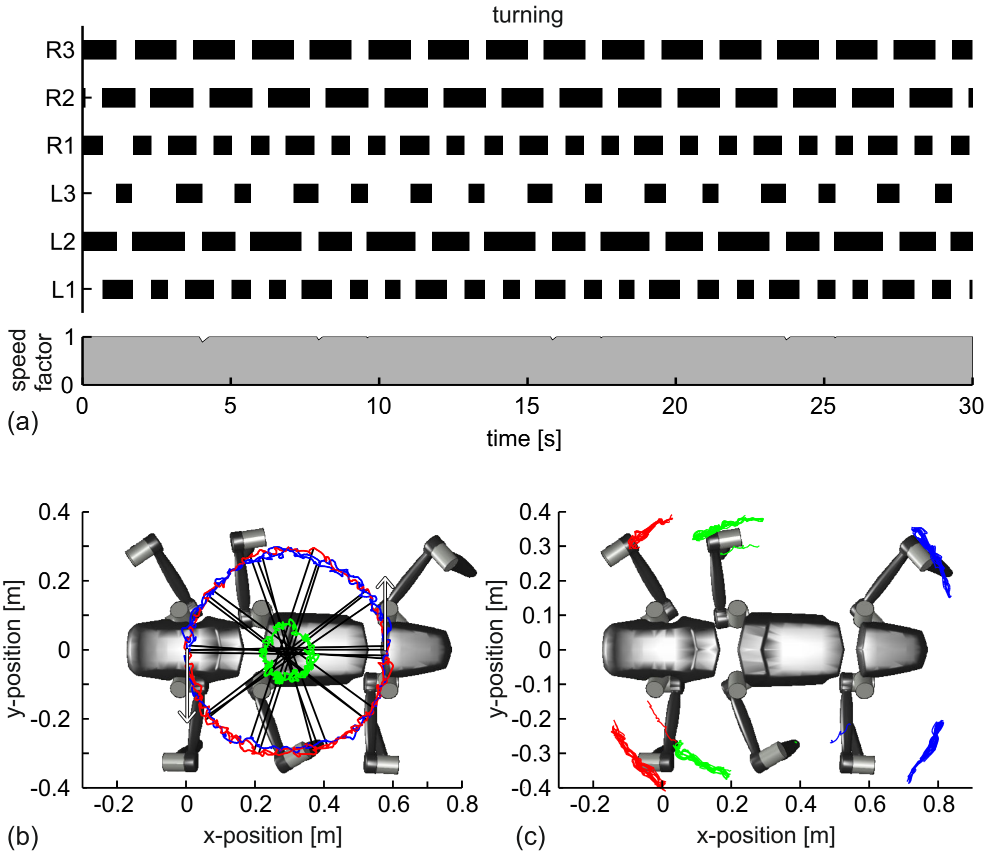


Figure 6.17.: Exemplary results for sharp turning using the pull vectors that are depicted as white arrows in (b). (a) shows the gait pattern and the speed factor for the walking period of 30 s. (b) visualizes the positions of the body axis as black lines at intervals of 2 s. The trajectories of the midpoints between the front, middle, and hind legs are plotted in blue, green, and red, respectively. (c) shows the leg trajectories relative to the local RCS.

sizes for this walking scenario are very short, as can be seen in fig. 6.16 (c). However, as shown in fig. 6.18 (a), the step size increases for lower and even for faster walking speeds.

For the turning movement only wave gait for low and tripod for high speeds can be reliably detected. A fraction of the starting postures also evokes a tetrapod gait pattern but this seems rather coincidental. As for curve walking also in this case the stance-swing frequency differs for each leg (see fig. 6.17 (a)) with the highest frequency at the front legs. Since the center of rotation is located between the frontal and the rear pull points, the front and hind leg trajectories overlap in fig. 6.17 (b). Correspondingly, fig. 6.17 (c) shows concentric stance trajectories for the legs.

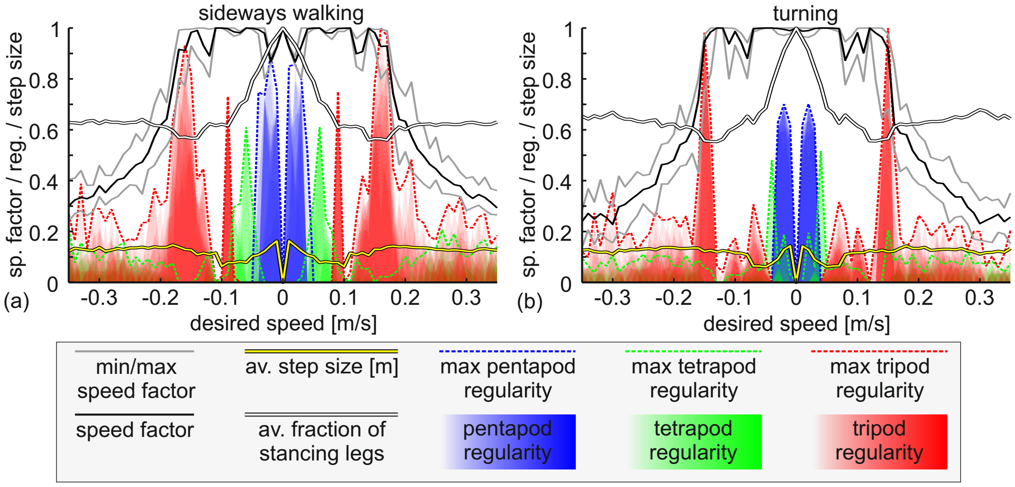


Figure 6.18.: Speed spectra for (a) sidewards walking and (b) tight turning. The corresponding pull vectors are depicted in figs. 6.16 and 6.17, respectively.

Summary

The walking controller that was described in chapter 5 extends the concepts of bioinspired leg coordination used within Walknet (see chapter 4), e.g., to consider the necessity of stability maintenance. For this purpose, additional control mechanisms have been introduced that can reduce the walking speed and alter the leg coordination. In comparison to Walknet that coordinates the legs in a bioinspired way based on the desired walking speed, the new controller will always prioritize stability over walking speed. As a consequence, the choice of parameters does not influence the stability as shown during the evaluation of Walknet in section 4.2. Instead, most notably, the choice of walking parameters influences the overall walking speed. However, to also achieve an adaptive gait generation, an optimization was conducted in simulation that considered not only the walking speed but also the speed-adaptiveness of the gait generation. For this purpose, three measures for the regularity of wave, tripod, and tetrapod gaits were introduced. For 80,000 randomly selected parameter sets the quality of gait generation was evaluated. From this data set, 75 were selected for a detailed analysis regarding the gait pattern generation at different walking speeds. For 6 of these parameter sets a starting posture independent generation of the bioinspired gait patterns at the different walking speeds could be demonstrated (wave gait at slow speeds, tetrapod at medium, and tripod gait at high walking speeds).

Moreover, since the definition of the PEP was extended to allow omnidirectional walking, also backward, curve, and sidewards walking could be realized with the new controller. Since the main focus of the preceding optimization was on forward walking, during omnidirectional walking (e.g., tight turns) seemingly suboptimal gait patterns emerged. Therefore, to achieve better results during special walking scenarios dedicated optimizations should be conducted to find suitable parameter sets. However, with these test it could be shown that the controller is operational also in situations for which it was not explicitly optimized.

To validate the simulation results of this chapter on the real robot, in chapter 7 exemplary walking trials were conducted on Hector.

7. Application of the Bioinspired Control on HECTOR

Abstract: The main advantage of legged compared to wheeled robots is their ability to navigate freely without the restrictions imposed by wheels. As shown in chapter 6, the combination of workspace definition based on the concept of unrestrictedness and the coordination influences that have been derived from WALKNET allow movements in all horizontal directions. As practical application of this schema and to prove the omnidirectional walking capability on a real system, the combination of bioinspired walking controller with bioinspired collision avoidance mechanisms is used to navigate autonomously towards a goal position. In addition, basic climbing abilities will be demonstrated that rely only on the ground contact detection using the sensor information of the joint drives. Some of the results and outcomes presented in this chapter have already been published in the following paper together with additional results from the specified co-authors: Meyer, Bertrand, Paskarbeit, Lindemann, Schneider, and Egelhaaf 2016; Paskarbeit, Schilling, Schmitz, and Schneider 2015

In section 6.4, the functionality of the presented controller was demonstrated in simulation for several locomotion scenarios like walking forward, backwards, sidewards and in circles. However, the presented trials show only continuous motions of the robot with an unchanged directional input (see section 5.3.2). In a real world scenario, the robot must be able to cope with varying directional inputs and it must adapt to the environment. Nevertheless, to show the ability of the new walking controller to generate adaptive gait patterns at different walking velocities, first of all, it was tested during straight forward walking. In the next step, to demonstrate the capability to deal with varying directional inputs, the robot was equipped with a camera to allow for vision-based collision avoidance. The camera images are fed into the bioinspired collision avoidance controller of Meyer et al. (2016) to obtain directional input for the walking controller. Thus, the walking controller must react to continuously changing goal directions. In a third experiment, the stance height adaptation is utilized to climb over an obstacle. In this case, no additional sensors like cameras or tracking devices were used to solve the task. Therefore, the object detection was achieved solely by observation of proprioceptive signals from tactile sensing using the information gathered from the joint drives and the implemented elastomer couplings.

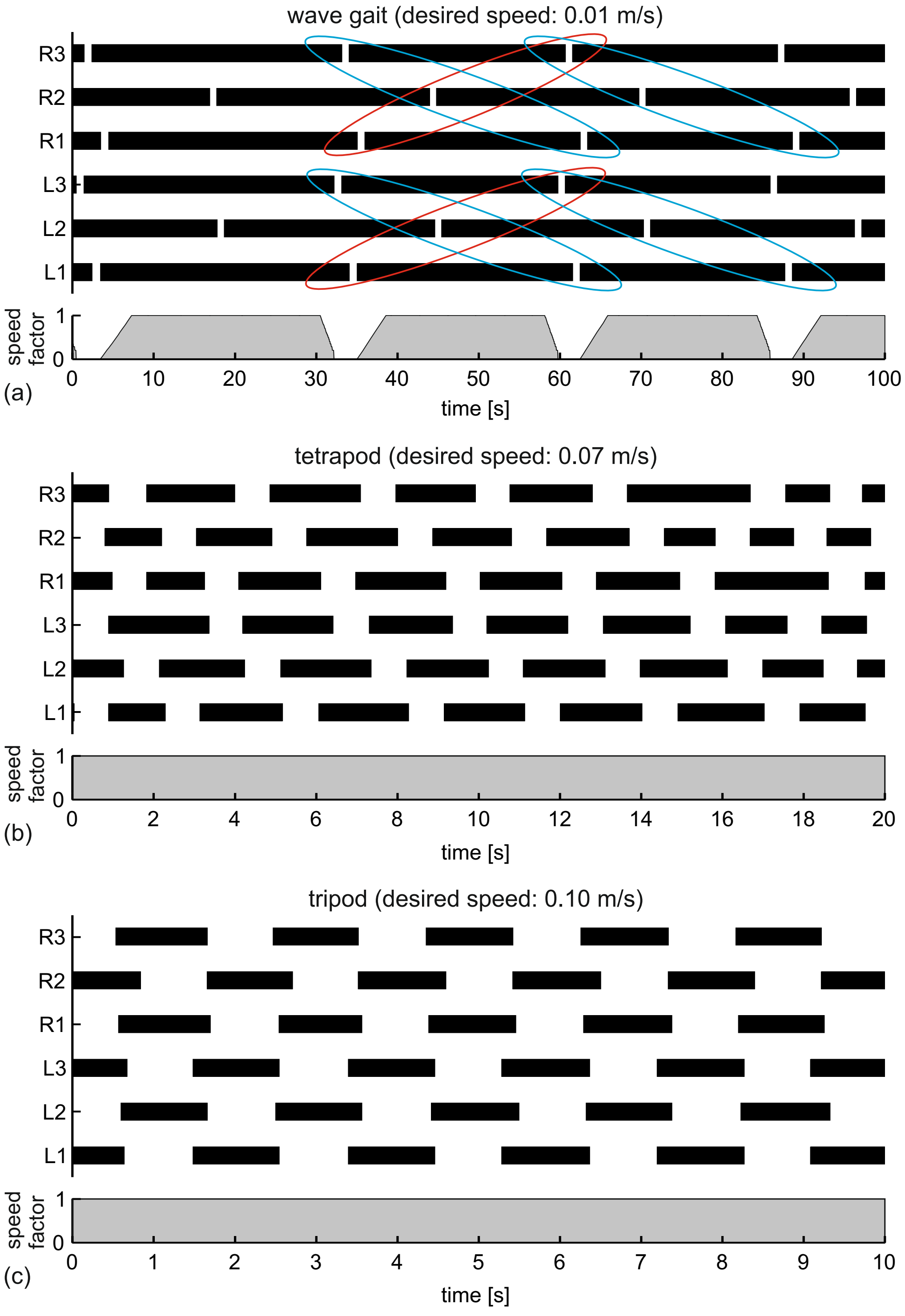


Figure 7.1.: Gait patterns and speed factors for parameter set PS_1 at different speeds for straight forward walking. Bars indicate stance phase. The subfigures show wave gait (a), tetrapod gait (b), and tripod gait (c). Red ellipses in (a) indicate metachronal waves traveling from hind to front legs. Blue ellipses indicate overlapping metachronal waves traveling in the opposite direction.

7.1. Evaluation of Forward Walking Gait Patterns

Using the simulation framework the new walking controller could be extensively tested prior to application on the real robot. In section 6.3 the simulation was used to find sets of control parameters that achieve an insect-like gait adaptability in dependence of the walking velocity. However, as often mentioned (Webb 2001), simulation results do not necessarily correspond to results obtained using a real robotic system. For example, although the characteristics of the joint drive compliance was emulated in the simulation, the real elastomer coupling is considerably more complex than the model (see section 2.3.1). Also, in the simulation the desired motor velocities are always achieved. In the real robot, however, due to the interaction between sensors, control electronics, and the motor, deviations from the desired angular velocities are inevitable. Therefore, a test on the real hardware is required to prove the applicability of the new walking controller.

To compare the results obtained from the simulation with the real robot, the walking controller was transferred to Hector. As described in section 2.6, the communication framework is designed to allow for a direct exchange between the simulation and the real robot. Therefore, it was not required to modify the controller prior to the transfer.

In fig. 7.1, gait patterns for three different walking speeds are depicted. As already shown in fig. 6.11 for walking within the simulation, also the real robot varies its gait pattern in dependence of the walking speed. To evaluate the regularity of the gait patterns, the regularity measures (see section 6.1) were applied to the three data sets.

In neither of the three trials, a wave gait pattern was detectable. For the gait patterns shown in fig. 7.1 (b) and (c), on average, more than one leg is in swing phase. Therefore, they do not fulfill the requirement for the application of the wave gait regularity measure. In the gait pattern shown in (a), on average, less than one leg is in swing phase. Therefore, the wave gait regularity measure can be applied. However, as highlighted by the red ellipses, the sequence of protractions is reversed. The metachronal wave of protractions starts at the front legs and travels backwards. Therefore, the wave gait regularity measure does not detect any regularity. However, as argued by Cruse (1979a), the gait pattern can be interpreted as overlapping, anteriorly traveling metachronal waves as depicted by the blue ellipses. In order to detect these patterns the regularity measure would need to be adapted.

The gait patterns which are depicted in fig. 7.1 (b) and (c) show the characteristics of tetrapod and tripod gaits. In fig. 7.2, the regularity measures are plotted over the duration of the concurrence interval. For the intermediate walking speed, the tetrapod regularity considerably exceeds the tripod regularity measure. At the higher walking speed, a clear tripod coordination is indicated by the corresponding regularity measure.

Overall, the walking controller generates similar gait patterns in simulation and on the real hardware. The differences of the results between the two systems can be accounted for by the simplifications of the simulation model.

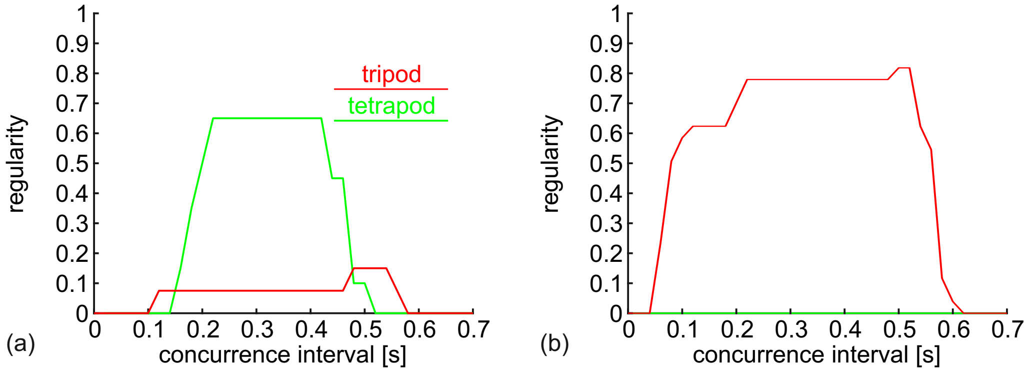


Figure 7.2.: (a) and (b) show the regularity measures for the gait patterns shown in fig. 7.1 (b) and (c), respectively, plotted over different durations of the concurrence interval.

7.2. Visual Collision Avoidance

Insects possess remarkable capabilities regarding vision processing. One example for these vision-based tasks is the collision avoidance of flying insects such as flies and bees. In this context, due to the proximity of their eyes, triangulation based on stereo vision as used by larger animals is not applicable. Therefore, flying insects must use some other mechanism to estimate the distances of objects. This ability is even more noteworthy considering the low spatial resolution of their eyes. Since mobile robots also have restricted space to house stereo cameras, they also have to cope with the challenge of small inter-camera distances. To cope with this limitation, a control approach based on optic flow which is described below was used on Hector as a mechanism for collision avoidance. Based on the active-gaze strategy observed in blowflies, Bertrand et al. (2015) developed a model that extracts the relative depth-structure of the environment from the optic flow that is perceived during translational motion. This concept is used in a module developed by Meyer et al. (2016) to avoid collisions with obstacles that obstruct the straight trajectory to a target. For the implementation on Hector, a camera system (an advanced version of the system presented by Irwansyah et al. (2014)) was incorporated into the front segment. The collision avoidance module of Meyer et al. (2016) was then used to provide the walking controller with a heading vector.

To evaluate the performance of the robot during walking, external tracking was used to obtain the position and orientation of the robot in a fixed coordinate system. The experiments were conducted in the *teleworkbench* (Tanoto et al. 2012), which is equipped with four cameras whose collective field of vision covers the ground (see fig. 7.3). Using markers attached to the robot, its x- and y-coordinates can be tracked.

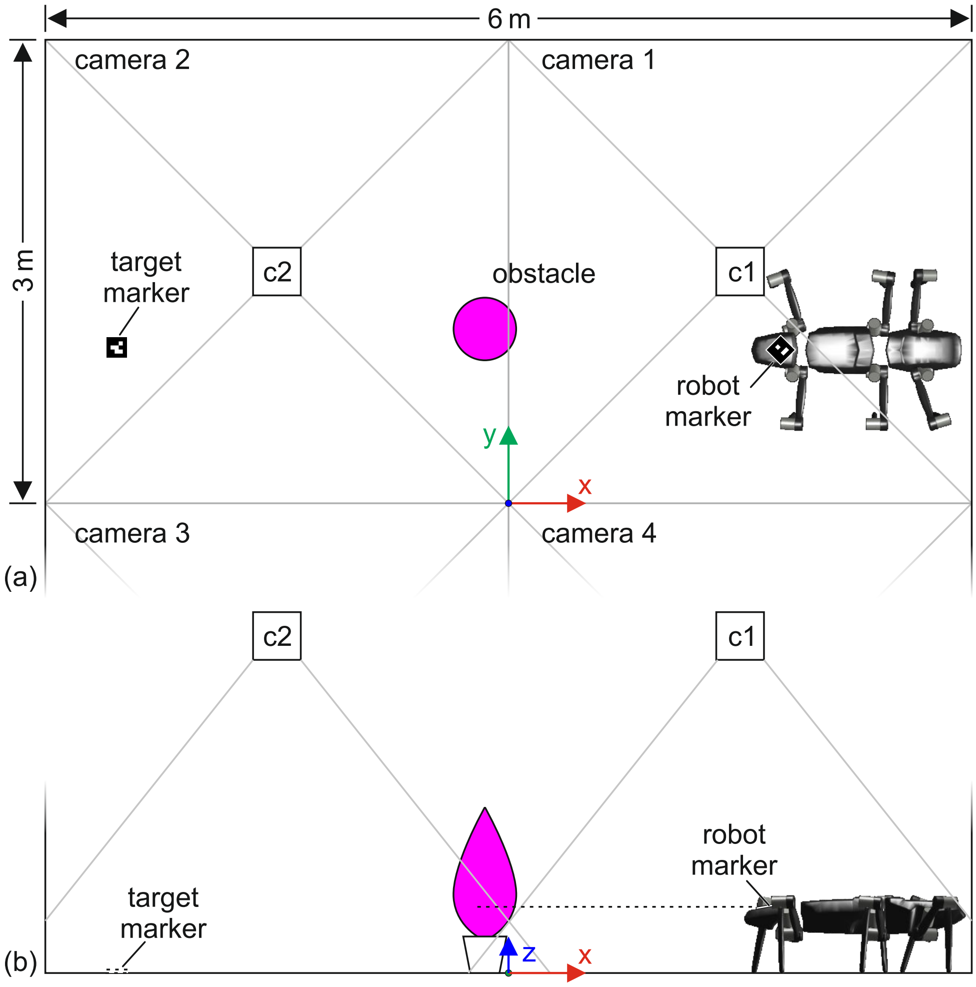


Figure 7.3.: Depiction of the setup of the teleworkbench. (a) top view of the setup with robot on the right and target on the left side. (b) side view of the setup.

7.2.1. Optic Flow Processing

During locomotion, the perspective on objects in a static environment changes depending on the distance between observer and object. The projection of these objects onto the retina of an animal or the camera sensor of a robot changes accordingly. Thus, relative movements shift the projection of each object in the field of view of the observer, which can be expressed as *retinal velocity*. The overall change of perception due to the movement of all objects is named *optic flow*. During rotational motion, the retinal velocity of all objects is equal—independent of the distance between object and observer. A translational motion of the observer, however, induces retinal velocities that depend on the distance between the objects and the observer. Therefore, the closer objects are

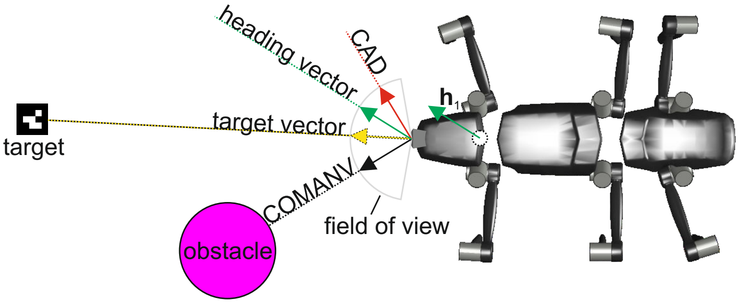


Figure 7.4.: Illustration of the relevant vectors for collision avoidance. The COMANV points into the average direction of close objects. Based on this direction, the CAD points in a direction, in which no close obstacles are detected. Using external tracking the target vector between the robot and the target marker is obtained. Compromising between the CAD and the target vector, the final heading vector of the robot is determined. This vector is used as input for the direction control of Hector to pull at the virtual robot body. The location of the frontal of the virtual pull points (see fig. 5.14) is depicted as white circle, the corresponding heading vector h_1 is shown in green.

the higher will be the optic flow. In insects optic flow is believed to be processed by a mechanism that can be modeled by EMDs (Elementary Motion Detectors; Buchner 1976) The response of EMDs depends on the contrast and the retinal velocity of a moving stimulus. Hence, the response of EMDs during translational movements corresponds to a contrast-weighted representation of distances to the objects (Schwegmann et al. 2014). This information can be used for obstacle avoidance (Bertrand et al. 2015). Based on the EMD output, the COMANV (Center-Of-Mass-Average-Nearness-Vector) is computed, which points towards the average direction of close objects. To evade these obstacles the CAD (Collision Avoidance Direction) is computed, which corresponds to the inverse of the COMANV, scaled to the horizontal field of view. This vector therefore points in a direction, in which no obstacles are present or at least not present close to the robot. Together with the target vector, the CAD is used to obtain a heading vector, which compromises between the tasks to reach the target position and to evade obstacles. Since the norm of the CAD increases with decreasing distance of object, a close obstacle will result in a vigorous evasion movement of the robot whereas a distant object does not influence the heading vector notably. Therefore, in the first case the heading vector will approximately correspond to the CAD, whereas in the latter case the heading vector will nearly coincide with the target vector (Meyer et al. 2016).

7.2.2. Navigation and Collision Avoidance

In order to obtain the target vector, the tracking system of the teleworkbench (Tanoto et al. 2012) is used. The tracking system is based on visual markers, of which one marker

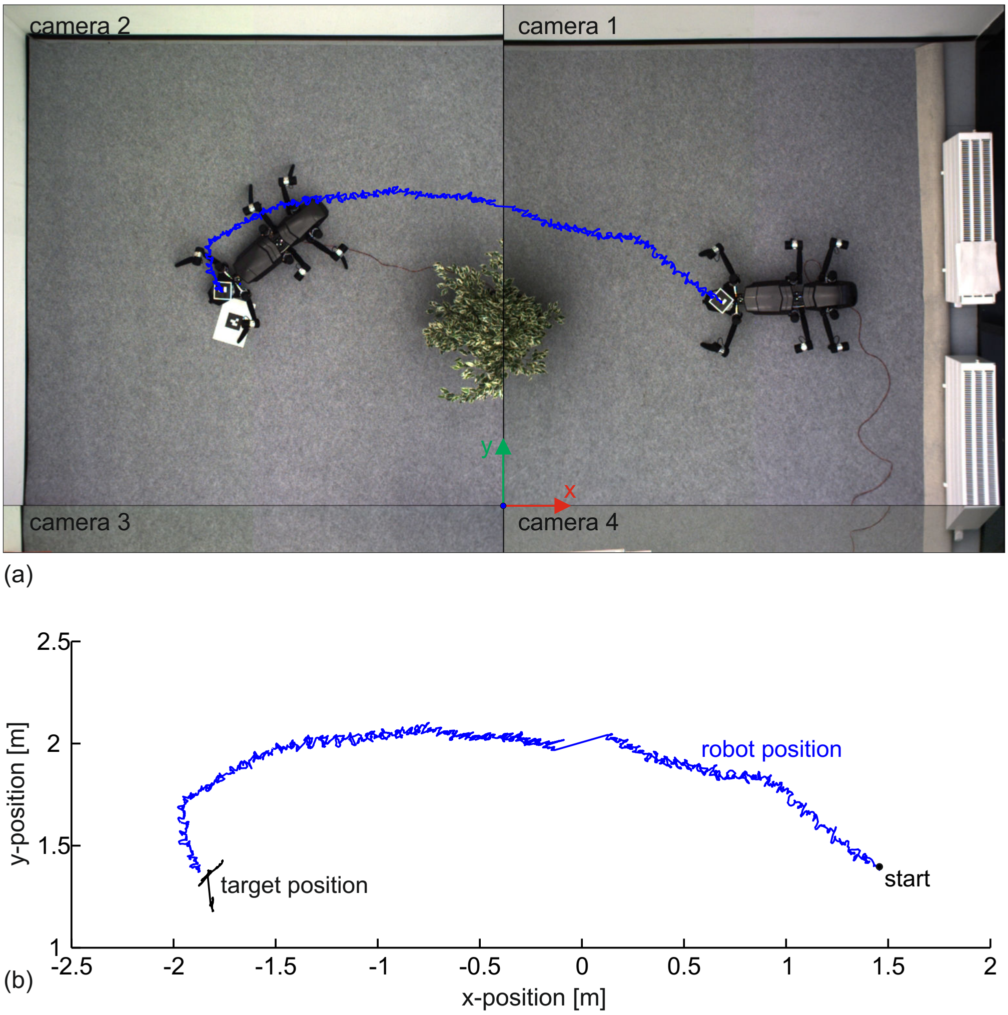


Figure 7.5.: Top view during experiment on collision avoidance. The robot is supposed to walk from its start position towards the marker on the left. Since an obstacle was placed in the direct trajectory the robot walks in a curve towards the target position. (a) shows the robot trajectory in pixel coordinates as overlay on images of the start (robot on right side in the field of view of camera 1) and end of the experiment (robot on the left side in the field of view of camera 2). (b) shows the trajectory of the robot in absolute coordinates.

is placed on the robot and another marker is used to define the target position. Based on the absolute positions and orientations of these markers the target vector is computed. The heading vector, as obtained from the optic flow processing, is used as input for the direction control of Hector (see section 5.3.2), thus pulling the virtual body model in

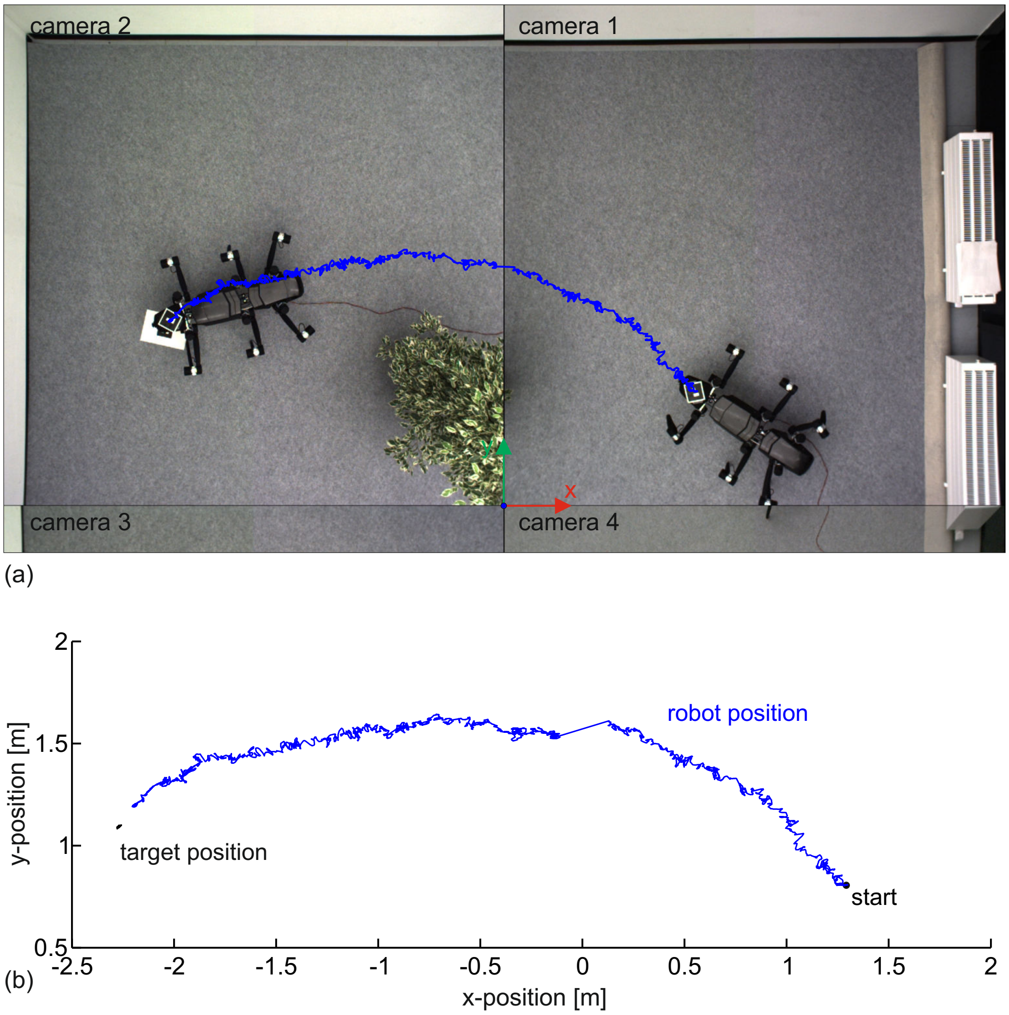


Figure 7.6.: Top view during experiment on collision avoidance. The robot is supposed to walk from its start position towards the marker on the left. Since an obstacle was placed in the direct trajectory the robot walks in a curve towards the target position. (a) shows the robot trajectory in pixel coordinates as overlay on images of the start (robot on right side in the field of view of camera 1) and end of the experiment (robot on the left side in the field of view of camera 2). (b) shows the trajectory of the robot in absolute coordinates.

the desired direction. Effectively, the robot is guided towards the target position, always avoiding the closest obstacles.

In figs. 7.5 and 7.6, results of two experiments are depicted, during which the robot was commanded to walk from the right side towards the marker on the left side. Between these positions, a plant was positioned as obstacle. Therefore, the robot had to walk around this obstacle to avoid collisions. Figure 7.3 shows the general setup of the scenario. The estimated fields of view of the four cameras overlap only close to the ground. This explains why the plant in fig. 7.6 does not appear in camera 1 as opposed to fig. 7.5 in which the plant is slightly shifted to the right and therefore turns up also in camera 1. The experiments were conducted without intervention by an operator. As the trajectory plots show, the robot is able to perform tight turns, e.g., in fig. 7.5 as the robot approaches the target position. Since the computation of the heading vector relies on noisy sensor input, also the direction of the heading vector fluctuates. However, oscillations visible in the trajectory plots can be attributed to the compliance of the system. Whenever a leg is lifted or lowered, the compliance in the stancing legs leads to movements of the main body. At the transition between cameras 1 and 2, the marker at the front segment of the robot is outside the field of vision of both cameras. Therefore, its position is not tracked. This leads to the straight line in the vicinity of x=0 m.

7.3. Stance Height Adaption for Locomotion in Rough Terrain

In section 5.3.1, the concept for stance height adaption is presented. It is based on a virtual network of springs and dampers that mediate changes of the stance height between the legs. If a leg steps on an obstacle, its stance height is reduced by the height of the obstacle. To compensate for this reduction, the stance heights of the neighboring legs are increased until an equilibrium is reached. In this equilibrium, the stance height of the leg on the obstacle is still lower than desired and the stance heights of the other legs are higher than desired. The tendencies of the legs to attain their desired stance heights counterbalance each other.

To test this terrain adaptation, the robot was positioned in front of a box with a height of 10 cm and a length of 42 cm, followed by a floor patch covered with coarse gravel. Using only the torsion sensors in the joint drives (see section 2.3.1), ground contacts and collisions were detected. Therefore, no additional sensors like tracking systems or explicit touch sensors were utilized for this test. Figure 7.7 shows results for this setup. Subfigures (a)–(f) show images of the robot throughout the experiment. In (g), the z-positions of the feet and the torsion of the β-drives of each leg are plotted. Due to the slanted setup of the α-drives in the robot, all joint drives of a leg tend to be loaded when stancing or during a collision in swing phase. However, in general, the β-drives are loaded the most during stance phases as this joint is predominantly responsible for levation and depression and therefore for the support of the robot. For the visualization, the torsion in all β-joints was set such that a positive torsion corresponds to a downward-directed force of the leg tip (thus indicating stance phases). In fig. 7.7, the swing phases are highlighted by gray background color. During these phases, the torsion of the β-joint (plotted in green) is usually low compared to the stance phases (white background). In fig. 7.7, the ascents onto the box and descents onto gravel are marked by vertical, red lines.

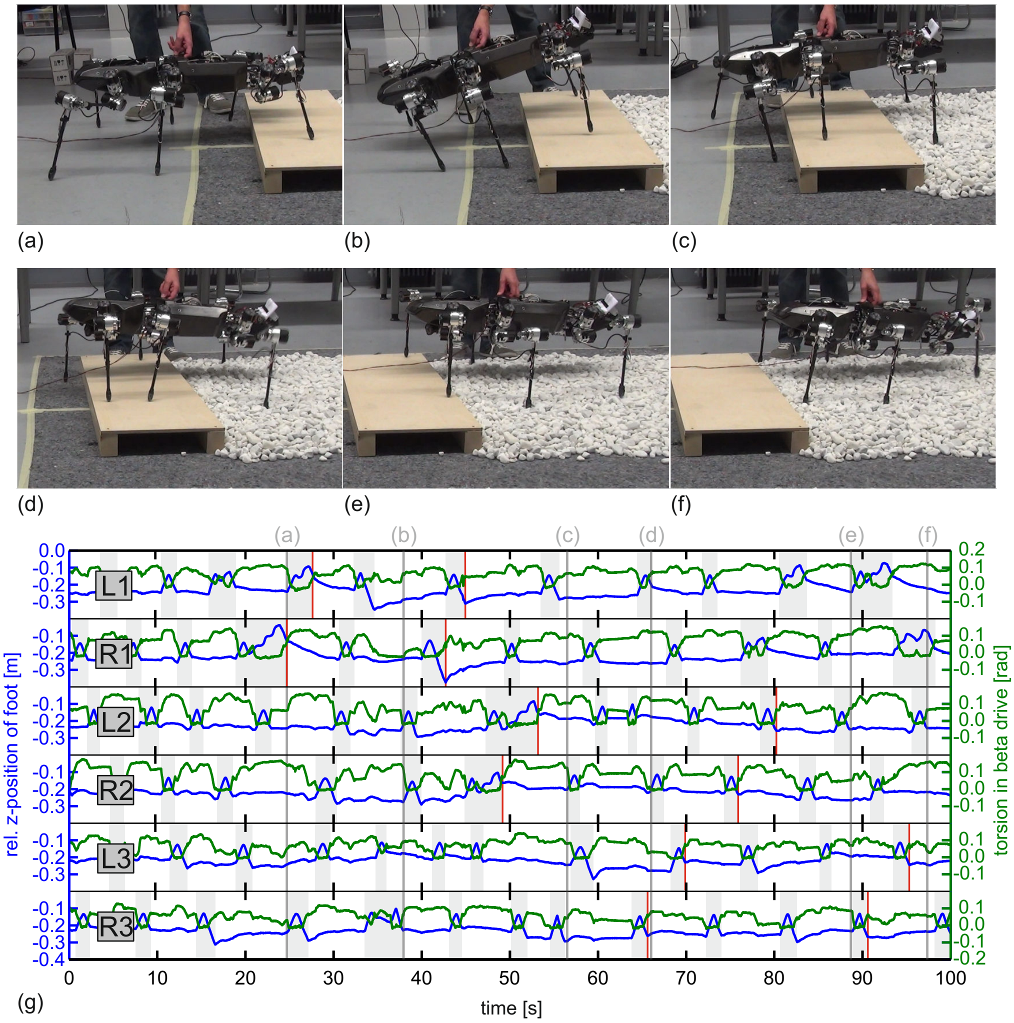


Figure 7.7.: Images and experimental data of robot climbing over an obstacle. (a)–(f) show images of the robot at different stages of the experiment (visible hand of the operator did not support the robot weight but remained close to the robot body to react in the case of a sudden topple over event). In (g), the z-positions (relative to the RCS) of the feet are plotted in blue. The torsions of the β-joint drives are shown in green. Swing phases are highlighted in gray. Ascents of the legs onto the box and descents onto gravel are marked by vertical, red lines. Note that the order of the legs is changed compared to the previously shown gait patterns. The points in time for which the robot is depicted in (a)–(f) are marked by vertical, gray lines (see also labels at top of (g) for allocation).

Summary

As shown in chapter 6, different parameter sets were obtained that achieve the demanded omnidirectional walking characteristics. Using one of these sets (PS_1) the walking controller was tested in various walking scenarios on the robot simulator. In this chapter, to prove the applicability on a real robot the controller was used for the control of Hector in different scenarios. During forward walking it could be shown that comparable gait patterns as for the simulation trials are generated.

Hector was designed to be equipped with different sensors to serve as a common, experimental platform. As a first test and to show the suitability of the walking controller for situations in which the heading direction changes dynamically, a camera system (advanced version of the system presented by Irwansyah et al. (2014)) was mounted in the front segment. The acquired video stream was evaluated by a bioinspired module based on EMDs to detect obstacles (Meyer et al. 2016). During the experiment, the robot was controlled by this module in order to reach a defined target position (the relative position was obtained by a tracking system (Tanoto et al. 2012)) and to evade any obstacles in its course. The results of two exemplary trials are depicted.

As a third test, the robot was commanded to climb over an obstacle. For this test, only the torsion of the joint drives was used to detect ground contact. In contrast to the previous experiments neither cameras nor external tracking were utilized. This experiment shows the suitability of the stance height controller to overcome obstacles, which are too big to be compensated solely by the inherent compliance of the joint drives.

8. Conclusion

Biological research and robotics have often been combined to verify biological concepts using real hardware or to use bioinspired control approaches in robotics. In the latter case, roboticists rely on the assumption that biological structures and controllers have co-evolved over millions of years, thus presenting an already optimized solution that can be transferred to the technical domain. However, since the optimization goals of the biological and the technical systems often differ, such a transfer must especially consider the differences between both systems to identify possible shortcomings. In insects, for example, one criterion for the optimization of the movement controller might have been the resource efficiency regarding the neuronal structures. Especially for flying insects this is relevant since a bigger brain means more mass that needs to be lifted. Therefore, a lighter insect with a less universal controller might have an evolutionary advantage. Although the weight is less critical in walking insects, a complex neuronal network still must be nourished, which requires more activity to find adequate nutrition. For example, *Carausius morosus* has evolved to a species that mimics dead twigs most of the time. Thus, they mislead predators and minimize their energy requirements. But although their nervous system is much simpler than that of mammals, they are still able to walk in uneven terrain, e.g., climbing along branches. Since the development of movement controllers is a current topic in robotics it is tempting to adapt the leg coordination of stick insects for the control of robots—especially due to its simplicity. However, as mentioned before, this demands a comparative analysis of the abilities and requirements of the biological model and the technical system to find a suitable level of abstraction for the transfer of the control concepts. In this thesis, this process is conducted for the development of an insect-inspired walking controller that is to be used on hexapod robots. As target system, the robot Hector was developed based on the model of the stick insect *Carausius morosus*.

robot design The design process of this robot is described in chapter 2. The robot emulates the morphology of the stick insect as its measures are roughly scaled up from the insect by a factor of 20. Since the robot was planned to be used as platform for various experiments, this scaling factor was chosen to allow for a rich sensorization and easy extension with additional modules. Beside the insect-like morphology, the robot also features an inherent compliance due to the integration of elastomer couplings in its joint drives. This compliance was implemented to emulate the elasticity of the biological model. However, beside these commonalities, there are many differences between the robot and the insect. Due to the increased size, the mass of the robot (13 kg) is necessarily much higher than that of the insect (<1 g). Also, since the abdomen of the insect was not replicated in the robot the mass distribution differs between the two systems. In the insect the COM is located approximately between the onsets of the hind legs. In the robot it is located close to the middle leg onsets.

Another major difference between the robot and its biological model that needs to be considered in the development of the walking controller is the ability of stick insects to cling to the substrate. This is accomplished by adhesive pads and claws. As a consequence the insects can climb vertically and even walk hanging upside-down. The robot on the other hand does not possess these features and therefore must maintain static stability to prevent tilting.

stability evaluation of the biological controller Although stick insects have usually been expected to maintain static stability when walking on even ground, Kindermann (2003) published experimental results that question this assumption. Therefore, to evaluate the direct applicability of the biological walking controller on a technical system like Hector, the walking patterns of stick insects were analyzed (see chapter 3). Based on kinematic data of walking stick insects that was recorded by Theunissen et al. (2014), the static stability was evaluated. Although some information (like the exact positions of the ground contacts) is missing, the results strongly suggest that stick insects do not permanently maintain static stability when walking on even ground. As observed by Kindermann (2003) the COM frequently leaves the support polygon if one of the hind legs is lifted. Usually the insects do not tilt since they can hold on to the substrate. Therefore, in the data by Theunissen et al. (2014), no tilting motion is observable. However, if the ground is loose, as in the experiments by Kindermann (2003), the insects topple backwards. Since Hector does not possess any means to cling to the ground the results imply that the insects' walking controller would not be suitable for a direct application on Hector.

stability evaluation of Walknet Based on observations of walking stick insects, Cruse (1990) developed Walknet, a walking controller that is able to reproduce the gait patterns of stick insects (see chapter 4). Walknet consists of six leg controllers, one for each leg. Within each of these leg controllers, ANNs are responsible for the generation of swing and stance trajectories. The coordination between these leg controllers and therefore the decision which legs are supposed to swing and which should perform stance movements, is organized decentrally using a set of coordination rules that act between neighboring legs. Especially this decentralized leg coordination was often adapted for the control of hexapod robots (see table 1.1). The general advantage of this bioinspired approach is the comparatively simple implementation and the inherent adaptability of the emerging gait patterns to the walking velocity. However, in most of these implementations, the main focus was the application of a bioinspired controller on a technical system. The general suitability for the application of the bioinspired controller on technical systems that require static stability was usually not evaluated.

Therefore, to test whether Walknet maintains static stability and therefore is suitable for the control of Hector, a physics simulation was used to evaluate the stability of the system during locomotion. The implementation that was used is based on the publication of Schilling, Hoinville, et al. (2013). Since the static stability of the system showed a strong dependency of the starting posture a systematic evaluation was conducted. During this evaluation, the strengths of the coordination rules were varied. Each parameter set was tested in three trials with different initial postures. For slow walking (0.15 m/s) coordination rule parameters were found that maintained the stability in all of these trials. However, during fast walking (0.25 m/s) permanent stability could not be achieved. Since only four of the parameters of Walknet were varied this test does not prove that the controller generally does not maintain static stability. If all parameters were optimized the system might be able to maintain stability even at higher speed. However, since Walknet does not include a module that supervises the stability, a further increase of walking speed is likely to result in instability.

Also, since it was derived from stick insects, which usually do not walk backwards, Walknet was originally designed only for forward walking. In its basic form also curve walking is only possible for big turning radii. Hector, however, is required to navigate within the constraints of interior space. Therefore, it must be able to perform tight turns and even walk backwards—both while maintaining static stability.

bioinspired control for technical systems Based on these findings a new walking controller was developed that incorporates the mentioned technical requirements (see chapter 5). Conceptually the leg coordination of this controller is based on the decentralized coordination rules of Walknet. Therefore, the individual leg controllers interact with each other to facilitate a switch to swing of their neighboring legs or to prolong their stance phases. However, in order to maintain stability, minimize disturbances during walking, and resolve deadlocks, the new controller features a central module that influences the individual leg controllers via additional, *technical coordination influences*. Also, in comparison to Walknet, the new controller is able to adapt the walking speed to the current situation. Whereas Walknet always achieved the desired velocity even at the cost of stability, the new controller can reduce the walking speed, e.g., if a leg gets close to the limits of its workspace or if the stability is at risk.

To facilitate omnidirectional walking the concept of *restrictedness* by Fielding and Dunlop (2004) was modified and utilized for the definition of the leg workspaces. In the original concept different restrictednesses (angular limits of leg joints, torque limits, etc.) were defined that indicate the “lack of freedom” (Fielding and Dunlop 2004) of a leg. Therefore, for each position of the leg tip the restrictedness of the leg could be specified—ranging from 0 ≙ *far away from workspace limits* to 1 ≙ *at or beyond workspace limts*. To allow simple combination of the different restrictions the *restrictedness*-measure was reversed, thus creating the *unrestrictedness*—ranging from 0 ≙ *at or beyond workspace limts* to 1 ≙ *far away from workspace limits*. The advantage of this reversal is that multiple *unrestrictedness*-components can be multiplied to obtain a combined measure. Thus, as long as the unrestrictedness of the leg is higher than zero the leg tip is within its workspace. To account for the different factors that must be considered during the phases of leg movement, combined measures of unrestrictedness were defined for the swing and stance movements and for the selection of swing targets. For example, during the stance movement the torque restrictions of the joint drives must be considered whereas these restrictions are negligible during the swing movement since the leg does not support the weight of the body.

Due to the flexible combination of unrestrictednesses for swing phase, stance phase, and swing-target selection, workspaces with complex boundaries are generated that continuously adapt to the current walking situation. This enables the robot to perform movements that use the legs to their full technological capacities. Compared to this concept the workspace definitions used by Espenschied, Quinn, Chiel, et al. (1995) and Görner, Wimböck, and Hirzinger (2009) simply reduce the leg workspaces to vertical cylinders which confines the movement ranges of the legs in general.

To combine the bioinspired leg coordination with the unrestrictedness-based definition of workspaces the originally position-dependent selection of swing or stance mode within the leg controllers was replaced by a time-dependent measure. Therefore, during stance phase the decision when to switch to swing phase is based on the estimation of the remaining time until the leg would reach the limits of its workspace. If this duration is longer than a given threshold the leg remains in stance phase—if it is shorter the leg switches to swing. To coordinate the legs, the individual time thresholds for this comparison are altered by the neighboring legs. This allows the generation of free gaits under consideration of the leg workspaces as defined by the technical constraints.

In chapter 6, the parameters of this new controller are varied to find sets of parameters that also reproduce the gait patterns observed in stick insects. For this purpose, three measures were defined for wave gait, tetrapod gait, and tripod gait that quantify the regularity of the gaits. In a physics simulation, 80.000 different parameter sets were tested. Of these, six sets showed the insect-like sequence of gaits that was expected for forward walking: During slow walking a wave gait emerges. For faster walking the gait pattern changes to tetrapod and for even faster velocities a tripod gait can be observed.

However, beside forward walking—which was already achieved with the classical Walknet—the new controller could be shown to master also backwards, tight curve, and sidewards walking. The foundations for this omnidirectional walking capability are the adaptable leg workspaces in combination with the bioinspired leg coordination. For the realization of omnidirectional stance trajectories a direction control similar to the body model - based concept by Schilling, Paskarbeit, Schmitz, et al. (2012) was utilized. Beside the generation of stance trajectories it is also used for the selection of suitable target positions for the swing movement.

experimental verification To prove the suitability of the new walking controller for application on Hector, first the basic, omnidirectional movements were performed that had already been shown in simulation (see section 7.1). Although the regularity of the gait patterns does not achieve the same level as in the simulation the robot nevertheless adapted to the desired walking speed. During all test trials the stability was maintained.

In order to test the walking controller in a real experimental setup an external controller was used to generate heading vectors (see section 7.2). This controller uses optic flow to estimate the distances of obstacles. Based on this data a local walking trajectory is generated that always avoids the closest obstacles. To control the walking direction online this controller only sent the desired walking direction to the walking controller. Using external, optical tracking that was used for the definition of a target position the robot was able to avoid an obstacle that stood between its start and goal positions.

In addition, results of basic climbing abilities were shown (see section 7.3). During an experiment the robot was commanded to climb over an obstacle that stood in its way. For this demonstration no external information was used. Instead, only the load estimation of the compliant couplings in the joint drives was used to predict ground contacts. The adaptation of the legs' stance heights was conducted using a network of virtual spring and dampers (see section 5.3.1).

Future Work

In this thesis the focus of research was on the leg coordination during walking and the realization of insect-like gait patterns on a technical system. With the setup of a bioinspired, compliant robot, the evaluation of insect and robot gaits, and the development a walking controller there are still open topics that can be addressed in future work. Some of these topics would require a redesign of the robot mechanics to optimize the leg setup or to integrate locking mechanisms in the joint drives. In the following paragraphs possible enhancements of the robot and the walking controller are listed.

reduction of energy requirements An advantage of the insect-like setup of Hector is the sprawled posture since the lateral positioning of the leg tips increases the stability of the system. In comparison, most quadruped robots use leg setups in which the distances between the leg tips corresponds to the distances of the leg onsets. Their legs therefore are rather vertically oriented instead of the lateral setup of Hector. The advantage of the vertical setup are the small lever arms in the leg kinematics. The energy that is required for standing is therefore minimal since the forces are structurally compensated. In Hector, however, even standing requires especially the β-joint drives to exert torques in the same magnitude as during walking.

To reduce this energy consumption different modifications of the joint mechanics are possible. One option is to use a gearbox that is not backdrivable. Thus, if the output of the joint is loaded, e.g., during standing, this does not result in a torque transmission to the motor at the input side. Another option would be to integrate a locking mechanism in the joint that passively blocks movements of the motor.

increase joint torque In order to realize bigger leg workspaces, the torque limits of the joints must be increased. Currently, the acceptable peak torques are limited by the elastomer coupling and the gearbox. For higher torques the elastomer will deform permanently, thus resulting in backlash. To increase the maximum load of the coupling the elastomer could be bonded to the metal hubs. Thus, the elastomer would not only transmit load via the compressed but also via the stretched lobes. Although this would increase the maximum load a plastic deformation might still appear. To rule out this elastomer inherent plasticity metal springs could be used.

A different gear box design could further increase the maximum peak torque. For example, cycloid gears could be used as replacement for the harmonic drive gearboxes (Sensinger and Lipsey 2012).

Beside these factors, the average torque the drive is able to exert is limited also by the motor. To prevent permanent damage the maximum winding temperature must not be exceeded (125°C for the motors in the joint drives). Therefore, the maximum power of the motor is limited by the thermal coupling to the environment. In HECTOR, external rotors are used since they achieve a high torque at a small length. However, since most of the heat in the current joint setup will be dissipated via a plate between the motor and the gearbox the maximum heat dissipation is limited. To reduce the thermal resistance an internal rotor could be used since this would allow a direct coupling of the stator to the housing.

optimize body morphology For HECTOR the stick insect geometry was roughly scaled up by a factor of 20. Also the orientations of the joint axes were transferred from the model to the robot. This was conducted to achieve a high level of conformity between both systems, e.g., to conduct load measures at the leg segments that can be retroceded for conclusions in animal behavior (Schäffersmann et al. 2014).

However, if the similarity of the robot to a biological model is not relevant for the intended tasks the positioning and the orientation of the legs could be optimized for various quality functions. If, for example, locking mechanisms or non-backdrivable gearboxes were integrated in the joint drives the orientation of the α-joints could be optimized to reduce the movements of the β-drives during stance phases, thus also reducing the energy consumption.

integration of body joint drives The body segments of HECTOR are connected via cardan joints that can be actuated by spindle drives. Although the mechanics of the body joint drives is functional, the control of the spindle drives still needs to be implemented. To utilize these additional DoFs in the control of walking, e.g., the stance module would need to be expanded to adapt the body posture to the terrain and the walking direction.

optimization of coordination influences For the data shown in chapters 6 and 7 the coordination influences were configured to act in the same directions as their WALKNET-counterparts did. Therefore, e.g., influences 1 and 2 act anteriorly whereas influence 3 acts posteriorly. To optimize the walking behavior the directions of influence could be modified as well. For example, in different implementations of WALKNET for application on robots coordination rule 1 was configured to act in all directions between neighboring legs, thus preventing these legs to swing simultaneously.

For stick insects a change of strengths and direction of the coordination rule was detected during curve and backwards walking (Dürr 2005; Schmitz and Düsterhus 2009).

Following the biological example also the strengths of the coordination influences for the control of a robotic system could be adapted to the direction of walking.

adaptation of home positions For the selection of the swing target, home positions are defined in the workspaces of the legs. The target point is selected such that the following stance movement would pass the home positions. In the presented experiments fixed home positions were used. To facilitate also climbing on steep slopes the home positions should be adapted to the respective situation since a fixed home position might even be outside of the leg workspace if the vector of gravity is slanted relative to the LCS.

swing trajectory generator As shown in the experiments the swing trajectory generator presented in this thesis is able to control the swing movement. However, as already discussed in chapter 5, the leg workspace definitions based on the unrestrictednesses are not included in the setup of the vector field which is used to compute the direction of movement. Therefore, whenever the leg reaches the limit of its workspace a mechanism is activated that tries to move the leg closer to its goal position without further consideration of the vector field. Although this approach solves most of these problems a compact solution would be preferable in which the trajectory is generated directly under consideration of the workspace limits.

compliant stability measure For the control of HECTOR the ESM by Messuri (1985) was used. During the conducted experiments the static stability of the robot was maintained. However, since the ESM assumes a stiff robot it cannot guarantee the stability of a compliant system. Therefore, in order to ensure the stability in all situations, a stability measure is required that considers the compliance of the joint drives.

cognitive extension Using the presented mechanisms many walking situations can be tackled without intervention of an operator. In some cases, however, these approaches will inevitably fail. For example, a leg might not find a ground contact at the end of its swing phase since the leg stepped into a hole. In this case the leg would extend downwards until the limit of its workspace. If this happens to a middle leg the robot will likely come to a halt since as soon as the hind leg of the same side reaches the limit of its workspace and cannot be lifted since this would compromise the stability. A solution could be to reverse the walking direction, find new footholds and try again to overcome the hole. To automate this process a cognitive extension as proposed by Schilling and Cruse (2017) could be used.

Appendices

A. Kinematics

A.1. Forward Kinematics

The forward kinematics of the legs was defined based on the DH (Denavit-Hartenberg) convention (Denavit and Hartenberg 1955; Paul 1981). In the following formulae, the convention of Craig (2004) will be used. For the explanation of this notation, a serial robotic manipulator is assumed that is built up from rigid links and revolute joints that connect these links.[1] According to this convention, local coordinate systems are defined for each link of the robotic manipulator. In this specific case, as depicted in fig. A.1, the coordinate frames are located at the end of the *coxa* (F_α), the *femur* (F_β), and the *tibia* (F_γ).

A central RCS (Robot Coordinate System) F_{RCS} is fixed to the hind segment of the body. Its origin is located between the hind leg onsets. Its x-axis points to the front, its z-axis points upwards, and the y-axis correspondingly points to the left of the robot. To simplify the leg kinematics, for each leg a LCS (F_{LCS}) is defined that has the same orientation as F_{RCS}, but is located at the onset of the leg. The offset of each of these relative to F_{RCS} is given in table A.1.

A.1.1. Denavit-Hartenberg Convention

The choice of the location and orientation of the coordinate systems F_i, spanned by the axes x_i, y_i, and z_i, is based on the following rules (Craig 2004):

1. The z_{i-1}-axis points in the direction of the rotational axis of joint i.

2. Locate the origin O_i of the coordinate system F_i at the intersection of axis z_i with the common normal of z_{i-1} and z_i. If the two z-axes are parallel, the common normal is not uniquely defined. In this case, the position of O_i can be chosen freely along z_i.

3. Orient the x_i-axis parallel to the common normal of the z-axes of reference frames F_{i-1} and F_i ($x_i = z_{i-1} \times z_i$), pointing away from O_{i-1}. Again, this is not uniquely defined if the z-axes are parallel.

4. y_i is chosen such that it forms a right-handed coordinate system together with x_i and z_i.

[1] Although the DH-convention includes prismatic joints, they will not be considered since the robot contains only rotational joints.

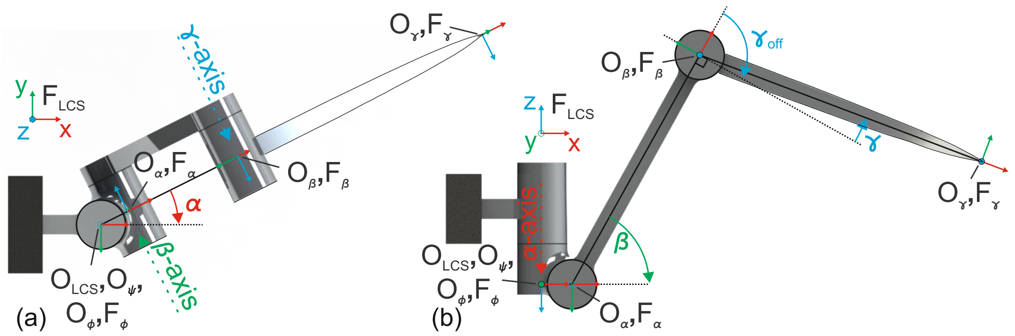


Figure A.1.: Overview of the coordinate frames used for the kinematics. The origins of the coordinate frames are labeled by O_x, the frames giving the orientation of the x-,y-, and z-axes are labeled by F_x. (a) shows a leg in top view, (b) in side view.

Using these rules, the transformation from coordinate system F_{i-1} to coordinate system F_i can be described by the four parameters d_i, θ_i, r_i, and α_i.[2]

- d_i is the distance along z_{i-1} to the common normal of z_{i-1} and z_i.
- θ_i is the rotational angle about axis z_{i-1} from x_{i-1} to x_i.
- a_i is the shortest Euclidean distance between z_{i-1} and z_i (the length of the connecting common normal vector).
- α_i is the angle about the common normal of z_{i-1} and z_i that vector z_{i-1} makes with z_i.

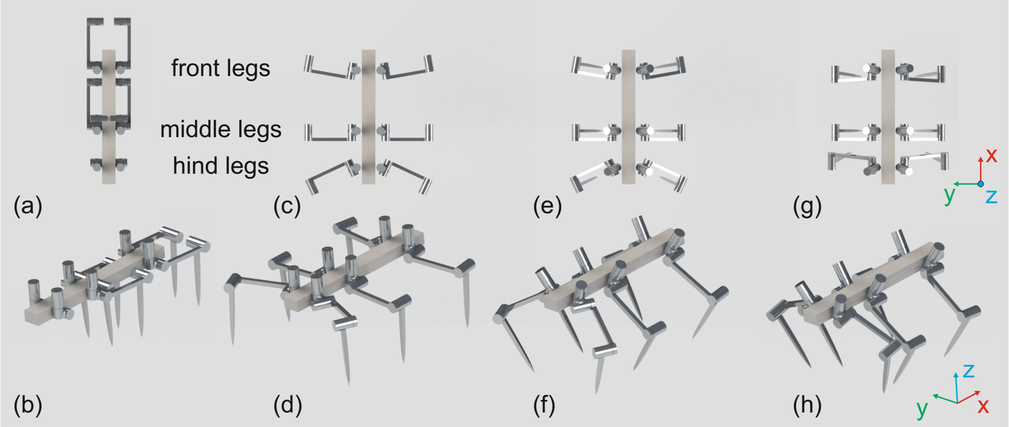
With these parameters, the transformation matrix ${}^{i-1}\boldsymbol{T}_i$ from F_{i-1} to F_i can be written as

$$
{}^{i-1}\boldsymbol{T}_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} . \tag{A.1}
$$

A.1.2. Forward Kinematics of the Legs

The starting points of the leg kinematics are the local F_{LCS}. The orientations of these coordinate systems are fixed to the respective body segments and for a straight alignment of the body segments, the F_{LCS}s have the same orientation as F_{RCS} with different origins: The origin of each F_{LCS} is located at the intersection of the rotational axis of the respective α-joint with the common normal of the leg's α- and β-joint rotational axes. In many six-legged robots, the α-axes are simply vertical (see table 1.1), therefore parallel

[2] Originally, the parameter r_i was named a_i (Denavit and Hartenberg 1955). In order to make it easier to discriminate from α_i, it is often renamed.



	(a,b)	(c,d)	(e,f)	(g,h)
ϕ	0°	(±80°,±90°,±115°)		
ψ	0°	0°	−30°	
α_{off}	0°	0°	0°	(∓11.5°,0°,±28.3°)

Figure A.2.: Orientation of the legs for different ϕ and ψ angles and offsets α_{off}. The respective angles are given in the table. If three values are given, they refer to the front, middle, and hind leg values, respectively. The upper row (subfigures a,c,e,g) shows the robot from top, the lower row (subfigures b,d,f,h) shows it from a skewed perspective.

to F_{RCS}'s and F_{LCS}'s z-axes. In stick insects, however, the α-axes are slanted relative to the vertical-axis (see, e.g., (Cruse 1976b)). These slanted leg setup was adapted in the design of Hector. The orientation of the α-axis is defined by the ϕ- and ψ-angles. In addition, to allow for easier mounting of the α-drive to the robot housing, the zero-position of the α-drive was rotated about F_{ψ}'s z-axis (that coincides with the rotation axis of α) by α_{off}.

The results of the two transformations that are necessary to slant the leg from a rectangular orientation with vertical α-axes to the slanted insect-inspired orientation are visualized in figs. A.2 and A.3. The rotation by ϕ turns the legs outwards. The rotation by ψ turns the legs downwards. The last image, fig. A.2 (c), visualizes the offset α_{off}.

For the actuators, the direction of positive rotation was defined according to the right-hand rule for a rotation vector pointing from the motor side to the output of the actuator. To consolidate the direction of rotation of the θ-angle of the DH-notation with the actual rotational direction of the actuators, the z-axes of the three reference frames for *coxa*, *femur*, and *tibia* will be aligned such that a positive change of the respective θ results in a positive change of the joint angle. Strictly following the DH-notation, the zero position of the γ-joint would correspond to a stretched leg posture (with femur and tibia parallel). Since the angular workspace of the joint drives is symmetric and

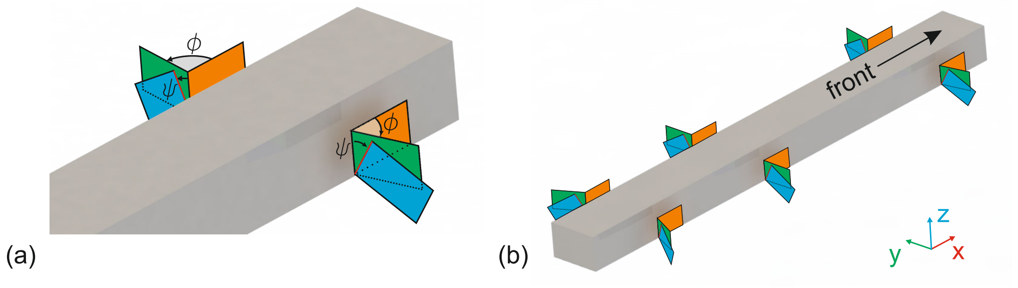


Figure A.3.: Visualization of the leg plane rotations due to the ϕ and ψ angles. The leg plane is the plane in which the leg tip moves at $\alpha = 0°$ for various β- and γ-angles. The final α-axes are marked in red. (a) shows a closeup of the front leg planes. (b) shows an overview of all leg planes.

limited to ±135° regarding the mechanically determined zero position (see section 2.3), the mounting according to the DH-notation would prevent the leg from reaching points located close to the β-joint. On the other hand, this would allow the leg to reach further upwards. However, since this feature is not required for the intended operation, it was decided to mount the γ-drive with an offset of ±90°. Thus, the leg can reach further inwards. In the notation of the forward kinematics, this is accounted for by the introduction of the angular offset γ_{off} (see fig. A.1). The DH-parameters for the forward kinematics are listed in table A.1. The complete forward kinematics to transform the coordinates of the tip of the *tibia* into the LCS would be:

$$\begin{pmatrix} \boldsymbol{p}_{\text{LCS}} \\ 1 \end{pmatrix} = {}^{\text{LCS}}\boldsymbol{T}_{\phi} \cdot {}^{\phi}\boldsymbol{T}_{\psi} \cdot {}^{\psi}\boldsymbol{T}_{\alpha} \cdot {}^{\alpha}\boldsymbol{T}_{\beta} \cdot {}^{\beta}\boldsymbol{T}_{\gamma} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{A.2}$$

A.2. Inverse Kinematics

To compute the required joint angles to reach a given position with the end effector, the inverse kinematics must be solved. Since the leg could—at least in principle—reach many points with two different postures (see fig. A.4), of which only one is desired, the inverse kinematics is supposed to return only the “correct” solutions.

For the definition of the inverse kinematics, an end effector point P_3 that is defined in reference frame F_{ψ} (see fig. A.5) is assumed. A point $P_{3,\text{LCS}}$ given in coordinates of F_{LCS} can be transformed by $P_{3,\psi} = {}^{\phi}\boldsymbol{T}_{\psi}{}^{-1} \cdot {}^{\text{LCS}}\boldsymbol{T}_{\phi}{}^{-1} \cdot P_{3,\text{LCS}}$ to a representation in F_{ψ}.

In a first step, the α^* angle (see fig. A.5 (a)) is computed based on the x- and y-

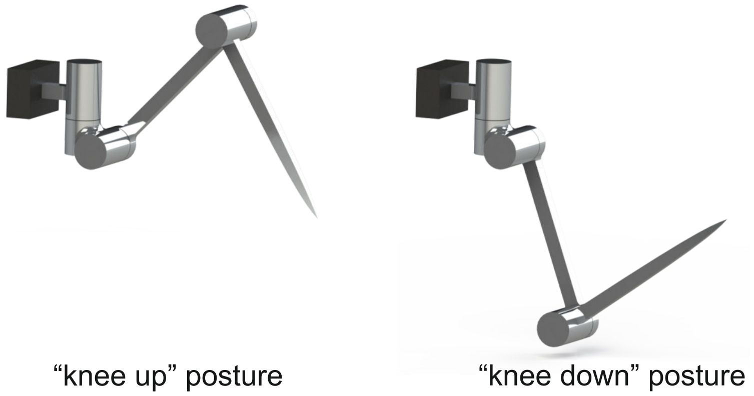


Figure A.4.: "knee up" and "knee down" postures of the leg that reach the same point. During walking, the leg tip should be the lowest point of the leg's kinematic chain, which is given for the "knee up" posture shown on the left side. In the "knee down" posture, which is shown on the right side, the γ-drive is the lowest point and during stance phase it would likely get damaged.

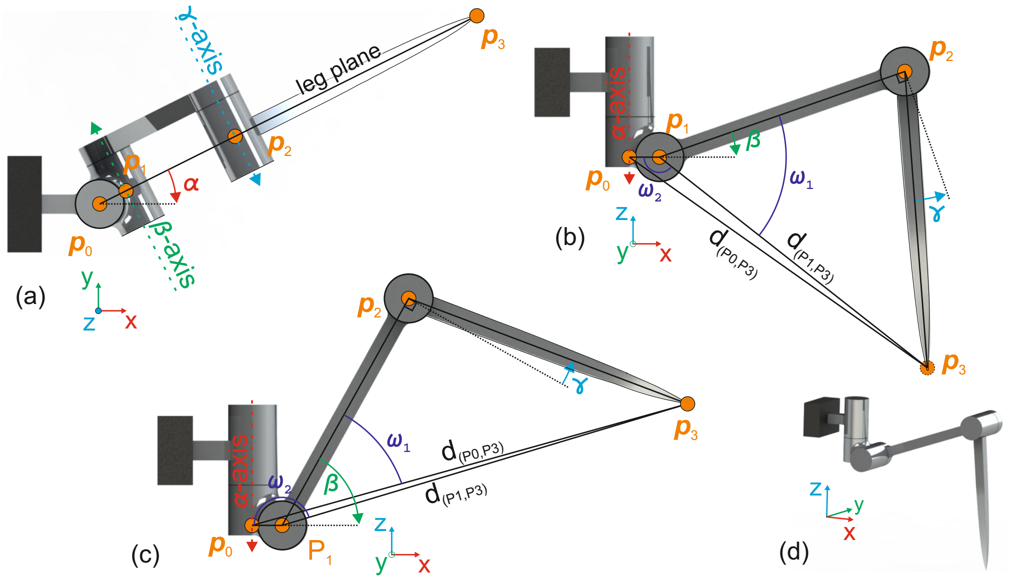


Figure A.5.: Illustration of the variables used in the solution of the inverse kinematics. (a) shows the leg in top-view, (b) and (c) are side-views of the leg, and (d) is a skewed perspective.

Table A.1.: Relative positions of the leg onsets and DH-parameters for the kinematics

		front		middle		hind	
		left	right	left	right	left	right
	Δx	578 mm		218 mm		0 mm	
	Δy	70 mm	-70 mm	70 mm	-70 mm	70 mm	-70 mm
	Δz	0 mm		0 mm		0 mm	
	α_{off}	$-11.5°$	$11.5°$	$0°$	$0°$	$28.3°$	$-28.3°$
	γoff	$-90°$	$90°$	$-90°$	$90°$	$90°$	$-90°$
${}^{\text{LCS}}\boldsymbol{T}_\phi$	$\theta_{\text{DH}}\,(\phi)$	$80°$	$-80°$	$90°$	$-90°$	$115°$	$-115°$
	d_{DH}	0 mm		0 mm		0 mm	
	a_{DH}	0 mm		0 mm		0 mm	
	α_{DH}	$-90°$	$90°$	$-90°$	$90°$	$-90°$	$90°$
${}^{\phi}\boldsymbol{T}_\psi$	$\theta_{\text{DH}}\,(\psi)$	$30°$	$-30°$	$30°$	$-30°$	$30°$	$-30°$
	d_{DH}	0 mm		0 mm		0 mm	
	a_{DH}	0 mm		0 mm		0 mm	
	α_{DH}	$-90°$	$90°$	$-90°$	$90°$	$-90°$	$90°$
${}^{\psi}\boldsymbol{T}_\alpha$	θ_{DH}	$\alpha + \alpha_{\text{off}}$	$\alpha + \alpha_{\text{off}}$	$\alpha + \alpha_{\text{off}}$	$\alpha + \alpha_{\text{off}}$	$\alpha + \alpha_{\text{off}}$	$\alpha + \alpha_{\text{off}}$
	d_{DH}	0 mm		0 mm		0 mm	
	$a_{\text{DH}}\,(l_{\text{cox}})$	30 mm		30 mm		30 mm	
	α_{DH}	$90°$	$-90°$	$90°$	$-90°$	$-90°$	$90°$
${}^{\alpha}\boldsymbol{T}_\beta$	θ_{DH}	β	β	β	β	β	β
	d_{DH}	0 mm		0 mm		0 mm	
	$a_{\text{DH}}\,(l_{\text{fem}})$	260 mm		260 mm		260 mm	
	α_{DH}	180°		180°		180°	
${}^{\beta}\boldsymbol{T}_\gamma$	θ_{DH}	$\gamma + \gamma_{\text{off}}$	$\gamma + \gamma_{\text{off}}$	$\gamma + \gamma_{\text{off}}$	$\gamma + \gamma_{\text{off}}$	$\gamma + \gamma_{\text{off}}$	$\gamma + \gamma_{\text{off}}$
	d_{DH}	0 mm		0 mm		0 mm	
	$a_{\text{DH}}\,(l_{\text{tib}})$	300 mm		300 mm		300 mm	
	α_{DH}	0°		0°		0°	

coordinates of P_3:

$$\alpha^* = \operatorname{atan2}(y_{P_3}, x_{P_3}) \quad \text{with} \tag{A.3}$$

$$\operatorname{atan2}(y_{P_3}, x_{P_3}) = \begin{cases} \arctan(\frac{y_{P_3}}{x_{P_3}}) & \text{if } x_{P_3} > 0, \\ \arctan(\frac{y_{P_3}}{x_{P_3}}) + \pi & \text{if } x_{P_3} < 0 \text{ and } y_{P_3} \geq 0, \\ \arctan(\frac{y_{P_3}}{x_{P_3}}) - \pi & \text{if } x_{P_3} < 0 \text{ and } y_{P_3} < 0, \\ +\frac{\pi}{2} & \text{if } x_{P_3} = 0 \text{ and } y_{P_3} > 0, \\ -\frac{\pi}{2} & \text{if } x_{P_3} = 0 \text{ and } y_{P_3} < 0, \\ \text{undefined} & \text{if } x_{P_3} = 0 \text{ and } y_{P_3} = 0. \end{cases} \tag{A.4}$$

Applying the α-offset, the actual joint angle can be computed:

$$\alpha = \alpha^* - \alpha_{\text{off}} \tag{A.5}$$

With the α-angle, P_1, the origin of F_α, can be computed using the forward kinematics:

$$\begin{pmatrix} \boldsymbol{p}_1 \\ 1 \end{pmatrix} = {}^{\psi}\boldsymbol{T}_\alpha \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{A.6}$$

This can be used to compute the distance between P_1 and P_3:

$$d_{(P_0,P_3)} = \|P_3 - P_0\| \tag{A.7}$$

$$d_{(P_1,P_3)} = \|P_3 - P_1\| \tag{A.8}$$

Applying the law of cosines, γ^* can be computed:

$$\gamma^* = \arccos\left(\frac{l_{\text{tib}}^2 + l_{\text{fem}}^2 - d_{(P_1,P_3)}^2}{2\, l_{\text{fem}}\, l_{\text{tib}}}\right) - \frac{\pi}{2} \tag{A.9}$$

To compute β, the two angles ω_1 and ω_2 must be computed first, again applying the law of cosines.

$$\omega_1 = \arccos\left(\frac{l_{\text{fem}}^2 + d_{(P_1,P_3)}^2 - l_{\text{tib}}^2}{2\, l_{\text{fem}}\, d_{(P_1,P_3)}}\right) \tag{A.10}$$

$$\omega_2 = \arccos\left(\frac{l_{\text{cox}}^2 + d_{(P_1,P_3)}^2 - d_{(P_0,P_3)}^2}{2\, l_{\text{cox}}\, d_{(P_1,P_3)}}\right) \tag{A.11}$$

To find only the "correct" posture for a desired end effector position, different cases must be distinguished:[3]

$$\beta^* = \omega_1 - \omega_2 + \pi \qquad \text{for} \quad z_{P_{3,\psi}} < 0 \quad \text{(see fig. A.5 (c))} \tag{A.12}$$

$$\beta^* = \omega_1 + \omega_2 - \pi \qquad \text{for} \quad z_{P_{3,\psi}} > 0 \quad \text{(see fig. A.5 (b))} \tag{A.13}$$

$$\beta^* = \omega_1 \qquad \text{for} \quad z_{P_{3,\psi}} = 0 \quad \text{(not depicted)} \tag{A.14}$$

To account for the different orientations of the β- and γ-joints (output directed to the front or to the back of the robot), the direction of rotation must be adapted. To obtain the real joint angles, the intermediate angles β^* and γ^* must be multiplied by -1 for certain configurations.

	front		middle		hind	
	left	right	left	right	left	right
β	$-\beta^*$	β^*	$-\beta^*$	β^*	β^*	$-\beta^*$
γ	γ^*	$-\gamma^*$	γ^*	$-\gamma^*$	$-\gamma^*$	γ^*

[3] Note that the z-axis in F_ψ is directed downwards!

A.3. Velocity Kinematics and Force-Torque Relationship

To compute the translatory velocity in Cartesian space for given joint velocities, the position of the end effector as established in eq. (A.2) must be derived after the joint angles. This results in the Jacobian matrix:

$$\boldsymbol{J} = \frac{\mathrm{d}\boldsymbol{p}_{\mathrm{LCS}}}{\mathrm{d}\boldsymbol{q}} = \left[\frac{\partial \boldsymbol{p}_{\mathrm{LCS}}}{\partial \alpha}, \frac{\partial \boldsymbol{p}_{\mathrm{LCS}}}{\partial \beta}, \frac{\partial \boldsymbol{p}_{\mathrm{LCS}}}{\partial \gamma}\right] = \begin{bmatrix} \frac{\partial x}{\partial \alpha} & \frac{\partial x}{\partial \beta} & \frac{\partial x}{\partial \gamma} \\ \frac{\partial y}{\partial \alpha} & \frac{\partial y}{\partial \beta} & \frac{\partial y}{\partial \gamma} \\ \frac{\partial z}{\partial \alpha} & \frac{\partial z}{\partial \beta} & \frac{\partial z}{\partial \gamma} \end{bmatrix} \tag{A.15}$$

with $\boldsymbol{q} = (\alpha, \beta, \gamma)$. By multiplying eq. (A.15) with $\dot{\boldsymbol{q}} = d\boldsymbol{q}/dt$, the velocity of the end effector $\boldsymbol{v}$ can be directly deduced:

$$\dot{\boldsymbol{p}}_{\mathrm{LCS}} = \boldsymbol{v} = \boldsymbol{J} \cdot \dot{\boldsymbol{q}} \tag{A.16}$$

This relates the joint velocities to the end effector velocity.

Using the concept of *virtual work*, a relationship between the forces exerted at the end effector and the torques in the joints can be derived. The required work for a movement of the end effector by can be written as $\mathrm{d}W = \boldsymbol{f}^T \cdot \mathrm{d}\boldsymbol{r}$ with the counteracting force $\boldsymbol{f}$. For the same movement, the work in the joints can be formulated as $\mathrm{d}W = \boldsymbol{\tau}^T \cdot \mathrm{d}\boldsymbol{q}$ with the counteracting torque $\boldsymbol{\tau}$. Since the work for the movement of the end effector must be the same, regardless whether it is computed based on the end effector or the joint movements, the right sides of the corresponding equations are equal:

$$\boldsymbol{f}^T \cdot \mathrm{d}\boldsymbol{r} = \boldsymbol{\tau}^T \cdot \mathrm{d}\boldsymbol{q} \tag{A.17}$$

By rearranging eq. (A.16), the relation between an infinitesimal change of position of the end effector and an infinitesimal change of joint angles can be obtained:

$$\boldsymbol{v} = \boldsymbol{J} \cdot \dot{\boldsymbol{q}} \tag{A.18}$$

$$\Rightarrow \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = \boldsymbol{J} \cdot \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} \tag{A.19}$$

$$\Rightarrow \mathrm{d}\boldsymbol{r} = \boldsymbol{J} \cdot \mathrm{d}\boldsymbol{q} \tag{A.20}$$

Replacing $\mathrm{d}\boldsymbol{r}$ by $\boldsymbol{J} \cdot \mathrm{d}\boldsymbol{q}$ (according to eq. (A.20)) in eq. (A.17) gives

$$\boldsymbol{f}^T \cdot \boldsymbol{J}\mathrm{d}\boldsymbol{q} = \boldsymbol{\tau}^T \cdot \mathrm{d}\boldsymbol{q} \quad . \tag{A.21}$$

By eliminating $\mathrm{d}\boldsymbol{q}$ and transposing both sides of the equation, the relationship between the force $\boldsymbol{f}$ exerted by the end effector and the torques $\boldsymbol{\tau}$ exerted by the joints can be obtained:

$$\boldsymbol{\tau} = \boldsymbol{J}^T \cdot \boldsymbol{f} \quad \text{or} \tag{A.22}$$

$$\boldsymbol{f} = \boldsymbol{J}^{-T} \cdot \boldsymbol{\tau} \tag{A.23}$$

A.4. Singularity of the Forward Kinematics

A singularity in the forward kinematics occurs if the determinant of the Jacobian is zero:

$$\det(J) = 0 \tag{A.24}$$

The determinant of the Jacobian (eq. (A.15)) is

$$\det(J) = l_{\text{fem}} l_{\text{tib}} \left(l_{\text{cox}} + l_{\text{fem}} \cos(\beta) + l_{\text{tib}} \cos\left(\beta - \gamma - \gamma_{\text{off}}\right)\right) \sin(\gamma + \gamma_{\text{off}}) \quad . \tag{A.25}$$

Assuming $l_{\text{fem}} \neq 0$ and $l_{\text{tib}} \neq 0$, solving eq. (A.24) for γ yields:

$$\gamma = \beta \pm \operatorname{acos}\left(\frac{-l_{\text{cox}} - l_{\text{fem}} \cos(\beta)}{l_{\text{tib}}}\right) - \gamma_{\text{off}} + 2\, n\, \pi \quad , \tag{A.26}$$

$$\gamma = -\gamma_{\text{off}} + 2\, n\, \pi \quad \text{and} \tag{A.27}$$

$$\gamma = -\gamma_{\text{off}} + \pi + 2\, n\, \pi \quad \text{with } n \in \mathbb{Z} \tag{A.28}$$

Since the γ-joint will be operating in the range $]-\frac{\pi}{2}, \frac{\pi}{2}[$, eqs. (A.27) and (A.28) do not apply. Inserting eq. (A.26) into the forward kinematics for $F_\alpha \rightarrow F_\gamma$ results in

$$\begin{pmatrix} \boldsymbol{p}_\alpha \\ 1 \end{pmatrix} = {}^{\alpha}\boldsymbol{T}_{\beta} \cdot {}^{\beta}\boldsymbol{T}_{\gamma} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{A.29}$$

$$= \begin{pmatrix} 0 \\ 0 \\ \pm\left(l_{\text{tib}} \sqrt{1 - \frac{(l_{\text{cox}} + l_{\text{fem}} \cos(\beta))^2}{l_{\text{tib}}^2}} + l_{\text{fem}} \sin(\beta)\right) \\ 1 \end{pmatrix} \quad . \tag{A.30}$$

Thus, the singularity includes all points along the z-axis of F_α, which coincides with the rotation axis of the α-joint. As a consequence, the inverse kinematics cannot be solved uniquely for points along this axis.

B. Tables of Parameters

name	boundaries	PS_1	PS_2	PS_3	PS_4	PS_5	PS_6
influence 1	[0,1]	0.112	0.493	0.504	0.585	0.644	0.763
influence 2 ipsilateral	[1,5]	4.119	3.079	3.968	3.620	4.057	3.460
influence 2 contralateral	[1,5]	3.695	4.850	3.945	4.203	3.554	3.610
influence 3 ipsilateral	[1,5]	3.062	4.614	3.004	3.582	3.630	4.337
influence 3 contralateral	[1,5]	2.857	4.849	3.622	4.063	3.148	2.825
influence 3 time threshold	[0,5]	2.498	4.513	4.948	4.700	0.712	3.925
restricted-neighbor-induced swing	[1,5]	1.543	3.463	4.987	3.256	2.525	2.820
stuck-induced swing	[1,5]	3.994	1.276	3.933	3.843	1.410	3.233
x-position of home front legs [m]	[-0.3,0.3]	0.195	0.042	0.158	0.169	0.120	0.154
x-position of home middle legs [m]	[-0.3,0.3]	-0.072	0.019	0.029	-0.056	0.000	0.031
x-position of home hind legs [m]	[-0.3,0.3]	-0.045	-0.128	-0.111	-0.126	-0.171	-0.150
time threshold front legs [s]	[0,3]	2.039	1.054	2.910	2.332	2.936	1.987
time threshold middle legs [s]	[0,3]	1.095	2.269	2.476	1.016	1.205	2.279
time threshold hind legs [s]	[0,3]	1.142	2.110	2.454	2.932	1.720	2.539

Table B.1.: List of parameters, effective directions for coordination influences, and corresponding limits that were used during the walking optimization.

		front		middle		hind	
		left	right	left	right	left	right
posture 0	x-pos. [m]	0.678	0.578	0.168	0.218	0.050	-0.050
	y-pos. [m]	0.29	-0.29	0.29	-0.29	0.29	-0.29
	z-pos. [m]	-0.2		-0.2		-0.2	
posture 1	x-pos. [m]	0.831	0.734	0.411	0.193	-0.076	0.043
posture 2	x-pos. [m]	0.704	0.737	0.286	0.241	-0.225	0.151
posture 3	x-pos. [m]	0.833	0.545	0.170	0.391	-0.157	0.041
posture 4	x-pos. [m]	0.839	0.641	0.162	0.001	-0.230	-0.197
posture 5	x-pos. [m]	0.632	0.799	0.270	0.148	0.097	-0.098
posture 6	x-pos. [m]	0.545	0.591	0.420	-0.016	0.000	-0.143
posture 7	x-pos. [m]	0.548	0.579	0.360	0.405	0.169	-0.211
posture 8	x-pos. [m]	0.474	0.688	0.382	0.191	0.077	0.023
posture 9	x-pos. [m]	0.535	0.361	0.395	0.124	-0.199	-0.010
posture 10	x-pos. [m]	0.653	0.814	0.373	0.372	-0.068	-0.178
posture 11	x-pos. [m]	0.797	0.417	0.367	0.353	-0.231	0.110
posture 12	x-pos. [m]	0.764	0.725	0.143	0.287	0.037	0.134
posture 13	x-pos. [m]	0.635	0.739	0.284	0.308	0.012	-0.160
postures 0-13	y-pos. [m]	0.216	-0.216	0.27	-0.27	0.27	-0.27
	z-pos. [m]	-0.18		-0.18		-0.18	

Table B.2.: List of foot positions for the starting postures used during the walking optimization. The positions refer to the RCS located between the onsets of the hind legs.

Acronyms

ABS Acrylonitrile Butadiene Styrene. 26

AEP Anterior Extreme Position. 55, 56, 58–62, 192

AFSM Augmented FSM. 5, 8

ANN Artificial Neuronal Network. 5, 9, 14, 53, 61, 73–75, 100, 101, 125, 170

BLDC BrushLess DC. 24, 25, 29

CAD Collision Avoidance Direction. 162

CFRP Carbon-Fiber–Reinforced Polymer. e, 32, 37

CNS Central Nervous System. 13

COM Center Of Mass. 3, 10, 20, 22, 38, 40, 45–51, 62, 63, 65, 66, 68, 84–87, 113–119, 170

COMANV Center-Of-Mass-Average-Nearness-Vector. 162

CPG Central-Pattern Generator. 6, 9, 14

DH Denavit-Hartenberg. 179, 181, 182, 184

DoF Degree of Freedom. 5, 6, 9, 174

EMD Elementary Motion Detector. 162, 167

ESM Energy Stability Margin. 84, 85, 113–115, 117, 126, 175

FEM Finite Element Method. 25–27

FSM Finite State Machine. 8, 191

GPS Global Positioning System. 9

Hector HExapod Cognitive auTonomously Operating Robot. 3, 16–19, 30, g, 34, 40, 43, 45, 49, 53, 56, 63, 64, 68, 71, 72, 79, 97, 103, 113, 115–117, 125, 127, 156, 159, 160, 162, 163, 167, 169–175, 181

iAEP intrinsic AEP. 56, 58, 59

IC Integrated Circuit. 29, 30

iPEP intrinsic PEP. 56, 58, 59, 61

LCS Leg Coordinate System. Coordinate system that is located at the onset of each leg. The axes of the LCS are oriented parallel to the axes of the RCS. 82, 107, 175, 179, 180, 182

LiPo Lithium-ion Polymer. 37

LPF Low Pass Filter. 100

LPS Leg Phase State. 14

MMC Mean of Multiple Computation. 97

MOSFET Metal–Oxide–Semiconductor Field-Effect Transistor. 29, 30

NBR Nitrile Butadiene Rubber. 26, 27

ODE Open Dynamics Engine. 39

PCB Printed Circuit Board. 29

PEP Posterior Extreme Position. 55–59, 61, 62, 74, 75, 87, 117, 118, 155, 192

RCS Robot Coordinate System. The origin of the coordinate system is located between the onsets of the hind legs. Its x-axis is oriented towards the center between the front legs, its y-axis points at the onset of the left hind leg. The z-axis is directed “upwards” to create a right-handed coordinate system. 79, 93, 94, 98, 99, 148–154, 166, 179, 190, 192

SEA Series Elastic Actuator. 17, 26

SVD Singular Value Decomposition. 92, 94, 95

TCP/IP Transmission Control Protocol/Internet Protocol. 39–42

TPE Thermoplastic Elastomer. 26

TPU Thermoplastic PolyUrethane. 26

USB Universal Serial Bus. 37, 39–42

VDA Variable Damping Actuator. 17

VSA Variable Stiffness Actuator. 17

XML eXtensible Markup Language. 39

Bibliography

Académie des sciences (1735). *Machines et inventions approuvées par l'Academie royale des sciences, depuis son établissement jusqu'à present.* Machines et inventions approuvées par l'Academie royale des sciences, depuis son établissement jusqu'à present 6 (cit. on pp. 1 f.).

Adams, D. N. (1985). *So Long, and Thanks for all the Fish.* Vol. 4. The Hitch Hikers' Guide to the Galaxy. Pan Books (cit. on p. b).

Alexandre, P. and A. Preumont (1996). "On the Gait Control of a Six-Legged Walking Machine". In: *International Journal of Systems Science* 27(8), pp. 713–721 (cit. on p. 14).

Allen, T., R. D. Quinn, R. Bachmann, and R. E. Ritzmann (2003). "Abstracted Biological Principles Applied with Reduced Actuation Improve Mobility of Legged Vehicles". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1370–1375 (cit. on p. 11).

Amendt, O. and M. Frik (1994). "Neuronale Steuerung einer sechsbeinigen Gehmaschine". In: *Automatisierungstechnik* (cit. on pp. 5, 9, 46).

Andeen, G. and R. Kombluh (1988). "Design of Compliance in Robots". In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 276–281 (cit. on p. 17).

Annunziata, S., J. Paskarbeit, and A. Schneider (2013). "Bio-Inspired Elbow Impedance Modulation Using a Compliant Technical Joint Drive". In: *Nature-Inspired Mobile Robotics.* Proceedings of the International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, pp. 249–256 (cit. on p. 30).

Arun, K., T. S. Huang, and S. D. Blostein (1987). "Least-Squares Fitting of Two 3-D Point Sets". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-9(5), pp. 698–700 (cit. on p. 95).

Baisch, A. T., C. Heimlich, M. Karpelson, and R. J. Wood (2011). "HAMR3: An Autonomous 1.7g Ambulatory Robot". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5073–5079 (cit. on p. 71).

Bares, J. E. and D. S. Wettergreen (1999). "Dante II: Technical Description, Results, and Lessons Learned". In: *The International Journal of Robotics Research* 18(7), pp. 621–649 (cit. on p. 8).

Bartsch, S., T. Birnschein, F. Cordes, D. Kühn, P. Kampmann, J. Hilljegerdes, S. Planthaber, M. Römmermann, and F. Kirchner (2010). "SpaceClimber: Development of a Six-Legged Climbing Robot for Space Exploration". In: *Proceedings of the 41st International Symposium and 6th German Conference on Robotics.* VDE Verlag, pp. 1–8 (cit. on pp. 6, 9).

Bartsch, S., T. Birnschein, M. Römmermann, J. Hilljegerdes, D. Kühn, and F. Kirchner (2012). "Development of the Six-Legged Walking and Climbing Robot SpaceClimber". In: *Journal of Field Robotics* 29(3), pp. 506–532 (cit. on p. 8).

Bartsch, S. and S. Planthaber (2009). "Scarabaeus: A Walking Robot Applicable to Sample Return Missions". In: *Research and Education in Robotics - EUROBOT 2008.* Vol. 33. Communications in Computer and Information Science. Springer, pp. 128–133 (cit. on p. 6).

Bässler, U. (1983). *Neural Basis of Elementary Behavior in Stick Insects.* Vol. 10. Studies of Brain Function ; 10. Springer, 169 S. (Cit. on pp. 20 ff., 113).

Batz+Burgel GmbH & Co. KG (2016). *Werkstoffdatenblatt: EN AW - 7075 (AlZnMgCu1,5).* URL: `http://www.batz-burgel.de/downloads/aluminium/low-res/BB_A12_150dpi.pdf` (visited on 04/12/2016) (cit. on p. 26).

Bauchhenß, E. (1971). Carausius morosus *Br.* Vol. 14c. Großes zoologisches Praktikum ; 14c. Fischer, 53 S. : Ill. (Cit. on p. 20).

Beer, R. D., H. J. Chiel, R. D. Quinn, K. S. Espenschied, and P. Larsson (1992). "A Distributed Neural Network Architecture for Hexapod Robot Locomotion." In: *Neural Computation* 4(3), pp. 356–365 (cit. on pp. 5, 13 ff., 62).

Bender, J. A., E. M. Simpson, B. R. Tietz, K. A. Daltorio, R. D. Quinn, and R. E. Ritzmann (2011). "Kinematic and Behavioral Evidence for a Distinction Between Trotting and Ambling Gaits in the Cockroach *Blaberus discoidalis*". In: *Journal of Experimental Biology* 214(12), pp. 2057–2064 (cit. on p. 72).

Berns, K., S. Cordes, and W. Ilg (1994). "Adaptive, Neural Control Architecture for the Walking Machine LAURON". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.* Vol. 2, pp. 1172–1177 (cit. on pp. 5, 9, 11, 46).

Bert, P. (1866). “Locomotion chez plusieurs espéces animales”. In: *Mémoires de la Société des Sciences Physiques et Naturelles de Bordeaux* 4(1), pp. 59–72 (cit. on p. 12).

Bertrand, O., J. P. Lindemann, and M. Egelhaaf (2015). “A Bio-Inspired Collision Avoidance Model Based on Spatial Information Derived from Motion Detectors Leads to Common Routes”. In: *PLOS Computational Biology* 11(11), pp. 1–28 (cit. on pp. 160, 162).

Bowerman, R. F. and T. Root (1978). “External Anatomy and Muscle Morphology of the Walking Legs of the Scorpion *Hadrurus arizonensis*”. In: *Comparative Biochemistry and Physiology Part A: Physiology* 59(1), pp. 57–63 (cit. on p. 9).

Bretl, T. and S. Lall (2008). “Testing Static Equilibrium for Legged Robots.” In: *IEEE Transactions on Robotics* 24(4), pp. 794–807 (cit. on p. 10).

Brooks, R. A. (1989). “A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network”. In: *Neural Computation* 1(2), pp. 253–262 (cit. on pp. 5, 8).

— (1992). “Artificial Life and Real Robots”. In: *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*. Complex adaptive systems. MIT Press, pp. 3–10 (cit. on p. 71).

Buchner, E. (1976). “Elementary Movement Detectors in an Insect Visual System”. In: *Biological Cybernetics* 24(2), pp. 85–101 (cit. on p. 162).

Bullock, J. M. R., P. Drechsler, and W. Federle (2008). “Comparison of Smooth and Hairy Attachment Pads in Insects: Friction, Adhesion and Mechanisms for Direction-Dependence”. In: *Journal of Experimental Biology* 211(20), pp. 3333–3343 (cit. on pp. 21, 45, 113).

Burkamp, T. (1996). “Kontrolle der Phasenübergänge eines sechsbeinigen Laufsystems. Biologische Experimente und Computersimulation.” MA thesis. Bielefeld University (cit. on pp. 61, 75).

Burns, M. D. (1973). “The Control of Walking in *Orthoptera*: I. Leg Movements in Normal Walking”. In: *Journal of Experimental Biology* 58(1), pp. 45–58 (cit. on p. 12).

Bußhardt, P., S. N. Gorb, and H. Wolf (2011). “Activity of the claw retractor muscle in stick insects in wall and ceiling situations”. In: *Journal of Experimental Biology* 214(10), pp. 1676–1684 (cit. on p. 45).

Byl, K., A. Shkolnik, S. Prentice, N. Roy, and R. Tedrake (2009). “Reliable Dynamic Motions for a Stiff Quadruped”. In: *Experimental Robotics: The Eleventh International Symposium*. Springer, pp. 319–328 (cit. on p. 16).

Čapek, K. (2004). *R.U.R. (Rossum’s Universal Robots)*. Trans. by C. Novack. With an intro. by I. Klíma. (originally published in 1921 by Aventinum). Penguin Publishing Group (cit. on p. 1).

Cham, J. G., S. A. Bailey, J. E. Clark, R. J. Full, and M. R. Cutkosky (2002). “Fast and Robust: Hexapedal Robots via Shape Deposition Manufacturing”. In: *The International Journal of Robotics Research* 21(10-11), pp. 869–882 (cit. on p. 10).

Chapman, K. and J. Pankhurst (1967). “Conduction Velocities and Their Temperature Coefficients in Sensory Nerve Fibres of Cockroach Legs”. In: *Journal of Experimental Biology* 46(1), pp. 63–84 (cit. on p. 60).

Chen, C.-H. and V. Kumar (1996). “Motion Planning of Walking Robots in Environments With Uncertainty”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 4, pp. 3277–3282 (cit. on p. 72).

Collins, J. W., L. M. Sanderson, and J. McGlothlin (1985). “Death by Robot: Safety Issues in Automated Plants”. In: *Business and Society Review: A Quarterly Form on the Role of Business in a Free Society* (54), pp. 56–59 (cit. on p. 16).

Cordes, S. and P. Bührle (1996). “LAURON II - Simulation und Realisierung einer sechsbeinigen Laufmaschine”. In: *Autonome Mobile Systeme*. Informatik aktuell. Springer, pp. 190–203 (cit. on pp. 5, 9).

Craig, J. J. (2004). *Introduction to Robotics: Mechanics and Control*. 3rd ed. Prentice Hall (cit. on p. 179).

Cruse, H. (1976a). “The Control of Body Position in the Stick Insect (*Carausius morosus*), When Walking Over Uneven Surfaces”. In: *Biological Cybernetics* 24(1), pp. 25–33 (cit. on pp. 9, 20, 46).

— (1976b). “The Function of the Legs in the Free Walking Stick Insect, *Carausius morosus*”. In: *Journal of Comparative Physiology. B, Biochemical, Systemic, and Environmental Physiology* 112(2), pp. 235–262 (cit. on pp. 9, 20 f., 32 f., 60, 181).

— (1979a). “A New Model Describing the Coordination Pattern of the Legs of a Walking Stick Insect”. In: *Biological Cybernetics* 32(2), pp. 107–113 (cit. on pp. 9, 159).

— (1979b). “The Control of the Anterior Extreme Position of the Hindleg of a Walking Insect, *Carausius morosus*”. In: *Physiological Entomology* 4(2), pp. 121–124 (cit. on pp. 57, 60).

— (1980a). “A Quantitative Model of Walking Incorporating Central and Peripheral Influences I. The Control of the Individual Leg”. In: *Biological Cybernetics* 37(3), pp. 131–136 (cit. on p. 9).

— (1980b). “A Quantitative Model of Walking Incorporating Central and Peripheral Influences II. The Connections Between the Different Legs”. In: *Biological Cybernetics* 37(3), pp. 137–144 (cit. on pp. 9, 11, 13).

— (1985). “The Influence of Load, Position and Velocity on the Control of Leg Movement of a Walking Insect”. In: *Insect Locomotion.* Parey, pp. 19–26 (cit. on p. 45).

— (1990). “What Mechanisms Coordinate Leg Movement in Walking Arthropods?” In: *Arthropod Locomotion Systems: from Biological Materials and Systems to Robotics* 33, pp. 237–250 (cit. on pp. 9, 13 f., 53, 56, 170).

Cruse, H., C. Bartling, G. S. Cymbalyuk, J. Dean, and M. Dreifert (1994). “A Neural Net Controller for a Six-Legged Walking System”. In: *Proceedings of the From Perception to Action Conference*, pp. 55–65 (cit. on p. 9).

Cruse, H., C. Bartling, M. Dreifert, J. Schmitz, D. Brunn, J. Dean, and T. Kindermann (1995). “Walking: A Complex Behavior Controlled by Simple Networks”. In: *Adaptive Behavior* 3(4), pp. 385–418 (cit. on p. 100).

Cruse, H., C. Bartling, and T. Kindermann (1995). “High-Pass Filtered Positive Feedback for Decentralized Control of Cooperation”. In: *Advances in Artificial Life: Third European Conference on Artificial Life.* Springer, pp. 668–678 (cit. on p. 97).

Cruse, H., B. Bläsing, J. Dean, V. Dürr, T. Kindermann, J. Schmitz, and M. Schumm (2004). “WalkNet — a Decentralized Architecture for the Control of Walking Behaviour Based on Insect Studies”. In: *Walking: Biological and Technological Aspects.* Vol. 467. International Centre for Mechanical Sciences. Springer, pp. 81–118 (cit. on pp. 54, 60, 73, 100).

Cruse, H., J. Dean, U. Müller, and J. Schmitz (1991). “The Stick Insect as a Walking Robot”. In: *Proceedings of the 5th International Conference on Advanced Robotics.* Vol. 2, pp. 936–940 (cit. on pp. 46, 60).

Cruse, H., V. Dürr, and J. Schmitz (2007). “Insect Walking Is Based on a Decentralized Architecture Revealing a Simple and Robust Controller”. In: *Philosophical*

Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences 365(1850), pp. 221–250 (cit. on p. 11).

Cruse, H., I. Ehmanns, S. Stübner, and J. Schmitz (2009). "Tight Turns in Stick Insects". In: *Journal of Comparative Physiology A* 195(3), pp. 299–309 (cit. on p. 45).

Cruse, H. and S. Epstein (1982). "Peripheral Influences on the Movement of the Legs in a Walking Insect *Carausius morosus*". In: *Journal of Experimental Biology* 101(1), pp. 161–170 (cit. on p. 57).

Cruse, H., T. Kindermann, M. Schumm, J. Dean, and J. Schmitz (1998). "Walknet–A Biologically Inspired Network to Control Six-Legged Walking." In: *Neural Networks* 11(7-8), pp. 1435–1447 (cit. on pp. e, 18, 53 f., 62, 71).

Cruse, H. and A. Knauth (1989). "Coupling Mechanisms Between the Contralateral Legs of a Walking Insect (*Carausius morosus*)". In: *Journal of Experimental Biology* 144(1), pp. 199–213 (cit. on p. 58).

Cruse, H. and G. Saxler (1980). "The Coordination of Force Oscillations and of Leg Movement in a Walking Insect (*Carausius morosus*)". In: *Biological Cybernetics* 36(3), pp. 165–171 (cit. on p. 54).

Cruse, H. and W. Schwarze (1988). "Mechanisms of Coupling Between the Ipsilateral Legs of a Walking Insect (*Carausius morosus*)". In: *Journal of Experimental Biology* 138(1), pp. 455–469 (cit. on p. 58).

Cruse, H. and R. Wehner (2011). "No Need for a Cognitive Map: Decentralized Memory for Insect Navigation". In: *PLOS Computational Biology* 7(3), pp. 1–10 (cit. on p. 54).

Daerden, F. and D. Lefeber (2002). "Pneumatic Artificial Muscles: Actuators for Robotics and Automation". In: *European Journal of Mechanical and Environmental Engineering* 47(1), pp. 11–21 (cit. on p. 17).

Dallali, H., P. Kormushev, Z. Li, and D. G. Caldwell (2012). "On Global Optimization of Walking Gaits for the Compliant Humanoid Robot COMAN Using Reinforcement Learning". In: *International Journal of Cybernetics and Information Technologies* 12(3), pp. 76–85 (cit. on p. 16).

Darwin, C. and A. Wallace (1858). "On the Tendency of Species to form Varieties; and on the Perpetuation of Varieties and Species by Natural Means of Selection." In: *Journal of the Proceedings of the Linnean Society of London. Zoology* 3(9), pp. 45–62 (cit. on p. 2).

De Schutter, J., H. Bruyninckx, W.-H. Zhu, and M. W. Spong (1998). “Force Control: A Bird’s Eye View”. In: *Control Problems in Robotics and Automation*. Springer, pp. 1–17 (cit. on p. 16).

de Sinéty, R. (1901). “Recherches sur la biologie et l’anatomie des Phasmes”. In: *La Cellule* 19(1), pp. 117–278 (cit. on p. 19).

Dean, J. (1989). “Leg Coordination in the Stick Insect *Carausius morosus*: Effects of Cutting Thoracic Connectives”. In: *Journal of Experimental Biology* 145(1), pp. 103–131 (cit. on p. 14).

— (1991). “A Model of Leg Coordination in the Stick Insect, *Carausius morosus* I. A Geometrical Consideration of Contralateral and Ipsilateral Coordination Mechanisms Between Two Adjacent Legs”. In: *Biological Cybernetics* 64(5), pp. 393–402 (cit. on p. 62).

Dean, J. and G. Wendler (1984). “Stick Insect Locomotion on a Wheel: Patterns Of Stopping And Starting”. In: *Journal of Experimental Biology* 110(1), pp. 203–216 (cit. on pp. 10, 45).

Delcomyn, F. (1971). “The Locomotion of the Cockroach *Periplaneta americana*”. In: *Journal of Experimental Biology* 54(2), pp. 443–452 (cit. on p. 12).

— (1999). “Walking Robots and the Central and Peripheral Control of Locomotion in Insects”. In: *Autonomous Robots* 7 (3), pp. 259–270 (cit. on p. 14).

Denavit, J. and R. Hartenberg (1955). “A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices.” In: *Transactions of the ASME Journal of Applied Mechanics* 22, pp. 215–221 (cit. on pp. 179 f.).

Devol, J. (1961). “Programmed Article Transfer”. US Patent US2988237 A. US Patent 2,988,237 (cit. on p. 1).

DFKI GmbH & University of Bremen (2016). *SpaceClimber*. URL: http://robotik.dfki-bremen.de/uploads/tx_dfkiprojects/Systemblatt_SpaceClimber_en.pdf (visited on 04/12/2016) (cit. on p. 8).

Dürr, V. (2005). “Context-Dependent Changes in Strength and Efficacy of Leg Coordination Mechanisms”. In: *Journal of Experimental Biology* 208(12), pp. 2253–2267 (cit. on pp. 54, 137, 146, 174).

Dürr, V. and W. Ebeling (2005). “The Behavioural Transition From Straight to Curve Walking: Kinetics of Leg Movement Parameters and the Initiation of Turning”. In: *Journal of Experimental Biology* 208(12), pp. 2237–2252 (cit. on p. 75).

Dürr, V., J. Schmitz, and H. Cruse (2004). “Behaviour-Based Modelling of Hexapod Locomotion: Linking Biology and Technical Application”. In: *Arthropod Structure & Development* 33(3), pp. 237–250 (cit. on pp. 53, 55).

Erden, M. S. (2011). “Optimal Protraction of a Biologically Inspired Robot Leg”. In: *Journal of Intelligent & Robotic Systems* 64(3), pp. 301–322 (cit. on p. 103).

Espenschied, K. S., H. J. Chiel, R. D. Quinn, and R. D. Beer (1993). “Leg Coordination Mechanisms in the Stick Insect Applied to Hexapod Robot Locomotion”. In: *Adaptive Behaviour* 1(4), pp. 455–468 (cit. on pp. 5, 62).

Espenschied, K. S. and R. D. Quinn (1994). “Biologically-Inspired Hexapod Robot Design and Simulation”. In: *Proceedings of the Conference on Intelligent Robotics in Field, Factory, Service, and Space (CIRFFSS).* Vol. 1, pp. 21–28 (cit. on pp. 61, 88, 101).

Espenschied, K. S., R. D. Quinn, R. D. Beer, and H. J. Chiel (1996). “Biologically Based Distributed Control and Local Reflexes Improve Rough Terrain Locomotion in a Hexapod Robot.” In: *Robotics and Autonomous Systems* 18(1-2), pp. 59–64 (cit. on pp. 5, 46).

Espenschied, K. S., R. D. Quinn, H. J. Chiel, and R. D. Beer (1995). “Biologically-Inspired Hexapod Robot Project: Second Robot”. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation.* Vol. 3, p. V11 (cit. on pp. 62, 172).

Evoy, W. H. and C. R. Fourtner (1973). “Crustacean Walking”. In: *Control of Posture and Locomotion.* Springer, pp. 477–493 (cit. on p. 152).

Ferrell, C. (1995). “A Comparison of Three Insect-Inspired Locomotion Controllers”. In: *Robotics and Autonomous Systems* 16(2-4). Moving the Frontiers Between Robotics and Biology, pp. 135–159 (cit. on pp. 5, 8 f., 62).

Fielding, M. R. (2002). “Omnidirectional Gait Generating Algorithm for Hexapod Robot”. PhD thesis (cit. on pp. 62, 76, 101, 103).

Fielding, M. R. and G. Dunlop (2001). “Exponential Fields to Establish Inter-Leg Influences for Omnidirectional Hexapod Gait”. In: *Proceedings of the 4th International Conference on Climbing and Walking Robots*, pp. 587–594 (cit. on p. 71).

— (2004). “Omnidirectional Hexapod Walking and Efficient Gaits Using Restrictedness.” In: *International Journal Of Robotics Research* 23(10-11), pp. 1105–1110 (cit. on pp. 15, 87, 171).

Flannigan, W., G. Nelson, and R. D. Quinn (1998). “Locomotion Controller for a Crab-Like Robot”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Vol. 1, pp. 152–156 (cit. on p. 62).

Frank, A. A. (1968). *Automatic Control Systems for Legged Locomotion Machines.* USCEE Report 273. University of Southern California Electronic Sciences Laboratory (cit. on p. 7).

Frik, M., M. Guddat, M. Karatas, and D. Losch (1999). “A Novel Approach to Autonomous Control of Walking Machines”. In: *Proceedings of the 2nd International Conference on Climbing and Walking Robots*, pp. 333–342 (cit. on pp. 5, 46).

Full, R. J. and M. S. Tu (1991). “Mechanics of a Rapid Running Insect: Two-, Four- and Six-Legged Locomotion”. In: *Journal of Experimental Biology* 156(1), pp. 215–231 (cit. on p. 10).

Garber, M. (2012). *A Brief History of Mechanical Horses.* URL: http://www.theatlantic.com/technology/archive/2012/09/a-brief-history-of-mechanical-horses/262942/ (cit. on p. 4).

Garcia, E., J. Estremera, and P. G. de Santos (2002). “A Comparative Study of Stability Margins for Walking Machines.” In: *Robotica* 20(6), pp. 595–606 (cit. on p. 10).

Garcia, M., A. Kuo, A. Peattie, P. Wang, and R. J. Full (2000). “Damping and Size: Insights and Biological Inspiration”. In: *Proceedings of the International Symposium on Adaptive Motion of Animals and Machines* (cit. on p. 22).

Gaßmann, B., K.-U. Scholl, and K. Berns (2001). “Locomotion of LAURON III in Rough Terrain”. In: *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics.* Vol. 2, pp. 959–964 (cit. on pp. 5, 9, 46).

Goldschmidt, D., F. Hesse, F. Wörgötter, and P. Manoonpong (2012). “Biologically Inspired Reactive Climbing Behavior of Hexapod Robots”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4632–4637 (cit. on p. 71).

Goodwin, W. F. (1867). “Automatic Toy”. US Patent 61,416 (cit. on p. 4).

Görner, M. and G. Hirzinger (2010). “Analysis and Evaluation of the Stability of a Biologically Inspired, Leg Loss Tolerant Gait for Six- and Eight-Legged Walking Robots.” In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4728–4735 (cit. on pp. 62, 88).

Görner, M., T. Wimböck, A. Baumann, M. Fuchs, T. Bahls, M. Grebenstein, C. Borst, J. Butterfaß, and G. Hirzinger (2008). “The DLR-Crawler: A Testbed for Actively

Compliant Hexapod Walking Based on the Fingers of DLR-Hand II". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1525–1531 (cit. on pp. 6, 16).

Görner, M., T. Wimböck, and G. Hirzinger (2009). "The DLR Crawler: Evaluation of Gaits and Control of an Actively Compliant Six-Legged Walking Robot". In: *Industrial Robot* 36(4), pp. 344–351 (cit. on pp. 11, 61, 172).

Graham, D. (1972). "A Behavioural Analysis of the Temporal Organisation of Walking Movements in the 1st Instar and Adult Stick Insect (*Carausius morosus*)". In: *Journal of Comparative Physiology* 81(1), pp. 23–52 (cit. on pp. 12, 54, 57).

— (1977). "Simulation of a Model for the Coordination of Leg Movement in Free Walking Insects". In: *Biological Cybernetics* 26(4), pp. 187–198 (cit. on pp. 12 f.).

— (1979). "Effects of Circum-Oesophageal Lesion on the Behaviour of the Stick Insect *Carausius morosus*". In: *Biological Cybernetics* 32(3), pp. 139–145 (cit. on p. 13).

Graham, D. and S. Epstein (1985). "Behaviour and Motor Output for an Insect Walking on a Slippery Surface: II. Backward Walking". In: *Journal of Experimental Biology* 118(1), pp. 287–296 (cit. on p. 136).

Haldane, D., K. Peterson, F. Garcia Bermudez, and R. Fearing (2013). "Animal-Inspired Design and Aerodynamic Stabilization of a Hexapedal Millirobot". In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3279–3286 (cit. on p. 21).

Haynes, G. and A. Rizzi (2006). "Gaits and Gait Transitions for Legged Robots". In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1117–1122 (cit. on p. 11).

Hennemann, F. H. and O. V. Conle (2008). "Revision of Oriental Phasmatodea: the Tribe Pharnaciini Gunther, 1953, Including the Description of the World's Longest Insect, and a Survey of the Family Phasmatidae Gray, 1835 With Keys to the Subfamilies and Tribes (Phasmatodea: "Anareolatae": Phasmatidae)". In: *Zootaxa* (1906), pp. 1–311 (cit. on p. 21).

Hoover, A., E. Steltz, and R. Fearing (2008). "RoACH: An Autonomous 2.4g Crawling Hexapod Robot". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 26–33 (cit. on p. 21).

Hou, J., M. Luo, and T. Mei (2010). "The Design and Control of Amoeba-Like Robot". In: *Proceedings of the International Conference on Computer Application and System Modeling*. Vol. 1, pp. V1-88–V1-91 (cit. on p. 2).

Hughes, G. M. (1952). “The Co-Ordination of Insect Movements: I. The Walking Movements of Insects”. In: *Journal of Experimental Biology* 29(2), pp. 267–285 (cit. on pp. 10 f., 45).

— (1965). “Locomotion: Terrestrial”. In: *The Physiology of Insecta*. Vol. 2. Academic Press Inc. Chap. 4, pp. 227–254 (cit. on p. 12).

Hughes, G. M. and P. Mill (1974). “Locomotion: Terrestrial”. In: *The Physiology of Insecta (Second Edition)*. Vol. 3. Academic Press Inc. Chap. 5, pp. 335–379 (cit. on p. 12).

Hutter, M., C. Gehring, M. Blösch, M. A. Hoepflinger, C. D. Remy, and R. Siegwart (2012). “Starleth: A Compliant Quadrupedal Robot for Fast, Efficient, and Versatile Locomotion”. In: *Adaptive Mobile Robotics*. World Scientific. Chap. 62, pp. 483–490 (cit. on p. 17).

Irwansyah, A., O. W. Ibraheem, D. Klimeck, M. Porrmann, and U. Rückert (2014). “FPGA-Based Generic Architecture for Rapid Prototyping of Video Hardware Accelerators using NoC AXI4-Stream Interconnect and GigE Vision Camera Interfaces”. In: *Bildverarbeitung in der Automation (BVAu) 2014* (cit. on pp. 160, 167).

Jakobi, N., P. Husbands, and I. R. Harvey (1995). “Noise and the Reality Gap: The Use of Simulation in Evolutionary Robotics”. In: *Proceedings of the Third European Conference on Advances in Artificial Life*. Springer, pp. 704–720 (cit. on p. 71).

Jander, J. (1985). “Mechanical Stability in Stick Insects When Walking Straight and Around Curves.” In: *Insect Locomotion*. Paul Parey, pp. 33–42 (cit. on pp. 46, 49).

Katzschmann, R. K., A. D. Marchese, and D. Rus (2016). “Hydraulic Autonomous Soft Robotic Fish for 3D Swimming”. In: *Experimental Robotics: The 14th International Symposium on Experimental Robotics*. Springer, pp. 405–420 (cit. on p. 2).

Kerscher, T., A. Rönnau, M. Ziegenmeyer, B. Gaßmann, J. M. Zöllner, and R. Dillmann (2008). “Behaviour-Based Control of a Six-Legged Walking Machine LAURON IVc”. In: *Proceedings of the 11th International Conference on Climbing and Walking Robots* (cit. on p. 72).

Khatib, O. (1986). “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”. In: *The International Journal of Robotics Research* 5(1), pp. 90–98 (cit. on p. 100).

Kindermann, T. (2003). “Positive Rückkopplung zur Kontrolle komplexer Kinematiken am Beispiel des hexapoden Laufens: Experimente und Simulationen”. PhD thesis. Universität Bielefeld (cit. on pp. 46, 49, 52, 170).

Kingsley, D., R. D. Quinn, and R. E. Ritzmann (2006). "A Cockroach Inspired Robot With Artificial Muscles". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1837–1842 (cit. on p. 6).

Klein, C. A. and R. L. Briggs (1980). "Use of Active Compliance in the Control of Legged Vehicles". In: *IEEE Transactions on Systems, Man, and Cybernetics* 10(7), pp. 393–400 (cit. on p. 11).

Koh, J.-S., E. Yang, G.-P. Jung, S.-P. Jung, J. H. Son, S.-I. Lee, P. G. Jablonski, R. J. Wood, H.-Y. Kim, and K.-J. Cho (2015). "Jumping on Water: Surface Tension–dominated Jumping of Water Striders and Robotic Insects". In: *Science* 349(6247), pp. 517–521 (cit. on p. 71).

Kramer, E. (1976). "The Orientation of Walking Honeybees in Odour Fields with Small Concentration Gradients". In: *Physiological Entomology* 1 (1) (cit. on p. 46).

La Greca, M. (1943). "Su una particolare maniera di deambulazione di un Acridide: Tropidopola cylindrica (Marsch.)" In: *Bolletino di zoologia* 14(4-6), pp. 83–104 (cit. on pp. 12, 45).

Lévy, J. and H. Cruse (2008). "Controlling a System With Redundant Degrees of Freedom: II. Solution of the Force Distribution Problem Without a Body Model". In: *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology* 194 (8), pp. 735–750 (cit. on p. 97).

Liao, L., F. (Xi, and K. Liu (2008). "Modeling and Control of Automated Polishing/Deburring Process Using a Dual-Purpose Compliant Toolhead". In: *International Journal of Machine Tools and Manufacture* 48(12–13), pp. 1454–1463 (cit. on p. 16).

Ma, K. Y., P. Chirarattananon, S. B. Fuller, and R. J. Wood (2013). "Controlled Flight of a Biologically Inspired, Insect-Scale Robot". In: *Science* 340(6132), pp. 603–607 (cit. on pp. 2, 71).

MacBride, E. W. and A. Jackson (1915). "The Inheritance of Colour in the Stick-Insect, *Carausius morosus*". In: *Proceedings of the Royal Society of London B: Biological Sciences* 89(611), pp. 109–118 (cit. on p. 19).

Manoonpong, P., F. Pasemann, and F. Wörgötter (2008). "Sensor-Driven Neural Control for Omnidirectional Locomotion and Versatile Reactive Behaviors of Walking Machines". In: *Robotics and Autonomous Systems*, pp. 265–288 (cit. on pp. 6, 14, 46).

Mason, M. T. (1981). “Compliance and Force Control for Computer Controlled Manipulators”. In: *Systems, Man and Cybernetics, IEEE Transactions on* 11(6), pp. 418–432 (cit. on p. 16).

McGhee, R. B. (1967). “Finite State Control of Quadruped Locomotion”. In: *Simulation* 9(3), pp. 135–140 (cit. on pp. 5, 7).

McGhee, R. B. and A. A. Frank (1968). “On the Stability Properties of Quadruped Creeping Gaits”. In: *Mathematical Biosciences* 3, pp. 331–351 (cit. on pp. 46, 113).

McGhee, R. B. and S.-S. Sun (1974). “On the Problem of Selecting a Gait for a Legged Vehicle”. In: *Proceedings of the VI. IFAC Symposium Control in Space*. Vol. 4. 4, pp. 53–62 (cit. on pp. 10, 45).

Meissner, O. (1909). “Biologische Beobachtungen an der indischen Stabheuschrecke, *Dixippus morosus* Br. (Phasiu.; Orth.).” In: *Zeitschrift für wissenschaftliche Insektenbiologie* (5), pp. 14–21, 55–61, 87–95 (cit. on p. 19).

Messuri, D. (1985). “Optimization of the Locomotion of a Legged Vehicle With Respect to Maneuverability”. PhD thesis. Ohio State Univ., Columbus (cit. on pp. 84, 113, 175).

Meyer, H. G., O. Bertrand, J. Paskarbeit, J. P. Lindemann, A. Schneider, and M. Egelhaaf (2016). “A Bio-Inspired Model for Visual Collision Avoidance on a Hexapod Walking Robot”. In: *Biomimetic and Biohybrid Systems: Proceedings of the 5th International Conference Living Machines*. Springer, pp. 167–178 (cit. on pp. 123, 157, 160, 162, 167).

Mittelstaedt, H. (1950). “Physiologie des Gleichgewichtssinnes bei fliegenden Libellen”. In: *Zeitschrift für vergleichende Physiologie* 32(5), pp. 422–463 (cit. on p. 21).

Moll, K., F. Roces, and W. Federle (2010). “Foraging Grass-Cutting Ants (*Atta vollenweideri*) Maintain Stability by Balancing Their Loads With Controlled Head Movements”. In: *Journal of Comparative Physiology A* 196(7), pp. 471–480 (cit. on p. 45).

Möller, R. (2000). “Insect Visual Homing Strategies in a Robot with Analog Processing”. In: *Biological Cybernetics* 83 (3) (cit. on p. 2).

Morrison, R. (1968). “Iron Mule Train”. In: *Proceedings of the Off-Road Mobility Research Symposium*, pp. 359–379 (cit. on pp. 4 f., 7, 11).

Mosher, R. S. (1968). “Test and Evaluation of a Versatile Walking Truck”. In: *Proceedings of the Off-Road Mobility Research Symposium*, pp. 359–379 (cit. on pp. 5, 7).

Murphy, M. P., C. Kute, Y. Menguc, and M. Sitti (2011). “Waalbot II: Adhesion Recovery and Improved Performance of a Climbing Robot using Fibrillar Adhesives”. In: *International Journal of Robotics Research* 30 (1) (cit. on p. 113).

Nagy, P. V. (1992). “An Investigation of Walker/Terrain Interaction”. UMI Order No. GAX92-09771. PhD thesis (cit. on p. 116).

Nelson, G. and R. D. Quinn (1999). “Posture Control of a Cockroach-Like Robot”. In: *Control Systems, IEEE* 19(2), pp. 9–14 (cit. on p. 5).

NYT (1893). “A MECHANICAL MAN. Some New Features of an Invention Many Will Remember.” In: *New York Times* (cit. on pp. 2, 4 f.).

Parness, A., M. Frost, N. Thatte, J. P. King, K. Witkoe, M. Nevarez, M. Garrett, H. Aghazarian, and B. Kennedy (2013). “Gravity-Independent Rock-Climbing Robot and a Sample Acquisition Tool with Microspine Grippers”. In: *Journal of Field Robotics* 30 (6), pp. 897–915 (cit. on p. 113).

Paskarbeit, J., S. Annunziata, D. Basa, and A. Schneider (2013). “A Self-Contained, Elastic Joint Drive for Robotics Applications Based on a Sensorized Elastomer Coupling—Design and Identification”. In: *Sensors And Actuators A-Physical* 199, pp. 56–66 (cit. on pp. 19, 27).

Paskarbeit, J., S. Annunziata, and A. Schneider (2013). “A Resilient Robotic Actuator Based on an Integrated Sensorized Elastomer Coupling”. In: *Nature-Inspired Mobile Robotics.* Proceedings of the 16th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, pp. 257–264 (cit. on p. 19).

Paskarbeit, J., S. Beyer, A. Gucze, J. Schröder, M. Wiltzok, M. Fingberg, and A. Schneider (2016). “OUROBOT — A Self-Propelled Continuous-Track-Robot for Rugged Terrain”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4708–4713 (cit. on p. 2).

Paskarbeit, J., M. Otto, M. Schilling, and A. Schneider (2016). “Stick(y) Insects — Evaluation of Static Stability for Bio-Inspired Leg Coordination in Robotics”. In: *Biomimetic and Biohybrid Systems: Proceedings of the 5th International Conference, Living Machines.* Springer, pp. 239–250 (cit. on pp. 45, 53).

Paskarbeit, J., M. Schilling, J. Schmitz, and A. Schneider (2015). “Obstacle Crossing of a Real, Compliant Robot Based on Local Evasion Movements and Averaging of Stance Heights Using Singular Value Decomposition”. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3140–3145 (cit. on pp. 71, 157).

Paskarbeit, J., J. Schmitz, M. Schilling, and A. Schneider (2010a). “Design of an Insectoid Robot as a Versatile Carrier for Bioinspired Sensors”. In: *Proceedings of the 13th International Conference on Climbing and Walking Robots.* World Scientific, pp. 173–180 (cit. on p. 19).

— (2010b). “Layout and Construction of a Hexapod Robot with Increased Mobility”. In: *Proceedings of the 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pp. 621–625 (cit. on p. 19).

Paul, R. (1981). *Robot Manipulators: Mathematics, Programming, and Control : The Computer Control of Robot Manipulators.* Artificial Intelligence Series. MIT Press (cit. on p. 179).

Pearson, K. G. (1976). “The Control of Walking”. In: *Scientific American* 235(6), pp. 72–86 (cit. on p. 9).

Pearson, K. G., R. Stein, and S. Malhotra (1970). “Properties of Action Potentials from Insect Motor Nerve Fibres”. In: *Journal of Experimental Biology* 53(2), pp. 299–316 (cit. on p. 60).

Pfeiffer, F. (2007). “The TUM Walking Machines”. In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 365(1850), pp. 109–131 (cit. on pp. 6, 9, 46, 62).

Pijnacker, L. P. (1966). “Effect of Centrifugation of the Eggs on the Sex of *Carausius morosus* Br.” In: *Nature* 210(5041), pp. 1184–1185 (cit. on p. 20).

Plooij, M., G. Mathijssen, P. Cherelle, D. Lefeber, and B. Vanderborght (2015). “Lock Your Robot: A Review of Locking Devices in Robotics”. In: *Robotics Automation Magazine, IEEE* 22(1), pp. 106–117 (cit. on p. 17).

Pratt, G. A. and M. M. Williamson (1995). “Series Elastic Actuators.” In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 399–406 (cit. on pp. 16 f., 26).

Pratt, J. E. and G. A. Pratt (1998). “Exploiting Natural Dynamics in the Control of a Planar Bipedal Walking Robot”. In: *Proceedings of the 36th Annual Allerton Conference on Communication, Control and Computing*, pp. 739–748 (cit. on p. 17).

Raibert, M. (1986). *Legged Robots That Balance.* Massachusetts Institute of Technology (cit. on pp. 3 f., 7, 10).

Raibert, M., M. Chepponis, and H. B. Brown (1984). “Experiments in Balance With a 3D One-Legged Hopping Machine”. In: *International Journal of Robotics Research* 3, pp. 75–92 (cit. on p. 7).

Riskin, J. (2003). “The Defecating Duck, or, the Ambiguous Origins of Artificial Life”. In: *Critical Inquiry* 29(4), pp. 599–633 (cit. on p. 1).

Rollinson, D. S. (2014). “Control and Design of Snake Robots”. PhD thesis. Carnegie Mellon University (cit. on p. 2).

Rönnau, A., G. Heppner, M. Nowicki, and R. Dillmann (2014). “LAURON V: A Versatile Six-Legged Walking Robot With Advanced Maneuverability”. In: *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 82–87 (cit. on pp. 6, 8 f., 46).

Rönnau, A., G. Heppner, L. Pfotzer, and R. Dillmann (2013). “Lauron V: Optimized Leg Configuration for the Design of a Bio-Inspired Walking Robot”. In: *Proceedings of the 16th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines* (cit. on p. 9).

Rönnau, A., T. Kerscher, M. Ziegenmeyer, J. M. Zöllner, and R. Dillmann (2009). “Adaptation of a Six-Legged Walking Robot to its Local Environment”. In: *Robot Motion and Control 2009*. Vol. 396. Lecture Notes in Control and Information Sciences. Springer, pp. 155–164 (cit. on pp. 6, 46).

Rygg, L. (1893). “Mechanical Horse”. US Patent 491,927 (cit. on p. 4).

Salmi, S. and A. J. Halme (1996). “Implementing and Testing a Reasoning-Based Free Gait Algorithm in the Six-Legged Walking Machine, "MECANT"”. In: *Control Engineering Practice* 4(4), pp. 487–492 (cit. on p. 14).

Schäffersmann, M. (2011). “Development of the BioFlex Bus, a Data Bus for BioFlex Applications”. MA thesis. Bielefeld University (cit. on pp. 29, 37, 41).

Schäffersmann, M., A. Schneider, and J. Schmitz (2014). “A Self-Adjustable Transducer for Bio-Inspired Strain Detection in Walking Legs”. In: *Proceedings of the 17th International Conference on Climbing and Walking Robots*, pp. 199–206 (cit. on p. 174).

Schilling, M. and H. Cruse (2017). “ReaCog, a Minimal Cognitive Controller Based on Recruitment of Reactive Systems”. In: *Frontiers in Neurorobotics* 11, 3 (cit. on pp. 54, 175).

Schilling, M., T. Hoinville, J. Schmitz, and H. Cruse (2013). “Walknet, a Bio-Inspired Controller for Hexapod Walking”. In: *Biological Cybernetics* 107(4), pp. 397–419 (cit. on pp. 18, 56 f., 63 f., 66 ff., 71, 75, 136, 171).

Schilling, M., J. Paskarbeit, T. Hoinville, A. Hüffmeier, A. Schneider, J. Schmitz, and H. Cruse (2013). “A Hexapod Walker Using a Heterarchical Architecture for Action Selection”. In: *Frontiers in Computational Neuroscience* 7, 126 (cit. on p. 31).

Schilling, M., J. Paskarbeit, J. Schmitz, A. Schneider, and H. Cruse (2012). “Grounding an Internal Body Model of a Hexapod Walker - Control of Curve Walking in a Biologically Inspired Robot”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2762–2768 (cit. on pp. 92, 97, 126, 172).

Schmitz, J. and D. Düsterhus (2009). *Sensory Control of Backward Walking in the Stick Insect: II. Intersegmental Influences.* Proc Soc Neuroscience Meeting Chicago, USA (cit. on p. 174).

Schmitz, J. and G. Haßfeld (1989). “The Treading-on-Tarsus Reflex in Stick Insects: Phase-Dependence and Modifications of the Motor Output During Walking”. In: *Journal of Experimental Biology* 143(1), pp. 373–388 (cit. on p. 60).

Schmitz, J., A. Schneider, M. Schilling, and H. Cruse (2008). “No Need for a Body Model: Positive Velocity Feedback for the Control of an 18-DoF Robot Walker”. In: *Applied Bionics and Biomechanics* 5(3), pp. 135–147 (cit. on pp. 9, 62, 97).

Schneider, A., H. Cruse, and J. Schmitz (2006). “Decentralized Control of Elastic Limbs in Closed Kinematic Chains”. In: *International Journal of Robotics Research* 25(9), pp. 913–930 (cit. on pp. e, 9, 17).

Schneider, A., J. Paskarbeit, M. Schäffersmann, and J. Schmitz (2011). “Biomechatronics for Embodied Intelligence of an Insectoid Robot”. In: *Proceedings of the 4th International Conference on Intelligent Robotics and Applications*. Springer, pp. 1–11 (cit. on pp. 19, 29).

— (2012). “HECTOR, a New Hexapod Robot Platform with Increased Mobility — Control Approach, Design and Communication”. In: *Advances in Autonomous Mini Robots*. Springer, pp. 249–264 (cit. on pp. 19, 39 f.).

Schumm, M. and H. Cruse (2006). “Control of Swing Movement: Influences of Differently Shaped Substrate”. In: *Journal of Comparative Physiology A* 192(10), pp. 1147–1164 (cit. on pp. 100, 103).

Schütz, C. and V. Dürr (2011). “Active Tactile Exploration for Adaptive Locomotion in the Stick Insect”. In: *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 366(1581), pp. 2996–3005 (cit. on pp. 60, 73).

Schwegmann, A., J. P. Lindemann, and M. Egelhaaf (2014). “Depth Information in Natural Environments Derived From Optic Flow by Insect Motion Detection System: A Model Analysis”. In: *Frontiers in Computational Neuroscience* 8, p. 83 (cit. on p. 162).

Send, W., M. Fischer, K. Jebens, R. Mugrauer, A. Nagarathinam, and F. Scharstein (2012). “Artificial Hinged-Wing Bird With Active Torsion and Partially Linear Kinematics”. In: *Proceedings of the 28th Congress of the International Council of the Aeronautical Sciences*, pp. 23–28 (cit. on p. 2).

Sensinger, J. and J. H. Lipsey (2012). “Cycloid vs. Harmonic Drives for use in High Ratio, Single Stage Robotic Transmissions”. In: *2012 IEEE International Conference on Robotics and Automation*, pp. 4130–4135 (cit. on p. 174).

Sensinger, J. and R. Weir (2005). “Design and Analysis of a Non-Backdrivable Series Elastic Actuator”. In: *Proceedings of the 9th International Conference on Rehabilitation Robotics*, pp. 390–393 (cit. on p. 17).

Smith, R. (2001). *Open Dynamics Engine*. URL: http://www.ode.org/ (cit. on p. 39).

Sorbello, R., A. Chella, C. Calí, M. Giardina, S. Nishio, and H. Ishiguro (2014). “Telenoid Android Robot as an Embodied Perceptual Social Regulation Medium Engaging Natural Human–Humanoid Interaction”. In: *Robotics and Autonomous Systems* 62(9). Intelligent Autonomous Systems, pp. 1329–1341 (cit. on p. 1).

Spenneberg, D., F. Kirchner, and H. Zhang (2007). “The Bio-Inspired SCORPION Robot: Design, Control & Lessons Learned”. In: Climbing & Walking Robots, Towards New Applications, pp. 197–218 (cit. on pp. 6, 9, 21).

Steingrube, S., M. Timme, F. Wörgötter, and P. Manoonpong (2010). “Self-Organized Adaptation of Simple Neural Circuits Enables Complex Robot Behavior”. In: *Nature Physics*, pp. 224–230 (cit. on p. 11).

Strauß, R. and M. Heisenberg (1990). “Coordination of Legs During Straight Walking and Turning in *Drosophila melanogaster*”. In: *Journal of Comparative Physiology A* 167(3), pp. 403–412 (cit. on p. 11).

Szufnarowski, F., A. Gollin, and A. Schneider (2014). “Insect-Inspired Drive Strategy Substantially Improves the Performance of a Piezo Motor”. In: *Sensors and Actuators A: Physical* 219, pp. 13–23 (cit. on p. e).

Tanoto, A., F. Werner, and U. Rückert (2012). “Multi-Robot System Validation: From Simulation to Prototyping with Mini Robots in the Teleworkbench”. In: *Advances in Autonomous Mini Robots: Proceedings of the 6-th AMiRE Symposium*. Springer Berlin Heidelberg, pp. 147–159 (cit. on pp. 160, 162, 167).

Tedeschi, F. and G. Carbone (2014). “Design Issues for Hexapod Walking Robots”. In: *Robotics* 3(2), pp. 181–206 (cit. on p. 8).

Theunissen, L. M., H. Bekemeier, and V. Dürr (2014). *Stick Insect Locomotion.* URL: http://movement.cit-ec.de/index.php?id=stickinsect (cit. on pp. e, 18, 45 f., 49, 170).

— (2015). “Comparative Whole-Body Kinematics of Closely Related Insect Species With Different Body Morphology”. In: *Journal of Experimental Biology* 218(3), pp. 340–352 (cit. on p. 49).

Theunissen, L. M., M. Hertrich, C. Wiljes, E. Zell, C. Behler, A. F. Krause, H. Bekemeier, P. Cimiano, M. Botsch, and V. Dürr (2014). “A Natural Movement Database for Management, Documentation, Visualization, Mining and Modeling of Locomotion Experiments”. In: *Biomimetic and Biohybrid Systems: Proceedings of the International Conference, Living Machines.* Vol. 8608. Springer (cit. on pp. 46 ff., 50 f., 133 f.).

Theunissen, L. M., S. Vikram, and V. Dürr (2014). “Spatial Co-Ordination of Foot Contacts in Unrestrained Climbing Insects”. In: *Journal of Experimental Biology* 217(18), pp. 3242–3253 (cit. on p. 73).

Thorson, I. and D. G. Caldwell (2011). “A Nonlinear Series Elastic Actuator for Highly Dynamic Motions”. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 390–394 (cit. on p. 26).

Todd, D. (1985). *Walking Machines: an Introduction to Legged Robots.* Kogan Page (cit. on p. 7).

Tsagarakis, N. G., M. Laffranchi, B. Vanderborght, and D. G. Caldwell (2009). “A Compact Soft Actuator Unit for Small Scale Human Friendly Robots.” In: *Proceedings of the IEEE International Conference on Robotics and Automation.* IEEE, pp. 4356–4362 (cit. on pp. 17, 26).

Uhlig, K., A. Spickenheuer, L. Bittrich, and G. Heinrich (2013). “Development of a Highly Stressed Bladed Rotor Made of a CFRP Using the Tailored Fiber Placement Technology”. In: *Mechanics of Composite Materials* 49(2), pp. 201–210 (cit. on p. 21).

Van Ham, R., B. Vanderborght, M. Van Damme, B. Verrelst, and D. Lefeber (2007). “MACCEPA, The Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator: Design and Implementation in a Biped Robot”. In: *Robotics and Autonomous Systems* 55(10), pp. 761–768 (cit. on p. 17).

Vanderborght, B., A. Albu-Schäffer, A. Bicchi, E. Burdet, D. G. Caldwell, R. Carloni, M. Catalano, O. Eiberger, W. Friedl, G. Ganesh, M. Garabini, M. Grebenstein, G. Grioli, S. Haddadin, H. Hoppner, A. Jafari, M. Laffranchi, D. Lefeber, F. Petit, S. Stramigioli, N. G. Tsagarakis, M. Van Damme, R. Van Ham, L. Visser, and S. Wolf (2013). "Variable Impedance Actuators: A Review". In: *Robotics and Autonomous Systems* 61(12), pp. 1601–1614 (cit. on p. 17).

Vaucanson, J. (1738). *Le mécanisme du fluteur automate présenté à messieurs de l'Académie royale des sciences.* Chez J. Guerin (cit. on p. 1).

von Holst, E. (1935). "Die Koordination der Bewegung bei den Arthropoden In Abhängigkeit von zentralen und peripheren Bedingungen". In: *Biological Reviews* 10(2), pp. 234–261 (cit. on p. 11).

Von Wattenwyl, C. B. and J. Redtenbacher (1908). *Die Insektenfamilie der Phasmiden.* W. Engelmann (cit. on p. 19).

Wada, K. and T. Shibata (2007). "Living With Seal Robots—Its Sociopsychological and Physiological Influences on the Elderly at a Care House". In: *IEEE Transactions on Robotics* 23(5), pp. 972–980 (cit. on p. 1).

Wallace, H. W. (1945). "Vehicle". US Patent US2371368 A (cit. on p. 7).

Walter, W. G. (1950). "An Imitation of Life". In: *Scientific American* (182) (2), pp. 42–45 (cit. on p. 1).

Webb, B. (2001). "Can Robots Make Good Models of Biological Behaviour?" In: *Behavioral and Brain Sciences* 24(06), pp. 1033–1050 (cit. on pp. 2, 71, 159).

Webb, B. and T. Scutt (2000). "A Simple Latency-Dependent Spiking-Neuron Model of Cricket Phonotaxis". In: *Biological Cybernetics* 82 (3) (cit. on p. 2).

Weiner, S. and P. M. Dove (2003). "An Overview of Biomineralization Processes and the Problem of the Vital Effect". In: *Reviews in Mineralogy and Geochemistry* 54(1), pp. 1–29 (cit. on p. 21).

Wendler, G. (1964). "Laufen und Stehen der Stabheuschrecke *Carausius morosus*: Sinnesborstenfelder in den Beingelenken als Glieder von Regelkreisen". In: *Zeitschrift für vergleichende Physiologie* 48(2), pp. 198–250 (cit. on pp. 12 ff., 20 f.).

— (1968). "Ein Analogmodell der Beinbewegungen eines laufenden Insekts". In: *Kybernetik 1968.* Vol. 18. Beihefte zur Zeitschrift „Elektronische Rechenanlagen", pp. 68–74 (cit. on pp. 12 f.).

Wilkinson, S. (2000). "'Gastrobots'—Benefits and Challenges of Microbial Fuel Cells in Food Powered Robot Applications". In: *Autonomous Robots* 9(2), pp. 99–111 (cit. on p. 22).

Wille, J. (1924). "Die verschiedenen Bewegungsarten des *Rhipipteryx chopardi*". In: *Zoologischer Anzeiger* 61, pp. 49–72 (cit. on pp. 12, 45).

Wilson, D. (1966). "Insect Walking". In: *Annual Review of Entomology* 11(1), pp. 103–122 (cit. on pp. 9, 12, 56, 135).

Wosnitza, A., T. Bockemühl, M. Dübbert, H. Scholz, and A. Büschges (2013). "Inter-Leg Coordination in the Control of Walking Speed in *Drosophila*". In: *Journal of Experimental Biology* 216(3), pp. 480–491 (cit. on pp. 11, 72, 136).

Wosnitza, A., J. Engelen, and M. Gruhn (2013). "Segment-Specific and State-Dependent Targeting Accuracy of the Stick Insect". In: *Journal of Experimental Biology*, pp. 4172–4183 (cit. on p. 73).

Wright, C., A. Buchan, B. Brown, J. Geist, M. Schwerin, D. S. Rollinson, M. Tesch, and H. Choset (2012). "Design and Architecture of the Unified Modular Snake Robot". In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4347–4354 (cit. on p. 27).

Yoshikawa, T. and X.-Z. Zheng (1993). "Coordinated Dynamic Hybrid Position/Force Control for Multiple Robot Manipulators Handling One Constrained Object". In: *The International Journal of Robotics Research* 12(3), pp. 219–230 (cit. on p. 16).

Zakotnik, J., T. Matheson, and V. Dürr (2006). "Co-Contraction and Passive Forces Facilitate Load Compensation of Aimed Limb Movements". In: *Journal Of Neuroscience* 26(19), pp. 4995–5007 (cit. on p. 22).

Ziegenmeyer, M., A. Rönnau, T. Kerscher, J. M. Zöllner, and R. Dillmann (2009). "Die sechsbeinige Laufmaschine LAURON IVc". In: *Autonome Mobile Systeme 2009.* Informatik aktuell. Springer, pp. 225–232 (cit. on p. 9).

Zill, S. N., J. Schmitz, and A. Büschges (2004). "Load Sensing and Control of Posture and Locomotion". In: *Arthropod Structure & Development* 33(3), pp. 273–286 (cit. on p. 21).

Zollikofer, C. (1994a). "Stepping Patterns in Ants I. Influence of Speed and Curvature". In: *Journal of Experimental Biology* 192, pp. 95–106 (cit. on p. 11).

— (1994b). "Stepping Patterns in Ants III. Influence of Load". In: *Journal of Experimental Biology* 192, pp. 119–127 (cit. on p. 45).