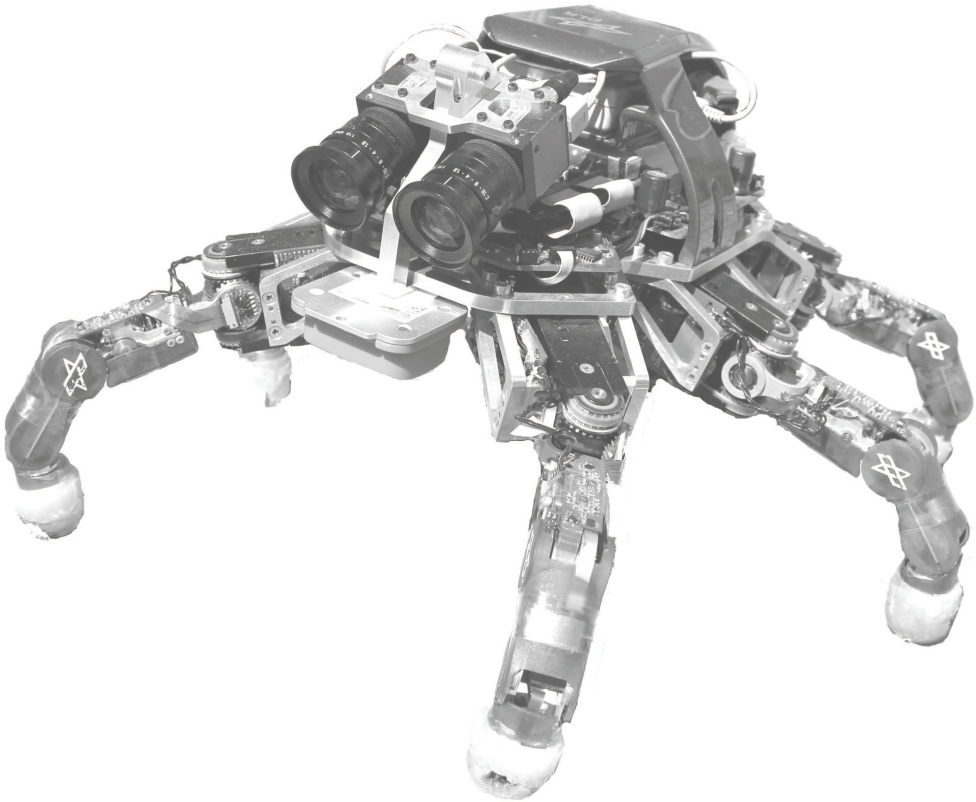
Locomotion and Pose Estimation in Compliant, Torque-Controlled Hexapedal Robots

Dipl.-Ing. Martin Görner, M.S.



Dissertation

urn:nbn:de:gbv:ilm1-2017000255

Locomotion and Pose Estimation in Compliant, Torque-Controlled Hexapedal Robots

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur
(Dr.-Ing.)

vorgelegt
der Fakultät für Maschinenbau der
Technischen Universität Ilmenau

von Herrn
Dipl.-Ing. Martin Görner, M.S.
aus Dessau

1. Gutachter:	Univ.-Prof. Dipl.-Ing. Dr. med. (habil.) Hartmut Witte Technische Universität Ilmenau
2. Gutachter:	Prof. Dr.-Ing. Gerd Hirzinger Technische Universität München
3. Gutachter:	Prof. Auke Jan Ijspeert, PhD École polytechnique fédérale de Lausanne

Tag der Einreichung:	01.11.2016
Tag der wissenschaftlichen Aussprache:	03.05.2017

urn:nbn:de:gbv:ilm1-2017000255

Für Lotte, Till und Nils

Acknowledgements

This thesis summarizes parts of the research that I have conducted during my employment at the Institute of Robotics and Mechatronics at the German Aerospace Center (DLR) in Oberpfaffenhofen. First of all, I want to thank Prof. Gerd Hirzinger and Prof. Alin Albu-Schäffer, the former and the current directors of the institute, for their support and their great efforts to provide an inspiring working environment. Next, I want to thank Prof. Hartmut Witte from Technische Universität Ilmenau for his advice and guidance while writing this thesis. I am very thankful to Markus Grebenstein and Christoph Borst who got me started to work on legged robots for my diploma thesis and always supported my work on the DLR Crawler. Many thanks go to Michael Suppa who encouraged the formation of a mobile robots group at the institute, and thus brought together smart people that were working on isolated topics at this time. In addition, I thank Christian Ott, the head of my department, for supporting me in juggling family and work.

The work presented within this thesis would not have been possible without great colleagues and their preceding or parallel work. Therefore, I want to thank the team that created the outstanding DLR Hand II, which provided the technological basis for the DLR Crawler. Also, I want to thank the members of the mobile robots group for the good collaboration. Many thanks go to Thomas Wimböck for fruitful discussions on control and for providing the initial control software infrastructure for the DLR Crawler. I would also like to thank Jörg Butterfass for the many hours he helped me to repair broken legs, and Thomas Bahls for his efforts in fixing and adapting the Spacewire setup. Especially during the initial phase of experimentation, I discovered several broken soldering points at difficult to reach locations. Those were always quickly fixed by my great colleagues of the electronics workshop. Thanks for this! Another thanks goes to Heiko Hirschmüller who developed and provided a high-performance stereo vision algorithm that was the basis of the visual navigation framework implemented for the DLR Crawler. I am very thankful for working with Annett Stelzer who developed this great navigation framework, and thus added much more sense to the Crawler project. She is a very smart person with a great sense of humour. Thanks for all the fruitful discussions and the fun hours of lab and office work! Further thanks go to Alexander Dietrich and Rachel Hornung for proofreading parts of this manuscript.

One of the great strengths of the institute is the familiar and open atmosphere, which was fostered by events like summer parties, Christmas celebrations, and great summer and winter weekends spend in a cottage at the Hausberg (Thanks to Oliver Eiberger and Andrea Schwier for organising and hosting those weekends!). For me personally, this atmosphere was created by many of my colleagues during joint work, lunch and coffee breaks, while having fruitful discussions, or just talking about nonsense. In alphabetical order my thanks go to my former and current colleagues Andreas Baumann, Wieland Bertleff, Robert Burger, Oliver Eiberger, Matthias Fuchs, Iris Grixa, Robert Haslinger, Hannes Höppner, Rachel Hornung, Simon Kielhöfer, Sophie Lantermann, Elmar Mair, Matthias Nickl, Alexander Nothhelfer, Florian Petit, Felix Rueß,

Simon Schätzle, Korbinian Schmid, Florian Schmidt, Martin Stelzer, Georg Stillfried, Michael Strohmayr (additional thanks for pushing me to finish), Teodor Tomic, Holger Urbanek, Jörn Vogel, Armin Wedler, and Roman Weitschat.

I am deeply grateful for the support and encouragement that I received from my entire family and my friends. Many thanks go to my parents, my grandparents, and my sister, as well as my parents in law. My warmest thanks belong to Andrea for being a great partner and friend, for her love and support, for sharing the same kind of humour, and for correcting my perspective when frustration about this work won over enthusiasm. Most of it all, I am thankful for our children, and the love, joy, and challenges they bring to our life.

Munich, 2016 Martin Görner

Kurzfassung

Für die planetare Exploration sowie den Einsatz in Katastrophengebieten sind autonome Laufroboter zunehmend von Interesse. In diesen Szenarien sollen sie den Menschen an gefährlichen oder schwer zugänglichen Orten ersetzen und dort Erkundungseinsätze sowie Probenahmen in schwierigem Gelände durchführen. Unter der Vielzahl an möglichen Systemen bieten im Besonderen kleinere Sechsbeiner einen sehr guten Kompromiss zwischen Stabilität, hoher Beweglichkeit, Vielseitigkeit und einer vertretbaren Komplexität der Regelung. Ein weiterer Vorteil ist ihre Redundanz, die es ihnen erlaubt, den Ausfall einzelner Beine mit geringem Aufwand zu kompensieren. Dementgegen ist die beschränkte Rechenkapazität ein Nachteil der reduzierten Größe. Um diesen auszugleichen und das autonome Agieren in einer unbekannten Umgebung zu ermöglichen, werden daher einfache und effiziente Algorithmen benötigt, die im Zusammenspiel jedoch ein komplexes Verhalten erzeugen.

Auf dem Weg zum autonom explorierenden Laufroboter entwickelt diese Arbeit einen robusten, adaptiven und fehlertoleranten Laufalgorithmus sowie eine 6D Eigenbewegungsschätzung für nachgiebige, drehmomentgeregelte Sechsbeiner. Besonders herauszustellen ist, dass alle in der Arbeit vorgestellten Algorithmen ausschließlich die propriozeptive Sensorik der Beine verwenden. Durch diesen Ansatz kann der Laufprozess von anderen Prozessen, wie der Navigation, getrennt und somit der Datenaustausch effizient gestaltet werden.

Für die Fortbewegung in unebenem Gelände kombiniert der vorgestellte Laufalgorithmus eine flexible, biologisch inspirierte Gangkoordination mit verschiedenen Einzelbeinreflexen und einer nachgiebigen Gelenkregelung. Hierbei übernimmt die Gangkoordination die zeitliche Steuerung der Schrittfolge, während die Einzelbeinreflexe für eine räumliche Variation der Fußtrajektorien zuständig sind. Die nachgiebige Gelenkregelung reduziert interne Kräfte und erlaubt eine Anpassung der Gelenksteifigkeiten an die lokalen Umgebungsbedingungen sowie den aktuellen Zustand des Roboters. Eine wichtige Eigenschaft des Laufalgorithmus ist seine Fähigkeit, den Ausfall einzelner Beine zu kompensieren. In diesem Fall erfolgt eine Adaption der Gangkoordination über die Erneuerung der Nachbarschaftsbeziehungen der Beine. Zusätzlich verbessern eine Veränderung der Pose und eine Erhöhung der Gelenksteifigkeiten die Stabilität des durch den Beinverlust beeinträchtigten Roboters.

Gleich dem Laufalgorithmus verwendet die 6D Eigenbewegungsschätzung nur die Messungen der propriozeptiven Sensoren der Beine. Hierbei arbeitet der Algorithmus in einem dreistufigen Verfahren. Zuerst berechnet er mit Hilfe der Beinkinematik und einer Optimierung die Pose des Roboters. Nachfolgend bestimmt er aus den Gelenkmomentmessungen den Gravitationsvektor und berechnet daraus die Neigungswinkel des Systems. Eine Fusion dieser Werte mit den Nick- und Rollwinkeln der ersten Stufe stabilisiert daraufhin die gesamte Odometrie und reduziert deren Drift.

Alle in dieser Arbeit entwickelten Algorithmen wurden mit Hilfe von Simulationen sowie Experimenten mit dem drehmomentgeregelten DLR Krabbler erfolgreich validiert.

Abstract

Several scenarios, such as disaster response or terrestrial and extra-terrestrial exploration, comprise environments that are dangerous or even inaccessible for humans. In those cases, autonomous robots pose a promising alternative to render such endeavours possible. While most of today's robotic explorers are wheeled or tracked vehicles, legged systems gained increased attention in recent years. With their unique combination of omnidirectional mobility and intrinsic manipulation capabilities, they are envisioned to serve as the rough terrain specialists in scouting or sample and return missions. Especially, small to mid-size hexapods are of great interest for those scenarios. Providing static stability across a wide range of walking speeds, they offer an attractive trade-off between versatility and complexity. Another important advantage is their redundancy, allowing them to tolerate the loss of single legs. However, due to their small size, the computational on-board resources are limited. Thus, the use of smart and efficient algorithms is of utmost importance in order to enable autonomous operation within a priori unknown rough environments.

Working towards such autonomous robotic scouts, this thesis contributes with the development, implementation, and test of a self-contained walking layer as well as a 6 degrees of freedom (DOF) leg odometry for compliant, torque-controlled, hexapedal robots. Herein, the important property of all presented algorithms is the sole use of proprioceptive measurements provided by the legs, i. e. joint angles and joint torques. Especially the joint torque sensors improve the walking process by enabling the use of sensitive compliance controllers and distributed collision detection.

Comprising a set of algorithms, the walking layer organises and structures the walking process in order to generate robust, adaptive, and leg loss tolerant locomotion in uneven terrain. Furthermore, it encapsulates the walking process, and thus hides its complexity from higher-level algorithms such as navigation. Its three main functional components are a flexible, biologically-inspired gait coordination algorithm, single leg reflexes, and active joint compliance control. Thereof, the gait coordination algorithm realises temporal adaptation of the step sequence while reflexes adjust the leg trajectories to the local terrain. The joint compliance control reduces internal forces and allows for situation dependent stiffness adjustments. An algorithmic extensions to the basic gait coordination enables the immediate adaptation to leg loss. In combination with stiffness and pose adjustments, this allows the hexapod to retain stable locomotion on five legs. In order to account for the emerging gait, the leg odometry algorithm employs an optimisation approach to obtain a kinematics-based pose estimate from joint angle measurements. Fusing the resulting pitch and roll angle estimates with joint-torque-measurement-based attitude data, reduces the associated drift, and thus stabilises the overall pose estimate.

Various simulations and experiments with the six-legged, torque-controlled DLR Crawler demonstrate the effectiveness of the proposed walking layer as well as the 6-DOF leg odometry.

Contents

List of Tables

List of Figures

List of Symbols

$\hat{}$	Hat indicates an estimate
Δ	An error value
α	Yaw angle
β	Pitch angle
ϵ	Sum of squared errors
γ	Roll angle
μ	Coefficient of friction
$\boldsymbol{\lambda}$	Vector of Lagrange multipliers
$\boldsymbol{\omega}$	Vector of angular velocities
$\boldsymbol{\tau}$	Vector of torques
$\dot{\boldsymbol{\theta}}_j$	Vector of joint velocities
$\boldsymbol{\theta}_j$	Vector of joint angles
d	Distance
i	Counter variable
j	Counter variable
k	Gain parameter
l	Kinematic parameter - length
m	Mass or mechanism output (Cruse's rules)
n	Total number
r	Radius
t	Time or time step

w	Weight parameter
$V(\boldsymbol{\theta}_j)$	Sum of spring potentials
$\mathbf{p}$	Vector of linear momentum or vector of Cartesian coordinates
$\mathbf{f}$	Vector of forces
$\mathbf{q}$	Unit quaternion in vector form
$\mathbf{r}$	Vector of linear displacements
$\mathbf{t}$	Vector of linear displacements
$\mathbf{v}$	Vector of linear velocities
$\mathbf{x}$	Vector of Cartesian coordinates
$\mathbf{D}$	Damping matrix
$\mathbf{I}$	Inertia tensor or identity matrix
$\mathbf{J}$	Jacobian matrix
$\mathbf{u}$	Vector of control signals
$\mathbf{L}$	Vector of angular momentum
$\mathbf{M}$	Mass matrix
$\mathbf{P}$	Error covariance matrix
$\mathbf{Q}$	Noise covariance matrix
$\mathbf{R}$	Rotation matrix
$\mathbf{T}$	Transformation matrix
$\mathbf{U}$, $\mathbf{D}$, $\mathbf{V}$	Matrices resulting from a singular value decomposition

1 Introduction

1.1 Motivation

Many scenarios, such as disaster recovery or terrestrial and extra-terrestrial exploration, comprise environments that are dangerous or even inaccessible for humans. In those cases, autonomous robots pose a promising alternative in order to map an area of interest, to collect environmental data or to take and return samples. Required to operate in challenging terrain, legged systems are expected to show superior performance over wheeled and tracked vehicles. Their advantages are omnidirectional mobility as well as the fact that legs do not require a continuous path of ground contact. Nevertheless, these advantages come at the price of increased complexity and the need for high structural and algorithmic robustness. Even after more than four decades of active research in legged robotics and much progress on technology and algorithms, only few robots are able to operate outside controlled laboratory environments. However, incidents like the 2011 nuclear disaster at Fukushima triggered industry and academia to create more robust and capable legged systems that are applicable to real world scenarios. In this context, the latest DARPA robotics challenge[1] is just one example for the strong efforts taken to push the limits in legged robotics.

While current research focusses on versatile humanoid and quadrupedal systems, there is a recurring interest in using groups of small to mid-size hexapedal robots for terrestrial and extra-terrestrial exploration. Within different mission scenarios, the hexapods are envisioned to serve as rough terrain scouts, offering several advantages over their larger bi- and quadrupedal counterparts. Clearly, operating in statically stable regimes is the most important advantage of hexapods as it drastically eases their control. Another advantage of hexapedal robots is their inherent redundancy with respect to the number of legs required for generating statically stable gaits. Therefore, they are able to quickly adapt to the damage or loss of single legs while maintaining static stability. Furthermore, using their body or any of their legs, hexapods are able

[1] http://archive.darpa.mil/roboticschallenge/

to manipulate their environment and to explore its properties by the sense of touch. As this is also true for bi- or quadrupeds, six-legged robots more easily preserve a statically stable configuration during those operations. Thus, with a proper design of the mechatronic system, the legs of a robotic hexapod enable a great variety of locomotion, perception, and manipulation skills.

While realising such versatile six-legged robots is a long-term goal, the most important and basic task in exploration is to move reliably and safely from point A to point B in an a priori unknown environment. In order to fulfil this task, the robot has to provide sufficient rough terrain mobility as well as navigation capabilities which should not rely on any external infrastructure. Thus, having no absolute reference, the robot needs to collect all relevant data by itself. While doing so, it has to account for uncertain and incomplete information about its own state, its local environment as well as the interaction therewith. Consequently, it is crucial for the robot to provide structural and algorithmic robustness as well as the ability to constantly adapt to changes of the terrain or the operational conditions.

1.2 Approach and Contributions

Working towards the creation of robust robotic scouts for exploration missions, this thesis focusses on two essential capabilities of torque-controlled, six-legged walking robots. The first and most important one is the capability to generate robust, adaptive, and leg-loss-tolerant gaits on natural terrain, whereas the second is the estimation of the pose of the robot. As already mentioned above, all developments have to account for uncertain and incomplete knowledge about the environment, the state of the robot, and its interaction with the local terrain. Having no external reference or a priori information, the robot needs to acquire all necessary data by itself. For this reason, the use of complex and parameter-sensitive model-based control algorithms is inappropriate. Instead, robust and adaptive locomotion should emerge from the interaction of a properly designed electro-mechanical system with a set of simple, distributed algorithms maximally exploiting the available sensor data of the legs, i. e. joint angle and joint torque measurements.

Therefore, this thesis aims to verify the following two hypotheses:

1. **Simple, distributed algorithms that only use the measurements provided by the proprioceptive sensors of the legs are able to generate robust, adaptive, and leg-loss-tolerant walking in torque-controlled hexapedal robots.**

2. For walking, leg proprioception alone is sufficient to estimate the pose of a torque-controlled hexapod with respect to its starting point.

Concerning these hypotheses, the list below states the contributions of this thesis.

- Hexapedal walking
  - Development, implementation and test of a robust, adaptive, and leg-loss-tolerant walking layer for compliant, torque-controlled hexapods, which
    - enables omnidirectional walking with a flexible, emergent gait based on the well-known coordination mechanisms of stick insects [Cruse, 1990, Cruse et al., 1998],
    - produces stable gaits with beneficial forward-directed waves of protractions which result in an increased stability margin,
    - employs three simple reflexes to negotiate obstacles within the walking height autonomously,
    - encapsulates the whole walking process and provides simple interfaces to higher-level algorithms such as navigation,
    - provides a binary safety value which allows higher-level algorithms to trigger a more risky behaviour,
    - immediately adapts the inter-leg couplings, and thus the gait coordination, to leg loss (within 1 ms after the detection of the incident),
    - improves the stability/smoothness of locomotion in case of leg loss by two simple adjustments, i. e. a shift of the centre of gravity (COG) with respect to the support polygon as well as a joint stiffness adaptation.
  - Calculation of feasible velocity commands, resulting in well-coordinated gaits for the fully functional robot as well as the impaired robot suffering leg loss, in order to inform higher-level algorithms.
- Pose estimation
  - Development, implementation, and test of a leg-proprioception-based 6 degrees of freedom (DOF) odometry for statically stable walking robots with compliant joints,
    - able to handle emergent gaits with varying ground contact configurations and
    - using joint-torque-based attitude data to stabilize the pose estimates.

For the work presented, the term “robustness” refers to the algorithmic level. There it addresses the robustness of the compliance controller with respect to external disturbances, uncertain geometric parameters, and uncertain measurements of joint angles and joint torques (but not in the strict sense of formal robust control). In limiting interaction torques, the underlying joint torque control provides robustness with respect to soft and hard impacts. Furthermore, the gait coordination algorithm is robust with respect to temporal delays due to reflexes, and within bounds, to the choice of coordination mechanism weight parameters. Moreover, the pose estimation is robust with respect to single leg slippage and varying numbers of legs in contact. Within this work the term “adaptive” refers to the gait coordination and the reflex-based, spatio-temporal adjustments of the foot trajectories to comply with the terrain. In addition, it addresses the variation of the joint compliance control parameters by higher-level algorithms. For pose estimation, adaptive relates to providing and using different sets of tuning parameters for different terrains and gaits.

1.3 Outline

Starting with this introductory chapter, the thesis proceeds as follows:

- **Chapter 2** presents related work on hexapedal walking robots. First, a general overview of former and current systems is given that marks several of the milestones in the development of six-legged robots. Next, this chapter introduces common methods of gait generation and briefly discusses their underlying principles. This is followed by a section on the role of torque sensing and compliance in hexapedal walking robots. Finally, this chapter introduces related work on pose estimation for multi-legged robots.

- **Chapter 3** introduces the torque-controlled DLR Crawler, which is used as a test platform within this thesis. At first, a general overview of the hardware is given followed by a presentation of the joint compliance controller as well as the forward and inverse kinematics of the legs. Thereafter, this chapter presents a simplified dynamics model of the robot used to test the developed algorithms prior to their implementation on the real hardware.

- **Chapter 4** treats the development, implementation, and test of a robust and adaptive gait algorithm applicable to torque-controlled hexapods. Following a brief introduction on gait generation in general, a flexible gait coordination method based on Cruse’s rules is developed and evaluated. The next section

presents the calculation of the joint reference trajectories that are fed to the joint compliance controller. To allow for adaptation to the terrain, the following section introduces three leg reflexes as well as a binary safety value and demonstrates their effectiveness by simulation. Finally, the chapter closes with a section on the experimental validation of the proposed algorithms using the DLR Crawler.

- **Chapter 5** extends the gait coordination algorithm in order to allow for the instantaneous adaptation to leg loss. Using simulations, the adaptation method is tested and the influence of leg loss on the overall walking performance of the robot is evaluated. In addition, experimental results obtained with the DLR Crawler validate the approach.

- **Chapter 6** introduces a leg-proprioception-based pose estimation algorithm for torque-controlled hexapods. Following a brief introduction of the associated problem, the kinematics-based first stage of the algorithm is presented. Next, the joint-torque-based second stage of the algorithm is introduced, followed by the presentation of the error-state-Kalman-filter-based fusion of the pitch and roll angles from the first and the second stage. The subsequent sections discuss the tuning of the algorithm and present experimental results for forward walking and turning with respect to the walking velocity, the joint stiffness and the substrate of the terrain. In the end, this chapter presents some experimental results of the leg odometry for combined motions as well as for being part of a multisensor pose estimation algorithm that is integrated within the visual navigation framework of the DLR Crawler.

- **Chapter 7** summarizes the work presented and discusses the results.

- **Chapter 8** presents concluding remarks and proposes future research tasks.

Statement on the reuse of text

None of the text presented within this thesis has been previously used to obtain any degree at a university. However, the work presented within this thesis has been published in large parts at scientific conferences and in scientific journals as listed in Appendix D. Parts of the text, that I have personally written as the first author of the respective articles, are reused within this thesis. While Chapters 1 to 5 only reuse minor parts of the text published in scientific articles, Chapter 6 reuses large parts of the journal article “A leg-proprioception-based 6 DOF odometry for statically stable walking robots” published in Autonomous Robots.

2 Related Work

Today, a large number of differently sized hexapedal robots exist, showing various levels of complexity. Despite their different appearance, most of these systems serve as laboratory test platforms that are used to verify control algorithms or to test hypotheses from biology. Due to their specific design, these robots are often not well suited for a broader range of tasks. However, in recent years, the focus increasingly shifts from specialized towards highly versatile robots that are applicable to complex real world scenarios. This requires the implementation of many different skills on a single platform, and thus strong engineering efforts.

To set the perspective of this thesis, the present chapter summarizes the related work on hexapedal walking robots. Starting with an overview of former and existing robotic systems, the following sections present more detailed information on gait generation, the role of torque control and leg compliance as well as on pose estimation.

2.1 Hexapedal Walking: A Robotic Systems Overview

Despite the recently growing interest in hexapedal robots, the related research dates back 40 years and more. It has seen several waves of attention and currently shows a broad range of research tracks spanning from large and highly versatile systems to simple, miniaturized running robots.

In between the late 1970 ies and the early 1980 ies, research groups at the Ohio State University (USA) as well as the Moscow State University (Russia) independently developed impressive six-legged walking robots. The American OSU hexapod [Klein and Briggs, 1980] and the Russian robot Masha [Gorinevsky and Shneider, 1990] show many similarities: they both consist of electrically driven 3-DOF legs, use off-board computation systems, and provide three-axis force sensing capabilities within each leg. In both cases, initial research targets the generation of kinematics-based gaits, which is later complemented by research on force control algorithms. As an example for the latter, in [Klein and Briggs, 1980] and [Klein et al., 1983] the authors describe an active compliance controller, allowing the OSU hexapod to cross irregular terrain under

supervisory control. For the Russian hexapod, Gorinevsky presents the implementation and test of different force control algorithms [Gorinevsky and Shneider, 1990]. Thereof, one algorithm properly distributes the vertical ground reaction forces, and thus enables the robot to adapt its locomotion to soft soil as well as to rigid surfaces. Furthermore, by controlling the complete ground reaction force vector of each leg, the robot is able to hold itself in between two opposing slanted walls and to walk along the gap. Hence, although being very slow, both systems show a remarkable performance with respect to the available computational power at this time. In 1985, researchers at the Ohio State University presented a much larger six-legged walking robot, the Adaptive Suspension Vehicle (ASV) [Waldron and McGhee, 1986, Pugh et al., 1990]. This 5 m long and 2700 kg heavy testbed for rough terrain transportation is powered by an on-board four cylinder motorcycle engine and uses hydraulic actuators to move its 3-DOF legs. In order to drive the vehicle, the ASV operator has the choice of six operating modes with different levels of automation. Within the most automated mode, the operator only commands a desired horizontal velocity vector as well as a desired yaw rate. The ASV then employs a free gait to place its feet on secure footholds identified using an on-board scanning laser rangefinder. Although the robot is not completely autonomous, it is one of the first self-contained legged vehicles.

Within the early 1990 ies, a research group at Carnegie Mellon University developed the six-legged robot Ambler [Bares and Whittaker, 1990, Krotkov et al., 1991]. Facing the future challenge of Mars exploration, Ambler was built to quantify various performance metrics for a legged planetary rover, such as its power consumption, its position accuracy, and its walking autonomy [Krotkov et al., 1995]. Despite its mass of 2000 kg, one goal for Ambler is to achieve high power efficiency. For this reason, the hexapod has a special leg design, which arranges the single rotational as well as the two telescoping DOF within an orthogonal configuration. Therefore, the decoupled actuators are able to propel and level the robot independently. Additionally, the research group presents several interesting results with respect to an increased level of autonomy. One of those results is a height map of the terrain created from the data of a scanning laser rangefinder. A planning module uses this map to generate feasible kinematic trajectories for the robot. Finally, Roston and Krotkov present a dead-reckoning-based leg odometry that allows the robot to keep track of its pose while walking along the planned path [Roston and Krotkov, 1992] .

In addition to the large robots developed within the 1990 ies, several research groups present smaller laboratory systems intended for testing distributed gait and control algorithms. Some of those draw inspiration from experimental results on insect locomotion, while others follow a more technical approach in implementing finite state

machines. Concerning their mechanical design, many of these robots loosely "copy" the stick insect. Thus, their 3-DOF legs are arranged in an M-shape configuration with respect to the frontal view. Resulting in a large leg workspace, this configuration enables farther reach and simultaneously realises a stabilising low COG. Some examples for this design are Robot II from Case Western Reserve University [Espenschied et al., 1996], the hexapod MAX from Technical University of Munich [Pfeiffer et al., 1995, Pfeiffer, 2007], the series of LAURON robots from FZI in Karlsruhe [Berns et al., 1994, Cordes et al., 1997] as well as the series of TARRY robots from University of Duisburg-Essen [Frik et al., 1999]. In addition to a similar kinematic design, each of these systems provides force sensors that are embedded within their legs. However, except for Robot II and its active compliance control, none of the other robots uses any force control algorithm. They only employ these sensors for contact detection. Furthermore, Robot II is the only system providing some passive compliance by design. Hence, the telescoping, spring-loaded mechanism within its distal links allows Robot II to more easily accommodate its posture to rough terrain.

In the early 1990 ies, two other important hexapods were built at the MIT Artificial Intelligence Lab. Designed for increased autonomy and robustness, Attila and its copy Hannibal [Angle and Brooks, 1990] served for testing distributed, behaviour-based control algorithms [Ferrell, 1993]. Within this approach, a central oscillator provides a basic pacemaker signal, while each of the legs generates adequate joint trajectories on its own. Using inter-leg communication pathways, the legs directly influence their neighbours by sending inhibition or activation signals. Furthermore, the controller adapts to leg faults by re-routeing the inter-leg communication according to a fixed scheme. Concerning the mechanical design, Hannibal and Attila provide 18 electrically driven joints as most of the other robots but additionally host one central motor. The task of this motor is to simultaneously change the orientation of all legs within the sagittal plane of the robot. Thus, Hannibal and Attila are able to keep their legs vertical on an incline while their body stays in parallel to the surface. As most of the other systems presented, Hannibal and Attila provide strain-gauge-based force sensors within each leg. However, the related literature does not report any implementation of force control algorithms for these robots.

In contrast, force control applications are one of the driving factors for the development of the hexapod Katharina [Schmucker et al., 1996] at the Fraunhofer Institute for Factory Operation and Automation in Magdeburg. Extending the results obtained with the Russian robot Masha, the researchers implemented an algorithm that controls the external contact force vector during insertion or drilling operations by properly distributing the ground reaction forces of all legs. Thus, in addition to

realising smooth locomotion and postural adaptation, the robot employs force control to carefully manipulate its environment. Another robot aiming at force controlled locomotion and manipulation is LEMUR IIa. It was developed at the Jet Propulsion Laboratory in order to maintain and inspect orbital structures during future space missions [Kennedy et al., 2006]. Its 4-DOF legs do not only support and propel the robot but also serve as manipulators. Using a quick-connect mechanism at each end-effector, the robot can easily exchange its tools. As an additional feature, LEMUR IIa provides a ring-mounted stereo camera that moves along its circumference. By this, the robot has an omnidirectional visual coverage enabling vision guided manipulation with each of its legs.

Apart from those highly articulated systems, several researchers present six-legged robots comprising underactuated, passively compliant legs. Drawing inspiration from cockroaches, these system are intended for research on robust dynamic locomotion by exploiting a self-stabilising configuration as proposed in [Full and Koditschek, 1999]. One prominent example is the robot RHex [Saranli et al., 2001], which was developed with DARPA funding by a consortium of US American and Canadian universities. This robot has six identical, C-shaped, passively compliant legs that are individually driven by hip mounted DC motors. Thereby, each hip motor modulates its rotational velocity to realise short swing and longer stance phases according to a specific gait pattern. During ground contact, each of the C-shaped legs acts as a spring that extends and shortens depending on the static and dynamic loading as well as the actual location of the contact point. Thus, at low speeds the compliant legs mainly adjust the posture of RHex to the roughness of the terrain, while the properly tuned system exhibits self-stabilising, limit-cycle-like dynamics at higher velocities. In one of its versions, the robot even achieves speeds up to 2.7 m/s. In addition, RHex masters many different locomotion tasks such as crossing rocky terrain, climbing steep slopes and ascending stairs [Johnson et al., 2011], as well as bipedal running with an upright posture [Neville et al., 2006]. Apart from RHex, the Sprawl family of robots from Stanford University [Cham et al., 2002] also exploits a mechanically self-stabilising sprawled posture for robust dynamic locomotion. In comparison to RHex, these robots possess a reverse actuation scheme with active telescoping legs and passively compliant rotary hips. However, with proper tuning and periodic feedforward actuation, the Sprawl robots establish very robust dynamic gaits [Clark and Cutkosky, 2005] as well. In recent years, several labs presented miniaturized running hexapods such as the 16 g DASH [Birkmeyer et al., 2009] or the 24 g DynaRoACH [Hoover et al., 2010]. Built of compliant composite materials, these systems employ single actuators and different linkage mechanisms to achieve self-stabilising alternating tripod gaits up to velocities

larger than one meter per second. Due to their folded and reinforced structure as well as their simple actuation scheme, these robots are highly robust to falls and collisions with the environment.

Returning to more versatile systems, the most recent examples are the robots SpaceClimber and its successor Crex from DFKI in Bremen [Bartsch et al., 2012], Hector from University of Bielefeld [Schneider et al., 2014, Paskarbeit et al., 2015], LAURON V developed at FZI Karlsruhe [Rönnau et al., 2014], and Weaver from the Autonomous Systems Lab of CSIRO in Brisbane [Bjelonic et al., 2016]. As a major improvement, all of these robots, except for HECTOR, provide at least one additional DOF within their legs. Those additional joints do not only improve their locomotion capabilities but also help to increase their static stability margin in rough terrain. Furthermore, with 4-DOF legs the robots are able to generate energy efficient leg trajectories while they have greater postural redundancy. This means, for a given set of ground contact points they can attain and hold a certain body pose using different leg configurations. This capability is especially useful for operations in obstacle-cluttered areas. In addition to the increased mobility, the 4-DOF legs enable the robots to manipulate their environment and to take samples. For this purpose, grippers can be attached to their front legs folded away during walking and engaged for manipulation. In case of Hector, its legs are not intended for active sampling or manipulation of the environment but to hold and propel the robot in rough terrain. Its novel feature is the mechanical design of its joints. Those include elastomer springs that render the robot passively compliant. This, in turn, allows researchers to test advanced, biologically-inspired leg and gait controllers that require some passive joint compliance. Inspired by insects, HECTOR has an articulated body with three segments connected by pan-tilt joints. As each of these segments hosts a pair of legs, the body joints will enable HECTOR to climb obstacles larger than its normal standing height. Besides any mechanical innovation, each of these five robots is equipped with a large set of proprioceptive and exteroceptive sensors in order to extend their perceptual capabilities.

2.2 Hexapedal Walking: Gait Generation

In hexapedal walking the most important task is to advance the body by generating coordinated leg motions with respect to the properties and the structure of the local environment. Thus, a large fraction of research targets gait generation and control. With respect to gait generation, very different methods have been developed. Those can be summarized in three large groups: fixed gaits, rhythmic-pattern-generator-based gaits, and reactive free gaits. Depending on the intended application of the

robot and the available computational resources, each method has certain advantages and disadvantages. The simplest and least adaptable implementation, fixed gaits, uses fixed stepping patterns and off-line-calculated joint trajectories. To increase the usability and flexibility of those gaits in uneven terrain, a fixed, pre-planned stepping sequence is often combined with adjustable single leg trajectories that allow adaptations of step height and step width. Those approaches usually require the definition of Cartesian foot trajectories and on-line computation of inverse kinematics. Additionally, a change of the gait pattern requires consistently designed transitions. Examples for systems that employ such gaits with fixed stepping sequences are the Adaptive Suspension Vehicle from Ohio State University [Waldron and McGhee, 1986] as well as the hexapod Katharina from Fraunhofer IFF [Ihme, 2002].

Apart from fixed gaits, many hexapedal robots employ gait pattern generators. Those form a large group of algorithms producing rhythmic patterns that can be modulated with respect to frequency, amplitude, and phase relations. One large sub-group are neural-oscillator-based central pattern generators (CPGs), which are motivated by the neural system of animals. The underlying idea is that neural circuits constitute dynamical systems capable of generating and propagating self-sustained, synchronized oscillations that produce high-dimensional rhythmic outputs based on a small number of non-rhythmic inputs. These oscillations can be further modulated by sensory feedback. With respect to robotics the CPG outputs are usually interpreted as desired joint position trajectories. A good general overview of CPGs in animals and robots is given in [Ijspeert, 2008], while [Manoonpong, 2007] presents an example for a CPG-driven six-legged walking robot. Goldschmid and collaborators extend the latter work by adding reflex mechanisms in order to improve the obstacle crossing capabilities [Goldschmidt et al., 2014]. An example for a rhythmic gait pattern generator that is not a classical CPG is presented by Guddat for the hexapod TARRY II [Guddat, 2002]. The basic algorithm employs neural networks to generate rhythmic joint angle trajectories, only using a small set of inputs such as directional motion commands and two pacemaker signals. These neural networks are trained with different off-line generated, kinematically consistent gait patterns to produce variable gaits. Combined with reflexes, they further allow the robot to cross small obstacles in uneven terrain.

The last group of gait generation methods are reactively emerging free gaits. For those, coordinated gait patterns result from decentralized inter-leg coordination mechanisms and sensor stimuli that represent the interaction with the local environment. The resulting gaits are highly versatile and adaptive in rough terrain. A prominent example are the inhibitory and excitatory coordination rules that Cruse, Dean, and collaborators identified for the stick insect [Cruse, 1990]. Those rules have been im-

plemented within a neural network, the Walknet [Cruse et al., 1998], and have been validated in producing proper hexapedal gaits by extensive simulation of kinematic and dynamic models of the stick insect. In a recent review, Schilling et al. summarize more than a decade of work related to the Walknet as well as its extensions such as curve walking and the adaptation to leg damage [Schilling et al., 2013a]. Not only in simulations but also on several hexapedal robots this set of rules is used to generate adaptive gaits. Espenschied, for example, demonstrates rough terrain walking with the 18-DOF hexapod Robot II [Espenschied, 1994]. For this purpose, he implements a reduced set of coordination rules, extends the framework to walking in the plane, and enables obstacle crossing using single leg reflexes as well as active and passive compliance of the legs. Furthermore, he presents lesion studies investigating the importance of single coordination mechanisms in establishing a stable gait pattern. Other systems that employ Cruse's coordination rules are Hannibal from MIT [Ferrell, 1995], built in the early 1990 ies, and Bill-Ant from Case Western Reserve University [Lewinger, 2005], built in 2005. In case of Hannibal, Cruse's rules are implemented for comparison with its original controller that is based on Brooks' subsumption architecture [Brooks, 1985]. For Bill-Ant, developed at the same lab as Robot II, the implementation follows Espenschied's work. The recently presented six-legged robot HECTOR [Schneider et al., 2014, Paskarbeit et al., 2015], briefly introduced above, employs the latest version of Walknet for control [Schilling et al., 2013b]. Its biologically-inspired control approach is anticipated to raise reactive walking to a level of embodied cognition in the future. A strongly reduced example of gait coordination resulting from local rules is presented by El Sayed Auf [El Sayed Auf et al., 2008] for the six-legged walking robot Oscar. In this case, just a single coordinating rule is used. This rule allows a leg to step only if all neighbouring legs are in contact, while otherwise the leg has to prolong its stance phase. In combination with an adjustable ratio of stance and swing time, a variety of gaits can be generated and delays caused by local reflexes can be compensated.

A mixture of methods for gait generation is employed by Lauron IV and Lauron V developed at FZI in Karlsruhe. Their so-called behaviour-based control approach [Kerscher et al., 2008] combines fixed stepping sequences with single leg reflexes and posture-related behaviours to generate an adaptive gait. Therein, specific behaviours slow down or accelerate single legs while the underlying basic gait pattern is not changed. In addition to the above presented methods for gait generation, on-line gait planning provides another highly adaptive solution allowing for the careful selection of footholds. However, this method is computationally expensive and requires detailed perception of the environment [Belter, 2013]. Thus, a robust implementation on

resource-limited hardware is difficult and challenging. Common to all methods for gait coordination is their combination with different sets of single leg and postural reflexes in order to increase the adaptability and robustness of the walking process.

2.3 Hexapedal Walking: Torque Sensors and Leg Compliance

In order to incorporate reflexes and to adapt the gait and the posture of a robotic hexapod to irregular terrain, kinematic joint data alone is often not sufficient. Instead, it should be combined with information on internal and external loads. This idea is supported by multiple experimental studies attributing a strong influence on the locomotive behaviour of insects to the load-sensitive mechanoreceptors embedded within their legs. [Delcomyn et al., 1996] present a good overview of the related sense organs, while [Ayali et al., 2015] review the current knowledge on the role of sensory feedback in cockroach locomotion.

Among the load-sensitive mechanoreceptors of insects, campaniform sensilla are the best understood. They were identified as strain sensors by [Pringle, 1938] almost 90 years ago, whereas [Zill et al., 2004] summarize much of today's functional understanding. Located in different parts of the leg, i. e. the trochanter and the tibia, the campaniform sensilla are directionally sensitive to compressive stress. Therefore, they can be compared to strain gauges commonly used in technical force sensors. Campaniform sensilla appear in groups that differ in orientation, while they are arranged in parallel within each group. Thus, individual groups are sensitive to specific forces, while the joint groups enable the detection of complex loads experienced by a leg. In addition, the functional redundancy within each group increases robustness with respect to failure of single receptors. Using finite elements analysis (FEA), Kaliyamoorthy et al. investigate the potential role of the four groups of campaniform sensilla located at the trochanter of cockroach legs [Kaliyamoorthy et al., 2005]. Based on their analysis, the authors postulate four main functions: First, the campaniform sensilla detect the magnitude and the rate of change of leg forces that result from body weight and inertia; Second, the signals of the campaniform sensilla encode the direction of the ground contact forces experienced during stance; Third, the campaniform sensilla help to identify slippage by detecting rapid force decreases; And fourth, they provide muscle activating feedback during support and propulsion.

Similar to insects, such distributed load feedback has the potential to improve rough terrain locomotion of hexapedal robots as well. However, only few systems employ

multiple load sensors within each leg. Some examples are Hannibal [Ferrell, 1993] and Robot II [Espenschied, 1994] that use strain-gauge-based load signals from various locations of their legs, as well as the DLR Crawler [Görner et al., 2008] and the hexapod HITCR-II [Zhao et al., 2012] that both utilize joint torque sensors. In most other cases, six-legged walking robots rely on motor-current-based joint torque estimates or some kind of foot force measurements. The simplest solution are contact switches within the feet that detect the swing-to-stance transition of a leg as used by the hexapods OSCAR [El Sayed Auf et al., 2008] and TARRY II [Guddat, 2002]. The more complex solutions either return single-axis foot loads, like the force-sensitive resistors of Bill-Ant [Lewinger, 2005], or a complete contact force vector, such as the foot force sensors of Katharina [Ihme, 2002] and Lauron IV [Kerscher et al., 2008]. In both cases, these sensors enable the use of force distribution algorithms as well as the implementation of ground contact enforcing reflexes. However, one problem associated to foot force sensors often remains unsolved: Due their distal location, these sensors have to sustain high impact loads while they should provide high-resolution measurements at the same time. Thus, their design is either sturdy resulting in poor resolution or the resolution is good but the sensors are prone to damage. Therefore, the use of joint torque sensors or other distributed load feedback could increase the robustness and accuracy of load sensing while providing improved spatial resolution during collisions. Furthermore, using distributed load feedback, reflexes could be triggered more effectively throughout each phase of a stride.

With respect to the role of leg compliance, Blickhan and collaborators demonstrate that the spring-loaded inverted pendulum model serves as a template for running and hopping in animals [Blickhan, 1989], and that it is applicable to a large variety of species [Blickhan and Full, 1993, Full and Tu, 1990]. Regarding the dynamics of compliant hexapedal runners, an extensive review is given in [Holmes et al., 2006]. Apart from these fundamental results on the role of leg compliance in hexapedal running, it is often helpful in standing or walking that joints simply “give way”. By this, internal forces are reduced that result from parameter uncertainties in closed kinematic chains. As these forces require motor activity but do not provide any support or propulsion, their reduction improves the energy efficiency of the robot and decreases the risk of slippage. In order to realise such leg compliance, either passive elastic elements or some kind of active spring-like control could be used. Robot II, for example, employs springs for passive adjustments and spring-like proportional control for generating active compliance. Another example is the joint-torque-measurement-based compliance control algorithm of the DLR Crawler, which is introduced in greater detail in Chapter 3.

2.4 Pose Estimation in Legged Robots

Not only for control but also for other tasks like pose estimation, proprioceptive data is of great importance. In wheeled robotics, for example, it is common practice to calculate partial pose estimates based on wheel encoder readings and steering angles. Usually, such a wheel odometry algorithm returns a planar position as well as the heading angle of the vehicle. Only very few wheeled robots allow the calculation of an additional vertical motion estimate that is based on their kinematics. One of the few examples is the Shrimp robot [Lamon and Siegwart, 2004] developed at EPFL. On this robot an advanced bogie concept provides the necessary information for estimating the vertical motion. Nevertheless, pitch and roll angles, like on other wheeled robots, have to be determined by use of an inertial measurement unit (IMU).

In contrast to their wheeled counterparts, legged robots usually provide enough proprioceptive data from sensors embedded within their legs to calculate a complete 6-DOF pose estimate. However, due to their mechanical complexity, the high number of DOF, and the high variety and variability of gaits, the problem is much harder. Only very few tested examples of leg odometries exist that return a full 6-DOF pose estimate. Each of those additionally relies on IMU data to either stabilise the results or to compensate for a missing DOF of the pose. The detailed work on the robot RHex [Lin et al., 2005, Lin et al., 2006] is one of the few examples presented in literature. The robot consists of six equal, passively compliant, single degree of freedom legs and uses its hip joint encoder readings and leg deformation measurements to estimate its pose. Due to its kinematic configuration no yaw angle can be calculated by the basic odometry. Therefore, the data needs to be fused with IMU readings to return a full 6-DOF pose estimate. However, a great advantage of this approach is that it also covers the flight phases occurring during dynamic running.

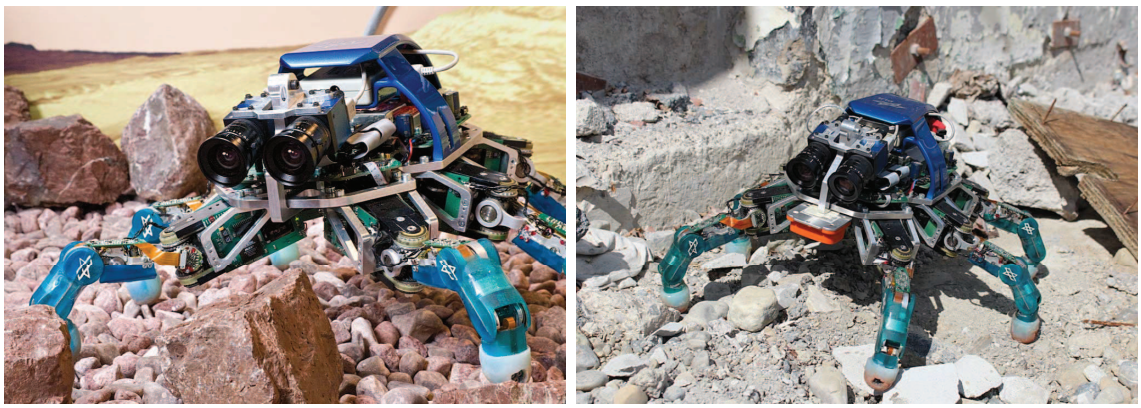
Another example for leg odometry is an algorithm developed for the hexapod Ambler [Roston and Krotkov, 1992] that was also implemented on the robot Lauron IV [Gassmann et al., 2005b, Gassmann et al., 2005a]. In this approach, the supporting legs are used to determine a rigid body transformation for the robot with respect to the world frame. The algorithm assumes an ideal no slip ground contact and finds a minimum error transformation that maps the positions of the supporting feet at the current time step with respect to the body frame onto the stored positions of the supporting feet with respect to the world frame. After finding the minimizing transformation, the positions of the supporting feet in world coordinates are recalculated and updated if they changed. This is necessary after a step but should not happen for legs in support according to the ideal no-slip condition. In order to reduce the

disturbing effect of slipping legs, individual leg weights are introduced that influence the transformation calculation. However, for Lauron as well as for Ambler the leg odometry experiences problems with drifting pitch angle and height estimates. To improve the results, the odometry of Ambler discards the tilt angles and replaces them by inclinometer readings. In case of Lauron the odometry estimates are fused with IMU- and magnetic-compass-based orientation data. While there is some performance data available for Ambler, there is no detailed data published for Lauron.

Recent examples for using leg odometry with dynamic quadrupeds are presented in [Reinstein and Hoffmann, 2011, Reinstein and Hoffmann, 2013] and [Ma et al., 2012]. In the first case, Reinstein et al. obtain a full pose estimate by fusing a velocity estimate based on leg odometry with data of an inertial navigation system using an extended Kalman filter. In the second case, Ma et al. publish an approach to improve the navigation robustness of the robot BigDog and its successor project LS3 by multi-sensor data fusion using leg odometry.

3 | The DLR Crawler

The DLR Crawler [Görner et al., 2008] is a torque-controlled, actively compliant, six-legged walking robot that was built during a diploma thesis project [Görner, 2007] to serve as a laboratory testbed for the development of gait and navigation algorithms. It is based on the fingers of DLR Hand II [Butterfass et al., 2001], which uniquely combine a comprehensive set of sensors with high performance actuation. Beginning with an overview of the robotic hardware, the chapter proceeds with a short description of the basic joint control algorithm, followed by the presentation of the leg kinematics and a simplified dynamics model of the robot.



(a) Indoor gravel testbed (b) Outdoor rubble environment

Figure 3.1: The DLR Crawler within different environments

3.1 Hardware Overview

Regarding its mechanical structure, the robot consists of six identical legs and is symmetric to its sagittal plane. It has a footprint of approximately 350 × 380 mm, stands on average 90 mm high, and has a mass of 3.5 kg. By design, all proximal leg joints are placed within a common plane, whereas their specific positions and orientations have been determined based on a careful analysis of the leg workspace as well as an

optimisation over a nominal tripod gait cycle. This optimisation aimed to minimize the peak loads of each leg and to maximize the stability margin of the support polygon while using large sections of the workspace. The outcome is the following: First, the middle legs have a larger base distance than the front and the hind legs; Second, with respect to the common plane spanned by all proximal joints, each leg attachment is tilted 15° downwards, leading to a better utilisation of the leg workspace shown in Fig. 3.2; Finally, the projection of the attachment of the front and hind legs into the common plane forms an angle of 50° with the sagittal plane.

Resulting from the initial design as a robotic finger, each leg has a length of 155 mm in full extension, four joints, and three DOF. The first two DOF are realised within the proximal joint using a differential bevel gear mechanism, which enables the additive use of motor torques about a single joint axis. The third degree of freedom is realised by a one to one coupling of the medial and distal joints of a leg. Each of the drive units consists of a permanent magnet synchronous motor followed by harmonic drive gears and a tooth belt transmission stage. The associated parameters, such as link lengths, joint motion ranges, and transmission ratios, are given in Table 3.1. Furthermore, each leg hosts a variety of proprioceptive sensors. These are Hall-effect sensors for relative motor angle measurement and commutation, link-side potentiometers for absolute joint angle measurement, strain-gauge-based link-side joint torque sensors as well as a 6-DOF force-torque sensor contained within the foot. In addition to the sensors embedded within the legs, an IMU provides acceleration and attitude data of the body, while a stereo camera enables visual odometry, obstacle avoidance, and terrain assessment and mapping.

In Fig. 3.3 an overview of the system setup is given. All locomotion-related computation and control is done off-board using a QNX-based real-time PC, while a Linux system is used for all vision-related computations as well as for path planning and navigation. A 1 kHz control loop is guaranteed by connecting the robot and the real-time PC using a fast IEEE-1355-based hierarchical serial communication link that transmits all sensor data to the PC and returns appropriate motor commands. Power is provided by an external 24 V supply and is transformed to various voltage levels on-board. The decision for the external computation and power supply has two reasons: First, it is to some extend a heritage of the DLR Hand II system architecture; But second and more importantly, it was chosen to eliminate any restrictions by limited computational power of on-board hardware. This allows testing of various control algorithms with different computational complexity while not requiring an efficient implementation at this stage of the development.

Table 3.1: Technical data of the DLR Crawler legs

Link lengths	
Proximal link	75 mm
Medial link	40 mm
Distal link	40 mm
Joint motion ranges	
Protraction/retraction proximal joint	±37 deg
Elevation/depression proximal joint	−55/+75 deg
Extension/flexion medial and distal joint	−20/+105 deg
Transmission ratios	
Harmonic drives	100:1
Tooth belt proximal joint	1.2:1
Tooth belt medial joint	2:1
Maximum joint velocity	>360 deg/s
Total leg mass	375 g

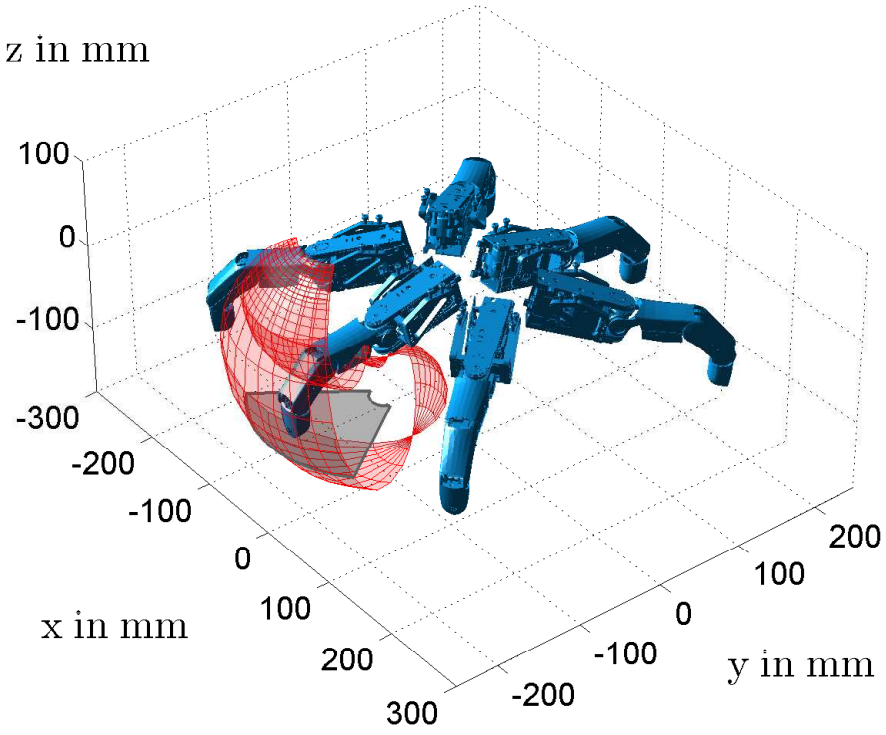


Figure 3.2: DLR Crawler and the workspace of its right middle leg: the grey patch indicates the useful region at a walking height of 80 mm

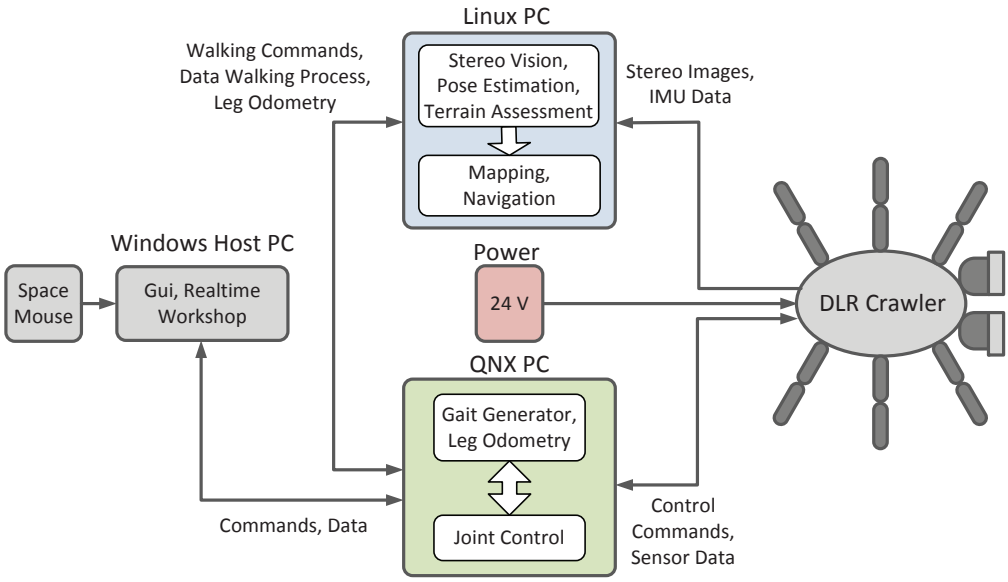


Figure 3.3: The DLR Crawler system setup

3.2 Joint Compliance Control

Since the robot is equipped with joint torque sensors as well as force-torque sensors within its feet, it enables the use of various torque and force control methods. However, following the assumption of uncertain or incomplete knowledge of the system parameters and states, control methods that require a detailed dynamics model of the robot are not considered within this thesis. Thus, the decision is to use simple joint compliance control. This is basically a PD control law with an underlying torque control loop, emulating a spring-damper system within each joint. The general advantage of such controller is that it enables on-line adaptation of the joint stiffness without the need of a complex mathematical model.

The joint compliance controller of each leg is implemented based on the well-known compliance control law,

$$\boldsymbol{\tau}_{j,d} = -\nabla V(\boldsymbol{\theta}_j) - \mathbf{D}(\boldsymbol{\theta}_j)\dot{\boldsymbol{\theta}}_j \tag{3.1}$$

$$= -\left(\frac{\partial V(\boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j}\right)^T - \mathbf{D}(\boldsymbol{\theta}_j)\dot{\boldsymbol{\theta}}_j, \tag{3.2}$$

$$V = \frac{1}{2}(\boldsymbol{\theta}_{j,d} - \boldsymbol{\theta}_j)^T \mathbf{K}_\theta (\boldsymbol{\theta}_{j,d} - \boldsymbol{\theta}_j). \tag{3.3}$$

Within those equations, $\boldsymbol{\tau}_{j,d} \in \mathbb{R}^3$ is the vector of desired joint torques, $\boldsymbol{\theta}_{j,d} \in \mathbb{R}^3$ the vector of desired joint angles, and $\boldsymbol{\theta}_j \in \mathbb{R}^3$ the vector of measured joint angles. The scalar $V(\boldsymbol{\theta}_j)$ is a potential function which collects all spring potentials and the matrix $\mathbf{K}_\theta$ is the related positive definite stiffness matrix. Furthermore, the matrix $\mathbf{D}(\boldsymbol{\theta}_j)$ is a positive semi-definite damping matrix. To account for the differential bevel gear mechanism of the base joint, a simple kinematic transformation matrix transforms the desired and measured joint torques to the respective motor-side torques $\boldsymbol{\tau}_{m,d}$ and $\boldsymbol{\tau}_m$. Since this transformation is only relevant for the first two DOF, the last row and column of the matrix realise an identity mapping for the third degree of freedom. Based on the motor-side torques, an underlying torque control loop, (3.4), is implemented that consists of a simple proportional controller with a gain matrix $\mathbf{K}_\tau$ and a friction compensation term $\hat{\boldsymbol{\tau}}_{m,\text{fric}}$. The latter is provided by a friction observer estimating the static and viscous friction torques. These mainly originate from the harmonic drive gears but also include velocity-related back electromotive force (EMF) effects.

$$\mathbf{u}_m = \mathbf{K}_\tau(\boldsymbol{\tau}_{m,d} - \boldsymbol{\tau}_m) + \hat{\boldsymbol{\tau}}_{m,\text{fric}} \tag{3.4}$$

While the above controller depicts the ideal case, the legs comprise some structural characteristics that have to be considered. For all joints, the controller assumes an axis of rotation that is identical to the axis of joint torque measurement. However, due to engineering reasons this could not be realised for the base joint. In this case, the 2D joint torque sensor is a bending beam structure placed 2.8 cm in a distal direction with respect to the joint axes. Therefore, the measured torques are smaller than the actual joint torques by an amount that is equal to the product of the distance in between sensor and joint axes and the shear force within the leg structure at the location of the sensor. Depending on the configuration of the leg and the application point of an external force, this difference is not negligible. To give some examples, the following two cases are considered. First, a force of 10 N is applied at the foot of the fully extended leg into a direction that is normal to the leg axis and normal to the axis of the second joint (elevation/depression). This causes an actual joint torque of 1.55 Nm at a distance of 15.5 cm from the force application point. However, the joint torque sensor located at a distance of 12.7 cm from the force application point only measures 1.27 Nm. Thus, the related error is about 18 % of the actual joint torque. If the leg is configured such that the force of 10 N is applied at a distance of 7.5 cm to the base joint, the error already grows to 37 %. Measuring or estimating the application point of an external force, the measurements could be corrected by a simple kinematic ratio. Assuming a fixed contact point at the tip of the foot, such a correction is applied for pose estimation presented in Chapter 6. However, this is not sufficient for

walking in general since multiple contacts might occur along the leg that cannot be resolved properly. Therefore, neglecting this source of error and applying the above controller without any correction of the measurement is an option. But what are the consequences? During the swing phase, the shear forces within the leg structure, resulting from gravity and inertial effects, are small. For this reason the controller will track the swing reference trajectories well. During the stance phase, the main loads on a leg originate in the body weight of the robot, and thus cause larger shear forces within its structure. In this case, the uncorrected joint torque measurements result in an actual joint stiffness that is larger than the desired value and grows with decreasing distance in between the force application point and the joint axis. In general, this is not a problem for walking. If the robot walks forward, the application point of the ground contact force roughly remains at a similar distance to the base joint. Therefore, the actual joint stiffness does not vary significantly. If the robot walks sideways, the force application point at the foot moves towards the base joint for all legs on one side of the robot, while it moves away for all legs on the other side. Thus, for a constant joint stiffness command, the base joint stiffens if the leg moves towards the body, while it softens for moving away. Fortunately, this behaviour is beneficial, since a leg that is closer to the body has to bear more weight.

In summary, the distance in between the joint torque sensors of the base joint and the joint axes has non-negligible effects that fortunately do not negatively influence the walking performance. Nevertheless, for using Cartesian force control algorithms, the measurements have to be corrected. This requires assumptions about the force application point or an additional hardware effort. For example, an improved sensor that determines the 2D shear forces within the intersection of the bending beam could easily solve the problem.

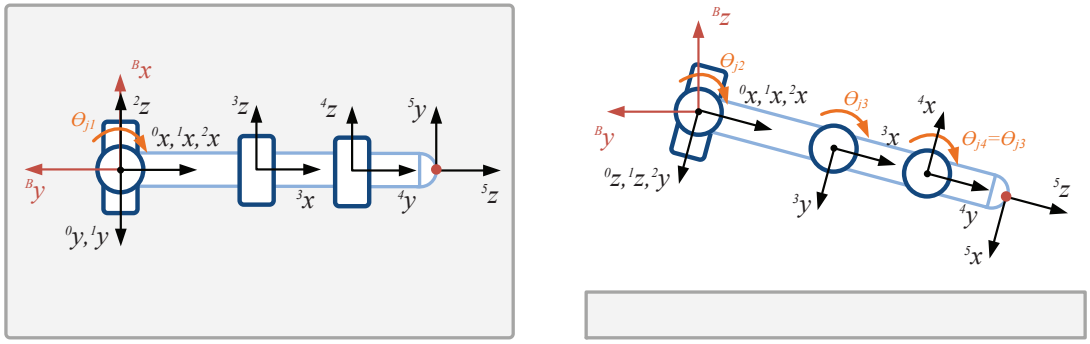
3.3 Leg Forward Kinematics

This section introduces the leg forward kinematics ${}^{B}\mathbf{x}_f(\boldsymbol{\theta}_j)$ of the DLR Crawler, which is part of the simplified dynamics model presented in Section 3.5 as well as the leg odometry algorithm developed in Chapter 6. Using transformation matrices ${}^{i-1}_{i}\mathbf{T}$, the Denavit-Hartenberg (DH) parameters given in Table 3.2, and the matrix product given by (3.6), the Cartesian position of the foot ${}^{B}\mathbf{x}_f$ is computed with respect to the leg base frame “B”. Each transformation matrix, ${}^{i-1}_{i}\mathbf{T}$, collects four successive single degree of freedom transformations that follow a fixed sequence according to [Craig, 2005] and uniquely define the transformation from link $i-1$ to link i. For each set of DH parameters, this sequence is a rotation of ϕ_{i-1} about the axis x_{i-1}, a fixed translation

Table 3.2: DH-parameters of the DLR Crawler legs

Transformation	ϕ_{i-1} in rad	a_{i-1} in mm	$\theta_i = (\theta_{j_i} + \theta_{\text{off}_i})$ in rad	d_i in mm
$\mathbf{B \rightarrow 0}$	$(\frac{15}{180} + 1)\pi$	0	$\frac{\pi}{2}$	0
$\mathbf{0 \rightarrow 1}$	0	0	θ_{j_1}	0
$\mathbf{1 \rightarrow 2}$	$\frac{\pi}{2}$	0	θ_{j_2}	0
$\mathbf{2 \rightarrow 3}$	0	75	θ_{j_3}	0
$\mathbf{3 \rightarrow 4}$	0	40	$\theta_{j_4} - \frac{\pi}{2}$	0
$\mathbf{4 \rightarrow 5}$	$-\frac{\pi}{2}$	0	π	40

of a_{i-1} along the axis x_{i-1}, a rotation of θ_i about the axis z_i, wherein θ_i is the joint angle plus some offset, and a fixed translation of d_{i-1} along the axis z_i.



(a) Top view (b) Side view

Figure 3.4: Coordinate systems and joint definitions used by the forward kinematics of the DLR Crawler legs (sketched for zero joint angle deflection)

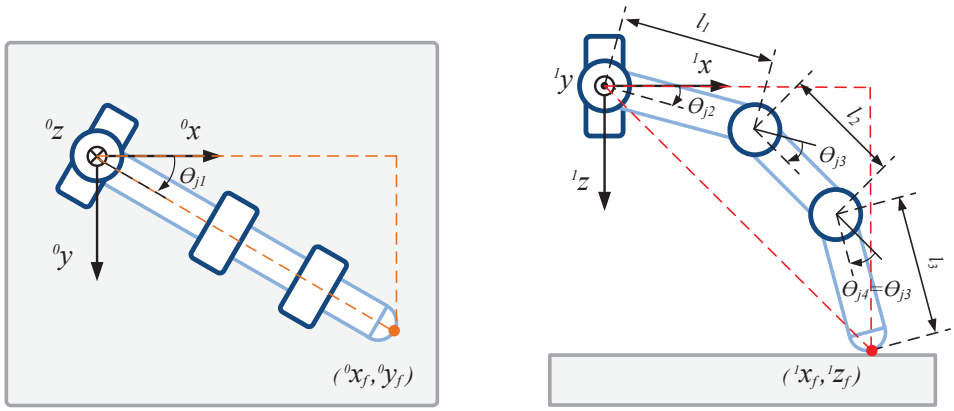
$$
{}^{i-1}_{i}\mathbf{T} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \cos\phi_{i-1}\sin\theta_i & \cos\phi_{i-1}\cos\theta_i & -\sin\phi_i & -d_i\sin\phi_{i-1} \\ \sin\phi_{i-1}\sin\theta_i & \sin\phi_{i-1}\cos\theta_i & \cos\phi_{i-1} & d_i\cos\phi_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}
$$

$$\begin{pmatrix} {}^{B}\mathbf{x}_f(\boldsymbol{\theta}_j) \\ 1 \end{pmatrix} = {}^{B}_{0}\mathbf{T}^{0}_{1}\mathbf{T}^{1}_{2}\mathbf{T}^{2}_{3}\mathbf{T}^{3}_{4}\mathbf{T}^{4}_{5}\mathbf{T} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.6}$$

Taking the time derivative of ${}^{B}\mathbf{x}_f(\boldsymbol{\theta}_j)$ yields the Cartesian velocity of the foot as well as the Jacobian matrix of the leg $\mathbf{J}_l$.

$${}^{B}\dot{\mathbf{x}}_f = \frac{\mathrm{d}}{\mathrm{dt}}\,{}^{B}\mathbf{x}_f(\boldsymbol{\theta}_j) = \frac{\partial\,{}^{B}\mathbf{x}_f(\boldsymbol{\theta}_j)}{\partial\boldsymbol{\theta}_j}\dot{\boldsymbol{\theta}}_j = \mathbf{J}_l\dot{\boldsymbol{\theta}}_j \tag{3.7}$$

3.4 Leg Inverse Kinematics



(a) Top view (b) Side view

Figure 3.5: Parameters and geometric relations used to compute the inverse kinematics of the DLR Crawler legs

As the gait coordination algorithm of the robot presented in Chapter 4 works on the Cartesian positions of the feet, an inverse kinematics algorithm has to be employed in order to compute the desired joint angles required by the joint compliance controller. For this purpose, an analytical approach is chosen, since the related equations can be efficiently solved for 3-DOF legs, even if these include an additional, passively coupled joint. After transforming the foot position from the body coordinate system into the leg-based coordinate system 0 shown in Fig. 3.5(a), the joint angle involved in the

protraction and retraction of the leg, θ_{j1}, is computed. Employing the inverse tangent function with two arguments, atan2, this angle is determined including the proper sign.

$$\theta_{j1} = \text{atan2}\left(\frac{{}^0y_f}{{}^0x_f}\right) \tag{3.8}$$

Using angle θ_{j1}, the position of the foot is transformed into the leg-based coordinate system 1 shown in Fig. 3.5(b) in order to compute the remaining two angles, θ_{j2} and θ_{j3}. These angles are involved in the elevation and depression of the leg as well as its extension and flexion. According to the Pythagorean theorem the following equation is obtained.

$$\begin{aligned} 0 \;=\; & \left(l_1\cos\theta_{j2} + l_2\cos\left(\theta_{j2}+\theta_{j3}\right) + l_3\cos\left(\theta_{j2}+2\theta_{j3}\right)\right)^2 \\ & + \left(l_1\sin\theta_{j2} + l_2\sin\left(\theta_{j2}+\theta_{j3}\right) + l_3\sin\left(\theta_{j2}+2\theta_{j3}\right)\right)^2 \\ & - \left({}^1x_f{}^2 + {}^1z_f{}^2\right) \end{aligned} \tag{3.9}$$

Simplifying this equation and rearranging terms, θ_{j2} is eliminated. The resulting quadratic equation is then easily solved for θ_{j3}.

$$0 = \cos^2\theta_{j3} + \frac{l_2\left(l_1+l_3\right)}{2l_1l_3}\cos\theta_{j3} + \frac{l_1^2+l_2^2+l_3^2-2l_1l_3-{}^1x_f{}^2-{}^1z_f{}^2}{4l_1l_3} \tag{3.10}$$

$$\theta_{j3} = \arccos\left(-\frac{l_2\left(l_1+l_3\right)}{4l_1l_3} + \sqrt{\left(\frac{l_2(l_1+l_3)}{4l_1l_3}\right)^2 - \frac{l_1^2+l_2^2+l_3^2-2l_1l_3-{}^1x_f{}^2-{}^1z_f{}^2}{4l_1l_3}}\right) \tag{3.11}$$

Finally, inserting θ_{j3} into the following equation yields the second angle of the base joint θ_{j2}.

$$\theta_{j2} = \arctan\left(\frac{{}^1z_f}{{}^1x_f}\right) - \arctan\left(\frac{l_2\sin\theta_{j3}+l_3\sin 2\theta_{j3}}{l_1+l_2\cos\theta_{j3}+l_3\cos 2\theta_{j3}}\right) \tag{3.12}$$

3.5 A Simplified Dynamics Model

This section introduces a simplified dynamics model of the DLR Crawler, which includes a body with distributed inertia but assumes massless legs. This assumption is justified by the negligible inertial forces that the legs of the robot exert on its body

during walking. The ground reaction forces are computed based on kinematic relations of the legs as well as joint torques that result from joint spring compressions. In addition, friction forces in between the tip of a slipping leg and the ground are approximated by a first-order model. The reason for this choice over a full multi-body model is simply faster computation. With respect to multi-body simulations there are several factors that slow down the simulation speed. First of all, the small masses and mass moments of inertia of the leg segments paired with high joint stiffness render multi-body simulations numerically "stiff", and thus slow. Next, the coupling of the medial and distal joint within the legs of the DLR Crawler adds constraints that cannot be resolved properly by many software packages. Furthermore, consistently integrating the dynamic equations for all closed kinematic chains formed by the legs in ground contact increases the computation time, too. In order to verify the validity of such a simplified model, it has been successfully tested against a multi-body simulation of the DLR Crawler using the commercial software package Simpack.

The simplified model of the DLR Crawler employs the parameters given in Appendix A and is set up as follows. In order to model the dynamics of the robot body, a set of first-order differential equations is employed. Given in vector notation, the state variables are the linear momentum of the body $\mathbf{p} \in \mathbb{R}^3$, the angular momentum of the body $\mathbf{L} \in \mathbb{R}^3$, the centre of mass (COM) position $\mathbf{x} \in \mathbb{R}^3$, and the unit quaternion $\mathbf{q} = (q_1, q_2, q_3, q_4)$ that represents the orientation of the body with respect to the inertial frame. The quaternion representation has been chosen over others due to not having the gimbal lock problem associated to Euler angles and only experiencing small numerical drifts during integration in comparison to rotation matrices. Given by the following equations, the time derivatives of the linear momentum as well as the angular momentum are simply computed by summing all externally acting forces and torques, respectively.

$$\dot{\mathbf{p}} = \sum_{i=1}^{6} \mathbf{f}_{r,i} + \mathbf{f}_g \tag{3.13}$$

$$\dot{\mathbf{L}} = \sum_{i=1}^{6} \boldsymbol{\tau}_{r,i} \tag{3.14}$$

$$\tag{3.15}$$

With respect to the inertial frame, $\mathbf{f}_{r,i} \in \mathbb{R}^3,\ i = 1, ..., 6$ and $\boldsymbol{\tau}_{r,i} \in \mathbb{R}^3,\ i = 1, ..., 6$ are vectors that represent the reaction forces and reaction torques applied to the body by leg i. For a leg in its stance phase, they result from the ground contact forces, while they are zero for a leg in its swing phase. Those ground-contact-based reaction forces

and reaction torques as well as the gravitational force $\mathbf{f}_g$ are the only ones that act on the body during simulation. Thus, they are responsible for all of its position and attitude changes.

Employing the simple 3×3 mass matrix of the body, $\mathbf{M}$, and its angular velocity vector with respect to the inertial frame, $\boldsymbol{\omega} \in \mathbb{R}^3$, the time derivatives of the robot position as well as the unit quaternion are calculated using the following equations,

$$\dot{\mathbf{x}} = \mathbf{M}^{-1}\mathbf{p} \tag{3.16}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{pmatrix} -\omega_1 q_2 - \omega_2 q_3 - \omega_3 q_4 \\ \omega_1 q_1 + \omega_2 q_4 - \omega_3 q_3 \\ \omega_2 q_1 + \omega_3 q_2 - \omega_1 q_4 \\ \omega_3 q_1 + \omega_1 q_3 - \omega_2 q_2 \end{pmatrix} \tag{3.17}$$

Hereby, the time derivative $\dot{\mathbf{q}}$ is the result of a quaternion product in between the quaternion $[0, \boldsymbol{\omega}]$ and the quaternion $\mathbf{q}$, wherein the required angular velocity vector $\boldsymbol{\omega} \in \mathbb{R}^3$ is computed as follows. First, by using the unit quaternion $\mathbf{q}$ a rotation matrix $\mathbf{R}$ is calculated.

$$\mathbf{R} = \begin{bmatrix} 1-2(q_3^2+q_4^2) & 2(q_2q_3-q_1q_4) & 2(q_2q_4+q_1q_3) \\ 2(q_2q_3+q_1q_4) & 1-2(q_2^2+q_4^2) & 2(q_3q_4-q_1q_2) \\ 2(q_2q_4-q_1q_3) & 2(q_3q_4+q_1q_2) & 1-2(q_2^2+q_3^2) \end{bmatrix} \tag{3.18}$$

Next, the body-coordinate-system-referenced inertia tensor $\mathbf{I}_b$ is transformed to the inertial frame. Finally, the angular velocity vector is calculated based on the angular momentum $\mathbf{L}$.

$$\mathbf{I} = \mathbf{R}\mathbf{I}_b\mathbf{R}^T \tag{3.19}$$

$$\boldsymbol{\omega} = \mathbf{I}^{-1}\mathbf{L} \tag{3.20}$$

In the following, the calculation of the reaction forces and torques applied to the body is presented in more detail. Knowing the positions of the hip and the foot contact point of leg i, a vector can be calculated that represents the extension of the respective leg. Applying the inverse kinematics algorithm introduced within the previous section, the actual joint angles can be calculated. Given a desired leg trajectory, the desired joint angles are know. Now, with knowledge of the actual and the desired joint angles as well as their derivatives, the joint compliance control law introduced in Section 3.2 is applied and returns the desired joint torques. Due to the assumption of massless legs

without joint friction, the desired torques are equivalent to the applied joint torques. Employing the forward-kinematics-based leg Jacobian matrix $\mathbf{J}_{l,i}$, the vector of applied joint torques $\boldsymbol{\tau}_{l,i}$ can be transformed to a force ${}^{B}\mathbf{f}_{l,i}$ that is exerted by the foot onto the ground with respect to the leg base frame B.

$$\boldsymbol{\tau}_{l,i} = \mathbf{J}_{l,i}^{T}\,{}^{B}\mathbf{f}_{l,i} \tag{3.21}$$

$${}^{B}\mathbf{f}_{l,i} = (\mathbf{J}_{l,i}^{T})^{-1}\boldsymbol{\tau}_{l,i} \tag{3.22}$$

Defining a vector with the same magnitude as ${}^{B}\mathbf{f}_{l,i}$ but opposite direction gives the ground reaction force of the leg in frame B. Next, by multiplying this vector with a rotation matrix, it is transformed from the leg base coordinate system to the inertial frame. The resulting force vector is then the reaction force $\mathbf{f}_{r,i}$ that leg i exerts on the body. Taking the cross product of the inertial-frame-based distance vector, which is spanned in between the ground contact point of leg i and the body COM, and the related reaction force vector $\mathbf{f}_{r,i}$, the reaction torque on the body, $\boldsymbol{\tau}_{r,i}$, is obtained.

For all previous computations, proper transitions of the leg states are of great importance. Within its swing phase, a leg has no ground contact and, being massless, perfectly follows the desired foot trajectory. During this phase the touch-down condition (TD) for a transition to the stance phase is constantly monitored. With respect to the inertial frame, this condition is met once the vertical position of the leg tip is smaller or equal to the corresponding ground height. At touch-down of the foot, the leg state switches to stance and the coordinates of the contact point are stored for the computation of the reaction torques and forces. Being within the stance phase a lift-off condition (LO) is monitored for each leg. This condition is met once the normal component of the reaction force vector with respect to the ground surface is smaller than zero and shows a negative derivative. It ensures that a rigid ground does not pull on the leg, which is physically not feasible. At lift-off the leg state switches to swing, the reaction forces and torques of the respective leg are set to zero, and the leg tip instantaneously follows its desired trajectory. However, a shortcoming off the massless leg assumption becomes apparent at lift-off when small amounts of energy stored within the legs springs are immediately lost.

To allow for slipping legs, a simple first-order model [Saranli, 2000] approximates the sliding velocity of a foot once the ground reaction force vector leaves the friction cone. For this purpose, the leg force $\mathbf{f}_{l,i}$ represented within the inertial frame is separated into a normal and a tangential component, $\mathbf{f}_{n,i}$ and $\mathbf{f}_{t,i}$, with respect to the ground. Next, the static friction force is calculated from the normal component of the leg force

and the coefficient of friction μ. If the tangential leg force is larger than the friction force, a sliding-velocity vector is computed according to (3.23). This velocity vector is integrated and the resulting position increment is added to the coordinates of the ground contact point, emulating a slipping leg. The corresponding tangential fraction of the reaction force onto the body is reduced to the friction force. The sliding motion of the contact point continues until the tangential force component is smaller than the friction force, which resets the sliding velocity to zero.

$$\mathbf{v}_{\text{slip},i} = v_{\text{slip,nom}} \frac{\|\mathbf{f}_{t,i}\| - \|\mu \mathbf{f}_{n,i}\|}{\|\mathbf{f}_{t,i}\|^2} \mathbf{f}_{t,i} \tag{3.23}$$

In order to demonstrate the effectiveness of this simplified dynamics model, the following figures present the results of an exemplary simulation run. All parameters are set to the values given in Appendix A. The soft joint stiffness setting is chosen and the robot is commanded to walk forward at a desired velocity of $v_{x,\text{des}} = 40\,\text{mm/s}$. The set of algorithms, introduced within the next chapter, generates a tetrapod gait that properly propels the robot. The resulting smooth forward motion is shown in Fig. 3.6, which also displays the lateral and vertical oscillations of the body.

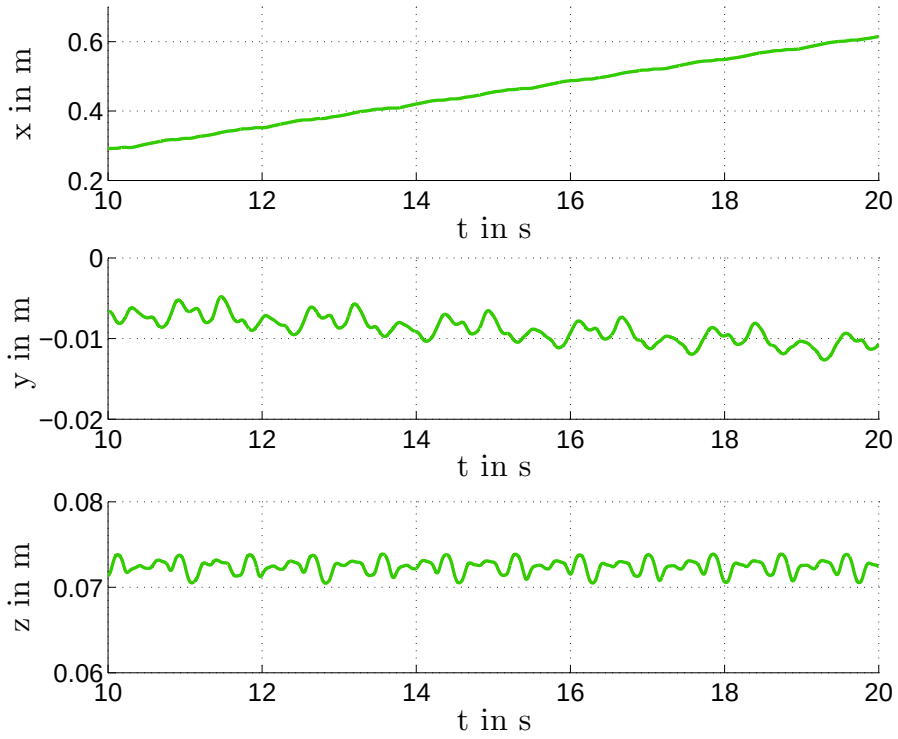


Figure 3.6: Centre of mass trajectories of the simulated DLR Crawler: soft joints, $v_{x,\text{des}} = 40\,\text{mm/s}$

In addition, Fig. 3.7 depicts the associated oscillations of the yaw, pitch, and roll angles. Thereof, the pitch angle shows the highest amplitude, whereas the other two angles oscillate at approximately twice the frequency. Finally, Fig. 3.8 displays the ground reaction forces of the legs, which properly share the vertical loads. However, similar to observations on running cockroaches [Full et al., 1991], the legs of the walking robot specialise with respect to their propulsive contribution. The front legs mainly serve as brakes, while the hind legs mostly propel the body. The middle legs do both as they change their behaviour from braking to propulsion at mid-stance. In addition, the y-components of the individual ground reaction forces reveal outward pushing legs. Again, similar observations were made on running cockroaches, even though the total ground reaction force characteristics do not match due to the different phase relations of the legs.

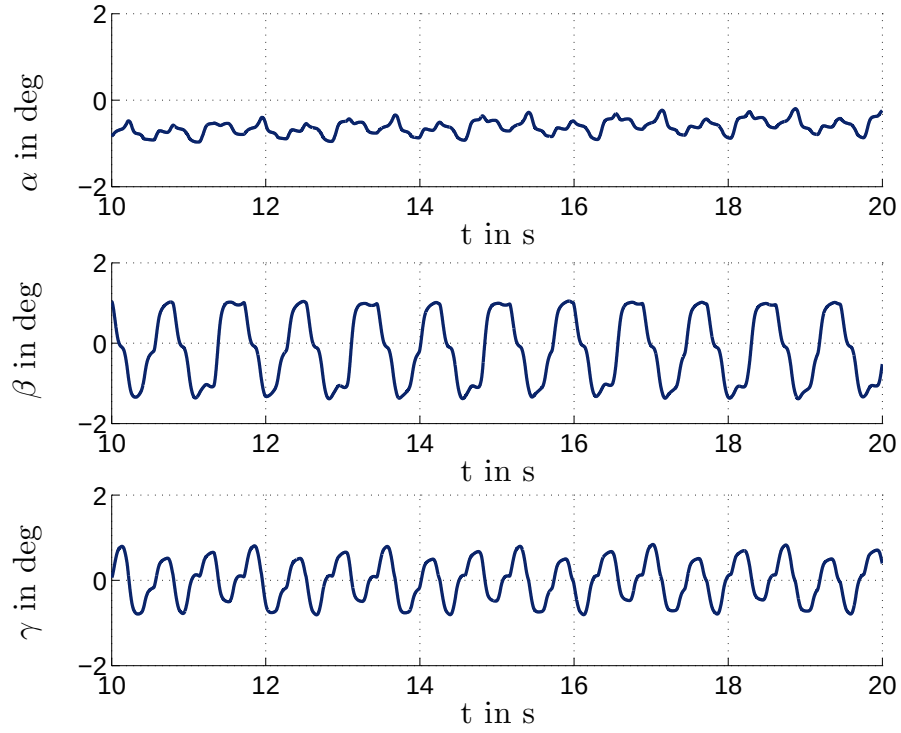
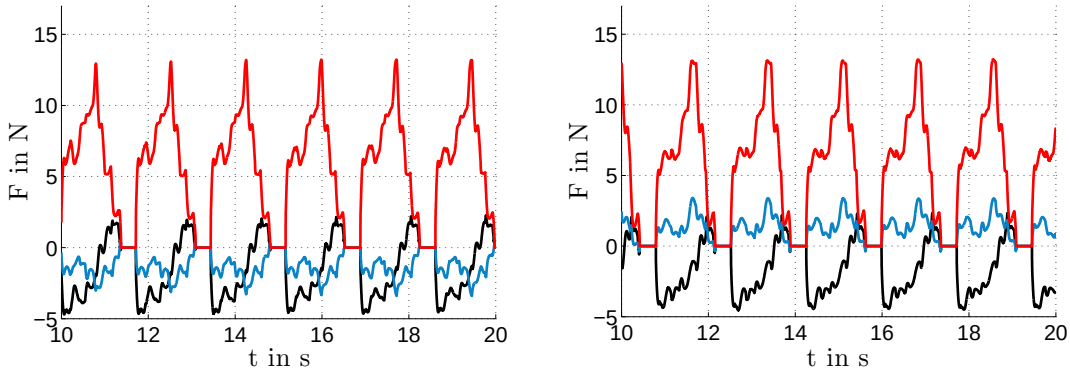
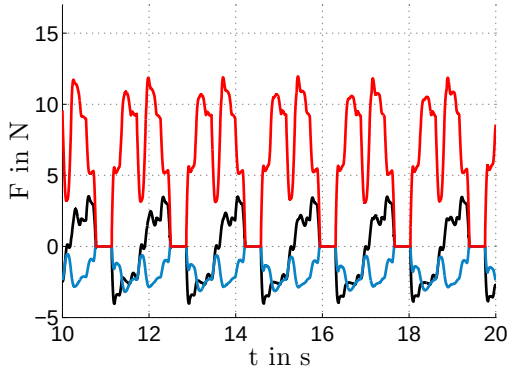


Figure 3.7: Yaw, pitch and roll angles of the simulated DLR Crawler: soft joints, $v_{x,\mathrm{des}} = 40\,\mathrm{mm/s}$

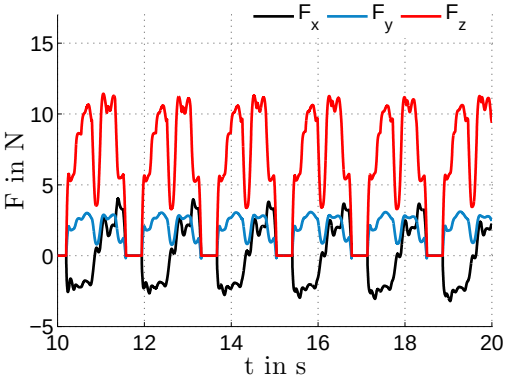


(a) Left front leg (L1)

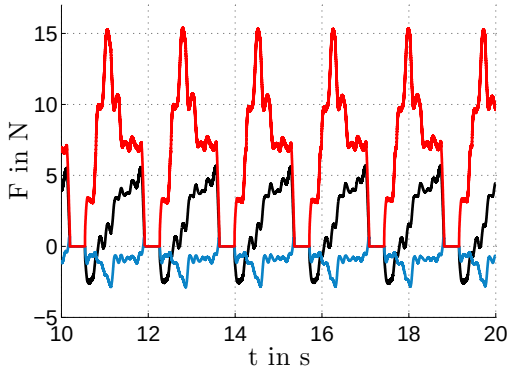
(b) Right front leg (R1)



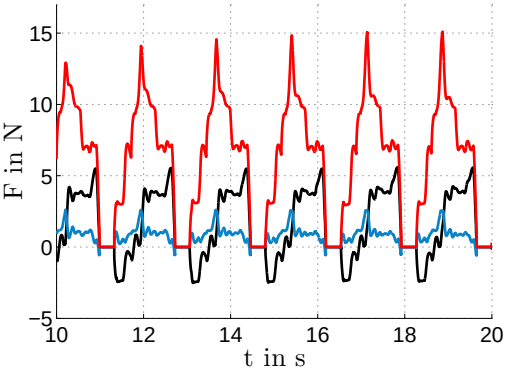
(c) Left middle leg (L2)



(d) Right middle leg (R2)



(e) Left hind leg (L3)



(f) Right hind leg (R3)

Figure 3.8: Ground reaction forces of the simulated DLR Crawler: soft joints, $v_{x,\mathrm{des}} = 40\,\mathrm{mm/s}$

4 Robust and Adaptive Gaits

4.1 Introduction

Autonomous, goal-directed walking across uneven and previously unknown natural terrain is a complex and challenging task for a six-legged robot. In addition to the process of walking itself, it requires the robot to collect data about its environment, to assess the terrain based on this information, and to plan a path towards the desired goal. In order to follow this path, the legs of the robot have to produce well-coordinated motions. For this purpose, each leg alternates in between two states. During stance it propels the body in a kinematically consistent way along the planned path, whereas it repositions the foot during swing by producing appropriate stepping motions. Due to the uneven terrain and varying motion commands, each leg is constantly forced to adapt its stance and swing cycle with respect to amplitude, timing, and direction. While doing so, it is crucial that the robot as a whole maintains a statically stable configuration at all times. Considering the fact that an exploration mission combines locomotion, terrain assessment, path planning, and other high level tasks, it quickly becomes a very complex problem. To reduce the related computational efforts and to enable the implementation on resource-limited on-board hardware, a hierarchical algorithmic structure based on strongly self-contained layers with different levels of abstraction appears to be the most promising option. Following such an approach of a layered structure, this chapter presents the development, implementation, and test of a set of algorithms that constitute the "walking layer" of a compliant six-legged robot. The main task of this walking layer is to generate robust and adaptive omnidirectional gaits that enable the robot to negotiate obstacles within the walking height autonomously. Hereby, it is intended to hide the complexity of generating coordinated, locally appropriate leg motions from other layers such as for example path planning and navigation. Clearly, for smooth operation, there must be an information exchange in between the layers, but each layer should act as a filter that only propagates relevant information at a useful level of abstraction. Thus, to connect with other layers, the walking layer provides a reduced set of commands like walking velocity, walking

direction, and turning rate, which allow for steering the robot. In addition, it returns useful information about the walking process itself as for example information about the pose of the robot, collisions with the environment, or actual loading conditions.

Based on the assumption of having uncertain information about the robot and its environment, all algorithms involved in the walking process target simplicity and follow a decentralised scheme that locally exploits proprioceptive sensor data provided by the legs. Thus, the presented approach favours to resolve complex situations by emerging behaviour instead of using complex models of the robot and its interaction with the local terrain. The chapter proceeds as follows. First, the gait-generating algorithms are presented. These consist of a decentralised flexible gait coordination based on well-known behavioural results from stick insect studies, a kinematics-based joint trajectory generation which sets the reference for the joint compliance controllers, and several simple leg reflexes which enable reactive obstacle crossing. Following these sections, experimental results are presented in order to validate the approach.

4.2 Coordination of Stepping Motions

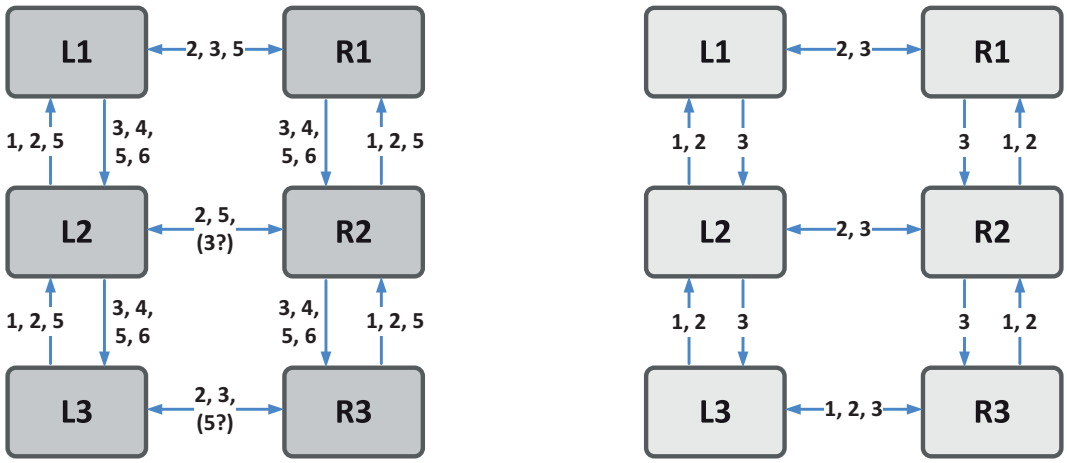
Within this work, the coordination of stepping motions is based on the well-known behavioural rules that Cruse and collaborators [Cruse, 1990, Cruse et al., 1998] identified for the stick insect. This approach is chosen over others due to its capacity to generate flexible stepping sequences while being simple and computationally inexpensive. In general, the following viewpoint is adopted. Each leg is considered as an abstract hybrid oscillator which cycles through swing and stance phases. These phases are roughly determined by a small set of centrally promoted variables, like walking velocity, walking direction, and turning rate, as well as some leg-specific variables, such as nominal step height and protraction velocity. During the stance phase each leg propels the body into the desired direction by moving in a kinematically consistent way. Upon reaching a kinematic threshold with respect to the body, i. e. the posterior extreme position (PEP), a leg changes from stance to swing and performs a return stroke to reposition its foot with respect to the walking direction at the anterior extreme position (AEP). While the AEP of each leg mostly remains at a fixed distance, the PEP is modulated by the behaviour of neighbouring legs. This modulation causes prolonged or shortened stance periods, resulting in flexibly coordinated stepping sequences.

Prior to describing the specific algorithmic implementation, this paragraph recalls Cruses's rules for gait coordination and their direction of influence as depicted by Fig. 4.1(a). The first mechanism acts in between ipsilateral legs and is directed towards the front of the animal. The rule states that, while performing a return stroke, the

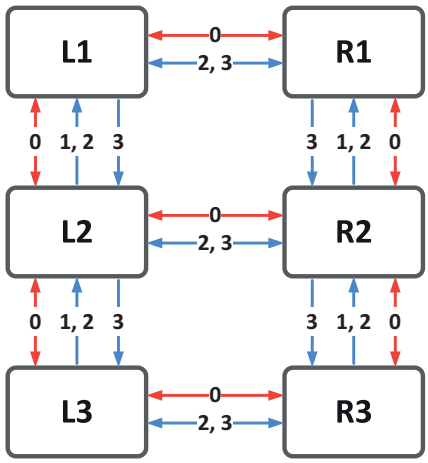
sending leg inhibits the onset of the return stroke of its anterior neighbour. The second mechanism acts in between ipsilateral and contralateral legs. With respect to the ipsilateral legs it is directed towards the anterior leg, while it is a bidirectional influence in between contralateral legs. By this mechanism the sending leg excites the return stroke of the receiving leg right after it has finished its own return stroke. The third mechanism also acts in both directions in between contralateral legs and is posteriorly directed in between ipsilateral legs. Using this mechanism the sending leg excites the return stroke of a receiving leg. This excitation grows stronger the closer the sending leg approaches its own PEP. Similar to mechanism 1, mechanism 4 acts in between ipsilateral legs but is directed from the front to the back of the animal. Hereby, the receiving leg targets the current foot position of the anterior leg as a potential next foothold. Mechanism 5 is about load-induced coupling of the legs, which on the one hand results in the co-activation of legs and on the other hand leads to prolonged stance phases to support larger loads. The last mechanism, called "treading on tarsus reflex", initiates a small correction step of the receiving leg if it steps onto the tarsus of the sending leg.

With respect to the temporal coordination of stepping motions, the first three mechanisms are the most important. They facilitate forward-directed waves of protractions along each side of the animal while contralateral legs are pushed towards a 180° phase shift. Considering the stability of a forward walking animal, those forward-directed waves of return strokes are clearly beneficial. Just having finished their own steps, the posterior legs better support the body, and thus provide additional time to the front legs for negotiating upcoming obstacles. In turn, the posterior legs will already benefit from the footholds found by the front legs. The fact that the first three coordinating rules are sufficient to produce a continuum of stable gaits for hexapedal robots is demonstrated by Espenschied using Robot II [Espenschied, 1994, Espenschied et al., 1996]. Within his approach all legs are connected by mechanisms 1, 2 and 3 as originally proposed by Cruse. In addition to those connections, Espenschied introduces a bidirectional coupling of both hind legs by mechanism 1, which is shown in Fig. 4.1(b). The introduction of this additional coupling is not explained in Espenschied's publications, but it most likely helps to establish the out-of-phase relation of contralateral legs.

The gait coordination developed within this thesis is based on the first three of Cruse's coordinating mechanisms as well. The major differences with respect to Espenschied's approach and to the original set of Cruse's rules are the following. At first, mechanism 2 is not only active for a short time after the footfall of a sending leg, but throughout its complete stance phase as introduced in [Ferrell, 1993]. This



(a) (b)



(c)

Figure 4.1: Coordination rules: a) Cruse's rules as presented in [Dürr et al., 2004] b) Espenschied's implementation [Espenschied et al., 1996] c) Implementation for the DLR Crawler

adjustment strongly supports forward-directed waves of protractions across the whole range of feasible walking velocities. The second difference is the use of an additional inhibitory mechanism. This simple mechanism is termed mechanism 0 and bidirectionally connects all neighbouring legs. Prior to initiating a return stroke, this new mechanism explicitly requires each leg to check the state of its neighbours. In case not all neighbours are within their stance phase, the leg requesting to step is not allowed to transition into the swing phase and subsequently has to prolong its own stance phase. In parallel to this work, El Sayed Auf employs a similar inhibitory rule in combination with a variable duty cycle to generate a continuum of coordinated gaits for the six-legged robot Oscar [El Sayed Auf et al., 2008, El Sayed Auf, 2010]. By this, he demonstrates that a single inhibitory rule applied to legs connected in a ring topology is sufficient to produce stable gaits. However, the combination of such a rule with Cruse's first three mechanisms grants much more influence on the emerging gait.

In order to investigate the influence of each mechanism on the characteristics of the emerging gait, Espenschied introduces lesions that suppress the connections of single or multiple mechanisms [Espenschied, 1994]. His observations are the following. Mechanism 2 promotes forward-directed waves of protractions, while mechanism 3 promotes 180° phase shifts in between contralateral neighbours. Furthermore, he states that using mechanism 3 alone enables locomotion over a limited range of speeds and that changing the weights of mechanism 3 has the most disruptive influence on the emergence of stable gaits. Additional results are presented by Calvitti who investigates the gait coordination by Cruse's rules with respect to hybrid system theory [Calvitti and Beer, 2000, Calvitti, 2004]. Central to his analysis is the question how the dynamics of small subnetworks influence the overall temporal coordination, and thus facilitate phenomena like phase locking. For this purpose, he analyses networks of two oscillators with either unidirectional or for some cases bidirectional coupling by mechanisms 1, 2 or 3. Those mechanisms are implemented using simple step or ramp functions as proposed by Espenschied. However, there are two differences in Calvitti's implementation. First, it only considers forward walking, and second, mechanism 2 is position- and not time-dependent. Regarding the unidirectional two oscillator network, Calvitti demonstrates that even those simple hybrid systems show rich dynamic behaviour which strongly depends on the parametrisation. He states that instead of asymptotically stable phase relations only neutral stability has been found and due to this the phase locking is rather a phase compression. One important result is that the individual mechanisms generate a funnel which draws a large range of initial conditions towards a narrow band of phase relations. Nevertheless, up to date an analysis of the full range of hexapedal coordination is still missing.

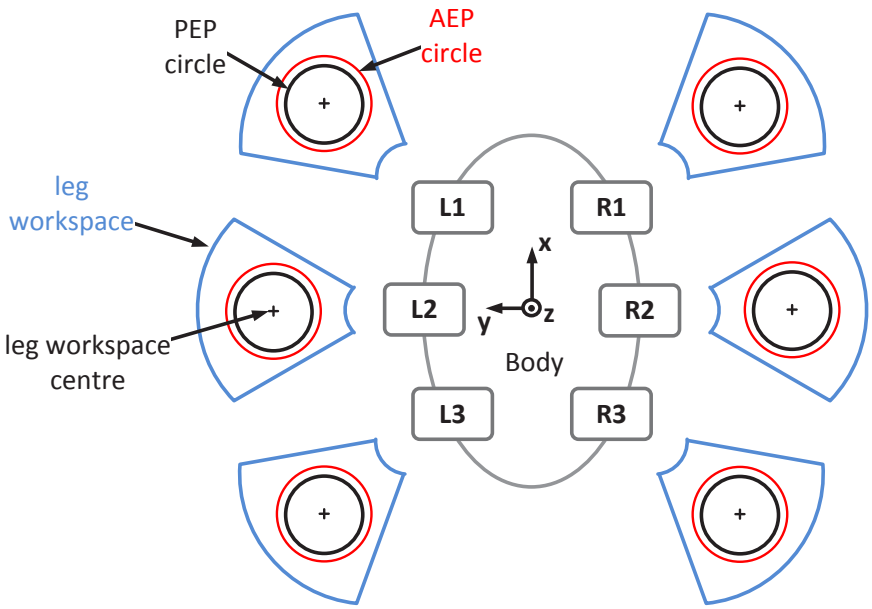


Figure 4.2: Schematic top view of the DLR Crawler indicating the leg labelling, the body-centred coordinate system, the leg workspaces as well as the AEP and PEP circles and their respective centres

As it is the clear goal to generate omnidirectional gaits for the DLR Crawler, Espenschied's definition of the AEP and PEP to be the radii of circles is adopted. In the following, all coordination-related kinematic measures are defined with respect to a coordinate frame that is centred within the robot body as shown in Fig. 4.2. The x-axis of this frame points towards the front of the robot, the y-axis towards its left side and the z-axis completes a right-hand system pointing upwards. The AEP and PEP circles of each leg are centred within a cross section of the corresponding leg workspace such that they all lie within a common plane that is in parallel to the horizontal plane of the body. Furthermore, by projecting the positions of the feet onto the common plane, this approach decouples the coordination from height variations that are required for posture adjustments or adaptation to the terrain. Considering a suitable range of heights, the resulting AEP and PEP cylinders represent the admissible volume for the motion of the stance feet. According to the coordination influences of neighbouring legs, the PEP circle and the corresponding cylinder radially shrink or extend, which results in varying stance-to-swing transitions and consequently leads to coordinated stepping motions. Being within its PEP cylinder, the foot of a stance leg freely moves such that it propels the body according to the desired walking velocity, walking direction, and turning rate. Once the projection of the foot onto the common

plane leaves its PEP circle, it requests to step. At this point mechanism 0 is activated and checks if all neighbouring legs are within their stance phase. If they are not, the leg prolongs its own stance phase and leaves its PEP cylinder in radial direction while waiting for its neighbour to change from swing to stance. In case the leg reaches a predefined kinematic boundary during this process, it stops and sends a stop signal to all other legs as well. At this point the neighbouring leg, which caused the delay, gets further time to finish its step. If somehow this is not possible, a higher-level algorithm has to resolve the situation by switching to a central coordination of leg motions. Under normal conditions Cruse's rules prevent simultaneous stepping of adjacent legs and leave enough time and space for adaptation. However, for abrupt changes of the walking command or due to the negotiation of obstacles, the return strokes of two neighbouring legs might overlap in time. In this case mechanism 0 guarantees that neighbouring legs never step simultaneously and subsequently destabilise the robot.

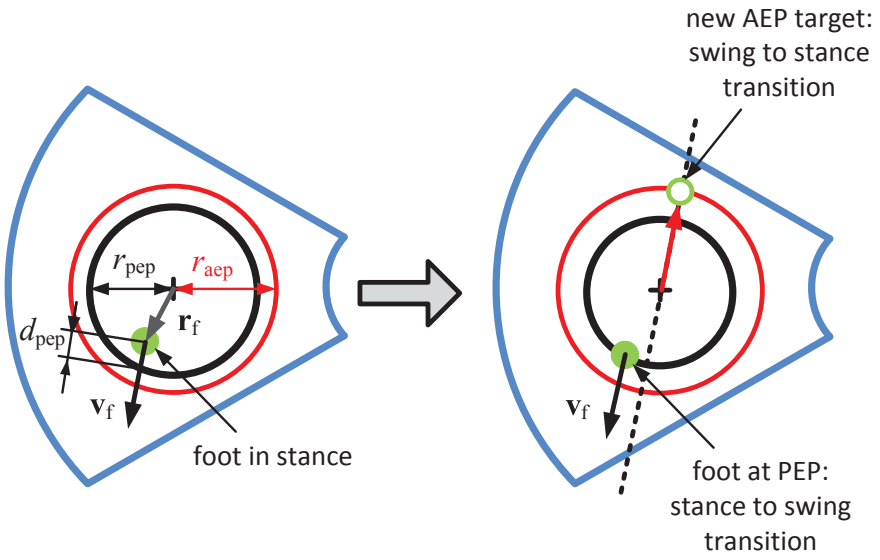


Figure 4.3: Schematic top view of the leg workspace and coordination / foot motion-related variables and parameters

Besides applying the coordination rules themselves, another important property of the algorithm is that each leg performs its return stroke at a high return stroke velocity. Following Graham's observations from stick insect studies [Graham, 1972], this velocity is equal for all legs and independent of the overall walking speed. In detail, Graham noticed that for adult animals the protraction time of a leg is independent of the step cycle time, and thus, the walking speed. Within the implementation for the DLR Crawler the return stroke velocity is defined with respect to the common plane that

contains the AEP and PEP circles. Herein, it is the velocity at which the projection of a foot onto this plane moves along a straight line from its final position on the PEP circle to a target position on the AEP circle. Thereby, the target point on the AEP circle is calculated at the onset of the return stroke as shown in Fig. 4.3. It is situated at the intersection of the AEP circle with a velocity-based vector that originates at its centre. This vector points into the direction opposite to the velocity vector that a stance foot, located at the centre of the AEP circle, would have based on the current walking commands. Furthermore, it is important that each return stroke is finished as planned at its onset, even if the walking command reverses during its execution. In this case, the leg will complete its swing phase in order to immediately initiate another return stroke directed towards the new target point on the AEP circle.

Next, the specific implementation of the three mechanisms that act on the PEP is described. It follows Espenschied's approach in a way that mechanisms 1 and 2 have constant outputs, while mechanism 3 produces a simple ramp that grows with a decreasing distance in between the foot and the PEP circle. Apart from Espenschied, this simple but yet powerful realisation of Cruse's coordination rules has already been successfully employed by other researchers, as for example by Lewinger to generate gaits for the hexapod Bill Ant [Lewinger, 2005]. In the following, all mechanism outputs are normalized. Thus, mechanism 1 has an output of 1 while mechanism 2 has an output of -1. The output of mechanism 3 is in between 0 and -1 as long as the foot of the sending leg is inside the nominal PEP circle and becomes smaller than -1 once the foot crosses the circle. Its value $m_{3,i}$ for leg i is computed according to the following equation wherein $r_{\text{pep\_nom}}$ stands for the nominal PEP radius of the leg and $d_{\text{pep},i}$ for the distance in between the foot and the PEP circle in the direction of the foot velocity as displayed by Fig. 4.3.

$$m_{3,i} = -\left(1 - \frac{d_{\text{pep},i}}{2 \cdot r_{\text{pep\_nom}}}\right) \tag{4.1}$$

At a receiving leg all incoming, normalized values of the mechanisms are multiplied by weights. The resulting sum of all weighted mechanisms that influence an individual leg is then added to its nominal PEP radius. This results either in a larger PEP circle, and thus a prolonged stance phase, or a smaller one causing an earlier transition to the swing phase. In addition to its nominal value, mechanism 1 has a timing component which has a strong influence on the gait characteristics. Being active during the complete swing phase of a sending leg, the mechanism remains active for a short period after the swing to stance transition of the respective leg. In contrast to the original work of Cruse, mechanism 2 does not only become active for a short

time span after the swing-to-stance transition of the sending leg, but remains active throughout its entire stance phase. Based on the influences of neighbouring legs, the PEP radius $r_{\text{pep},j}$ of leg j is then computed at each time step as follows,

$$r_{\text{pep},j} = r_{\text{pep\_nom}} + \sum_i \sum_n k_{n,ij} \cdot m_{n,i}. \tag{4.2}$$

Within this equation, the term $k_{n,ij}$ is the weight to determine how strong the value of mechanism n, $m_{n,i}$, originating at the sending leg i influences the PEP radius of the receiving leg j. The following equation exemplifies the detailed computation of the PEP for the left front leg L1 which is influenced by the behaviour of the left middle leg L2 and the right front leg R1.

$$r_{\text{pep,L1}} = r_{\text{pep\_nom}} + k_{1,\text{L2L1}} \cdot m_{1,\text{L2}} + k_{2,\text{L2L1}} \cdot m_{2,\text{L2}} + k_{2,\text{R1L1}} \cdot m_{2,\text{R1}} + k_{3,\text{R1L1}} \cdot m_{3,\text{R1}} \tag{4.3}$$

Not only the choice and the specific implementation of the coordinating mechanisms have a large influence on the gait characteristics, but also the overall parametrisation. Therefore, an appropriate set of values for the mechanism weights, the nominal AEP and PEP radii, the return stroke velocity, and the time components of mechanism one and two has to be found. Except for Lewinger, who used an optimisation to determine those parameters for the robot Bill Ant [Lewinger, 2005], most previous applications rely on experimentation and manual tuning. This is also the method of choice within this work. The reason for choosing manual parameter tuning is the ambiguity of the optimization criteria related to this problem. Furthermore, it is not clear how to interpret the local optima that result from the large number of parameters, possible initial conditions, and potential walking commands. In this case, despite of being tedious work, manual tuning helps to develop an intuition about how to obtain favourable gait characteristics and to gain some insight into the relevance of the different parameters used. Interestingly, removing the original timing component of mechanism 2 simplified this tuning process drastically. A suitable set of parameters found for the DLR Crawler is given in Table 4.1.

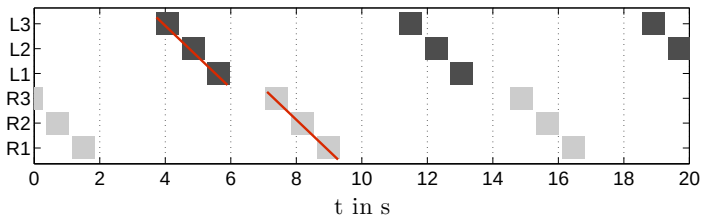
With respect to the effectiveness of the implementation presented within this work and its parametrisation the following observations were made. First, mechanism 0 alone is sufficient to generate stable gaits but the resulting pattern strongly depends on the initial configuration. Employing only mechanism 1 and 2 already establishes stable gaits that show forward-directed waves of protractions along the left and the right side of the robot. Those waves along each side have fixed phase relations in between ipsilateral legs but weak contralateral coupling. Adding mechanism 3 strengthens the

Table 4.1: Gait parameters and coordination mechanism weights

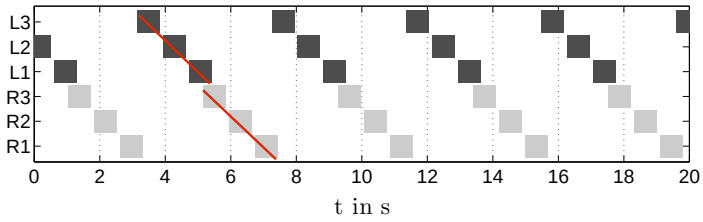
Parameter	Value	Parameter	Value	Parameter	Value
$r_{\text{pep\_nom}}$[mm]	40	$k_{2,\text{L3L2}}$[mm]	5	$k_{3,\text{L2L3}}$[mm]	5
$r_{\text{pep\_max}}$[mm]	50	$k_{2,\text{L2L1}}$[mm]	5	$k_{3,\text{L1L2}}$[mm]	5
$r_{\text{aep\_nom}}$[mm]	40	$k_{2,\text{L3R3}}$[mm]	5	$k_{3,\text{L3R3}}$[mm]	5
v_{swing}[mm/s]	100	$k_{2,\text{L2R2}}$[mm]	5	$k_{3,\text{L2R2}}$[mm]	5
t_{m1}[s]	0.1	$k_{2,\text{L1R1}}$[mm]	10	$k_{3,\text{L1R1}}$[mm]	5
t_{m2}[s]	-	$k_{2,\text{R3R2}}$[mm]	5	$k_{3,\text{R2R3}}$[mm]	5
$k_{1,\text{L3L2}}$[mm]	10	$k_{2,\text{R2R1}}$[mm]	5	$k_{3,\text{R1R2}}$[mm]	5
$k_{1,\text{L2L1}}$[mm]	10	$k_{2,\text{R3L3}}$[mm]	5	$k_{3,\text{R3L3}}$[mm]	5
$k_{1,\text{R3R2}}$[mm]	10	$k_{2,\text{R2L2}}$[mm]	5	$k_{3,\text{R2L2}}$[mm]	5
$k_{1,\text{R2R1}}$[mm]	10	$k_{2,\text{R1L1}}$[mm]	10	$k_{3,\text{R1L1}}$[mm]	5

contralateral coupling and creates a phase shift in between the wave of protractions along the left side and the wave of protractions along the right side. The result at low walking speeds is that a wave of protractions on one side starts upon completion of the wave on the opposite side. However, the value of this phase shift strongly depends on the initial configuration. Additionally enabling mechanism 0 fixes this phase shift and renders it independent for a larger set of initial configurations. Thus, combining the mechanisms 0, 1, 2, and 3 does not only generate stable gaits but also gaits that show the desired forward-directed waves of protractions.

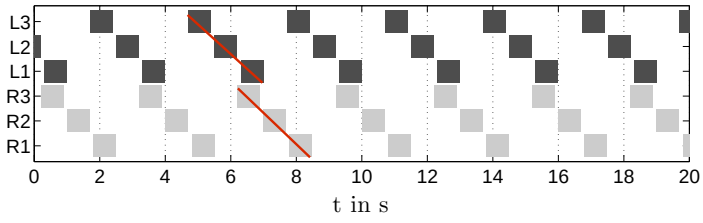
Figure 4.4 shows the resulting gait patterns for forward walking at different velocities. As indicated by the red diagonal lines, forward-directed waves of protractions run along each side of the robot. For increasing walking speeds, their relative phase relations are maintained in between ipsilateral legs, while the waves along the left and the right side are shifted towards each other with respect to time. Thus, starting with a pentapod gait at low speeds the pattern evolves via a tetrapod towards a tripod gait at high speeds. A similar observation is made for rotations about the yaw axis such as for turning to the left shown in Fig. 4.5. Again, with increasing velocity the phase relations in between ipsilateral legs are maintained while the gait pattern of the left side is shifted towards the pattern of the right side.



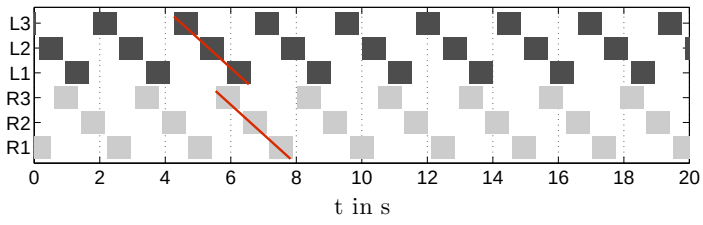
(a) $v_x = 10\,\mathrm{mm/s}$



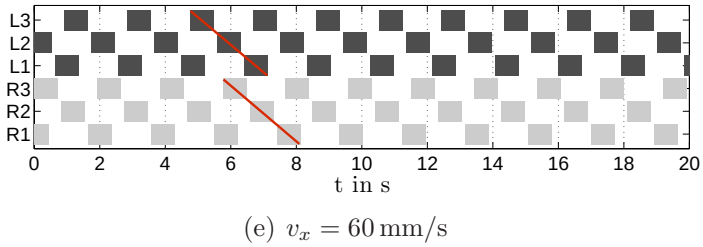
(b) $v_x = 20\,\mathrm{mm/s}$



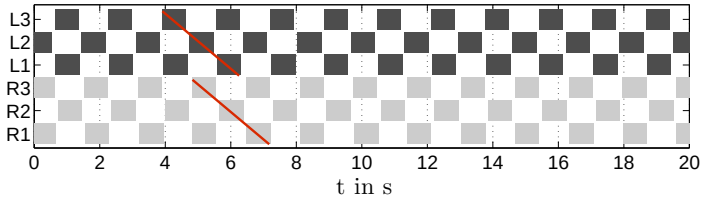
(c) $v_x = 30\,\mathrm{mm/s}$



(d) $v_x = 40\,\mathrm{mm/s}$

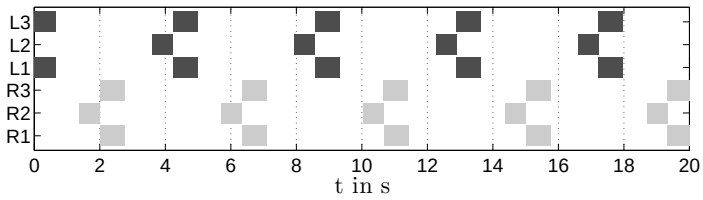


(e) $v_x = 60\,\mathrm{mm/s}$

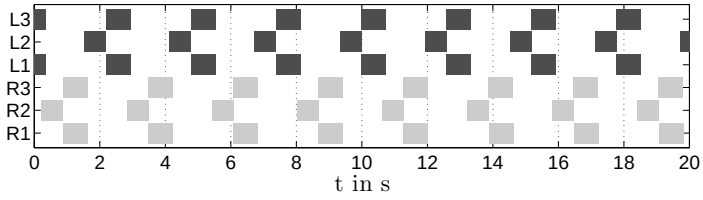


(f) $v_x = 80\,\mathrm{mm/s}$

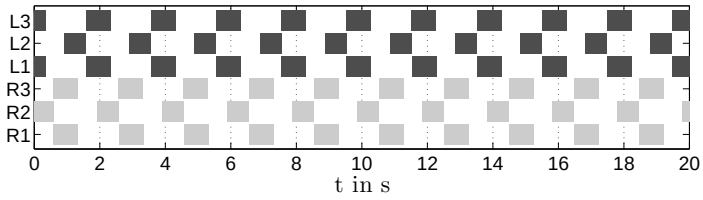
Figure 4.4: Gait simulations for the DLR Crawler walking forward at different velocities (white: stance phase; dark/light grey: swing phase left/right side)



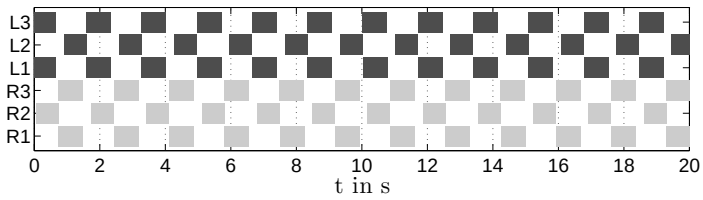
(a) $\omega = 5\,^\circ/\mathrm{s}$



(b) $\omega = 10\,^\circ/\mathrm{s}$



(c) $\omega = 15\,^\circ/\mathrm{s}$

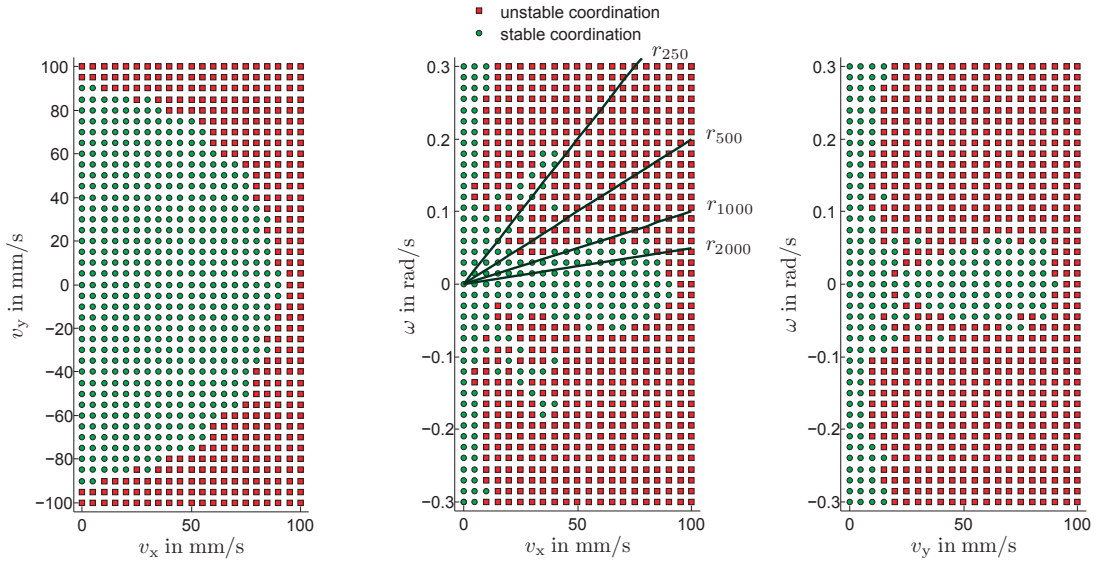


(d) $\omega = 20\,^\circ/\mathrm{s}$

Figure 4.5: Gait simulations for the DLR Crawler turning left at different velocities (white: stance phase; dark/light grey: swing phase left/right side)

In order to assess the capability of the algorithm to generate well-coordinated, omnidirectional gaits, a large number of kinematic simulations has been performed using the parameters given in Table 4.1. First, straight line locomotion was tested by commanding various combinations of forward and lateral velocities according to an equidistant grid with 5 mm/s increments in each direction. The simulations cover 861 combinations of forward and lateral velocities and have been run for a timespan of 3 minutes each. The results are shown in Fig. 4.6(a) wherein the green dots indicate the range of motion commands that produce stable gaits assuming no external disturbance. Within this context, the term "stable" is related to the gait coordination and refers to the emergence of regular gait patterns which are characterized by the following properties. First, neighbouring legs must not step at the same time, and second, no leg has to

stop its motion due to a neighbour not being finished with its return stroke. This does not conclude that all velocity combinations indicated by a red square do not result in effective locomotion, but in those cases some legs regularly invoke brief periods of a central stop due to reaching their preset kinematic limits. For this reason, these gaits are called unstable with respect to the coordination of the stepping sequences. The circular shape, which is formed by the green dots in Fig. 4.6(a), clearly demonstrates the capability of the algorithm to generate omnidirectional gaits across a large range of velocity commands without any impairment in relation to the walking direction. The maximum walking velocity that can be achieved with the given parametrisation is about 90 mm/s and is bounded by the swing velocity of 100 mm/s. Thereby, no perfect tripod gait at the swing velocity can be generated. Such gait would require instantaneous transitions from the left to the right tripod and vice versa, which is prevented by the time component of mechanism 1 that remains active for a short time after the swing-to-stance transition of a sending leg. Combinations of sideways and backward walking have been tested as well, but the presentation of the results is omitted in this place as they show similar behaviour to combined forward and sideways walking.



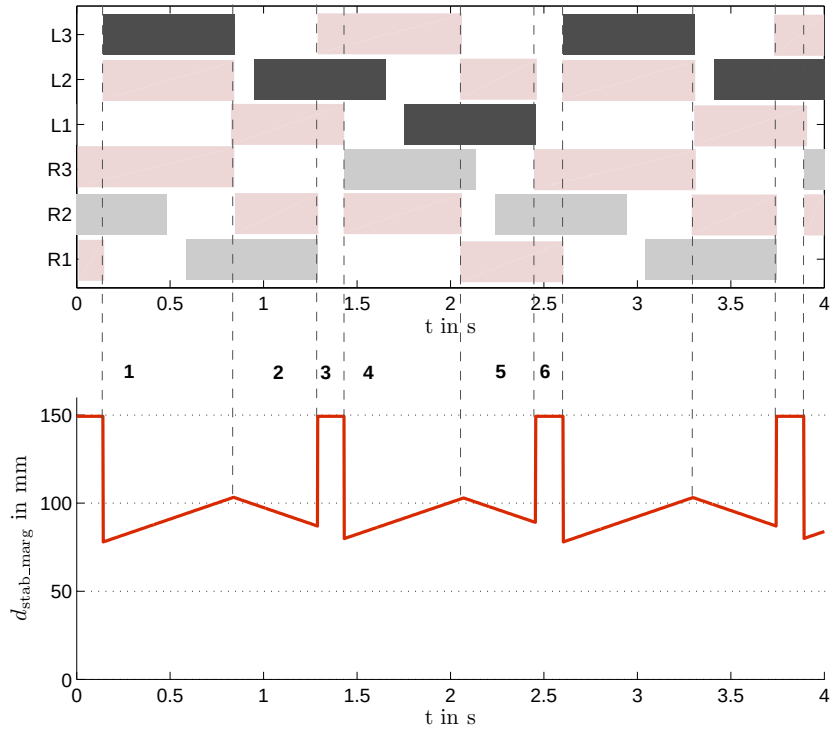
(a) forward / sideways (b) forward / turning (c) sideways / turning

Figure 4.6: Stability assessment of the gait coordination of the DLR Crawler for various combinations of velocity commands; green dots indicate stable and red squares unstable gaits; the dark lines in b) indicate combinations that result in curves with an equal radius

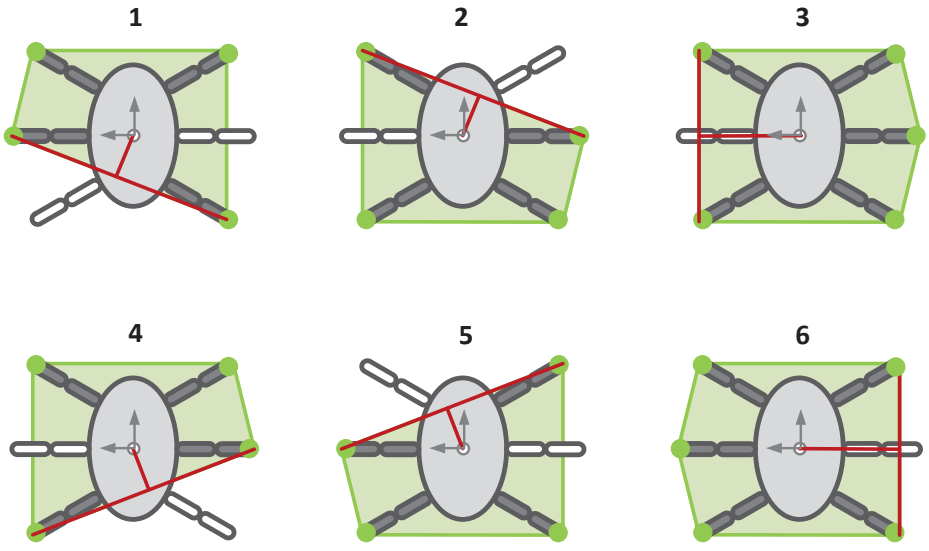
Within Figs. 4.6(b) and 4.6(c) the results for turning and curve walking are presented. Hereby, Fig. 4.6(b) shows the combination of a forward velocity and a yaw

rate, i. e. an angular velocity about the body-fixed z-axis, which results in curves to the left for positive values of ω and curves to the right for negative ones. The diagram shows that pure turning on the spot is possible up to an angular velocity of about 20 °/s to both sides. Considering the combination of angular velocity and forward velocity, the tongue-shaped cluster to the right indicates that the maximum attainable walking speed is tightly coupled to the radius of the curved path. To visualize this relation, the 4 dark green lines in Fig. 4.6(b) indicate different combinations of yaw rate and forward velocity that result in curves with radii of 0.25 m, 0.5 m, 1 m, and 2 m denoted by r_{250}, r_{500}, r_{1000}, and r_{2000}, respectively. While walking along a curve with a radius of 0.5 m is only possible up to a velocity of 20 mm/s, curves with radii larger than 2 m nearly allow walking across the full range of possible forward speeds. A similar behaviour is observed for motions that combine lateral and angular velocity commands, as shown in Fig. 4.6(c). Herein, negative yaw rates result in inward facing curves to the left while positive yaw rates produce outward facing curves to the left. Again, the presentation of the results for negative lateral velocities is omitted as they produce similar curves that are directed to the right of the robot.

So far, the gait coordination presented guarantees stable gaits in the sense that no neighbouring legs step at the same time. With respect to the overall locomotion this is a helpful property. Nevertheless, it is not sufficient to conclude static stability, which requires the gravity-aligned projection of the COM onto the ground to be within the polygon of support. In order to quantitatively assess the static stability, the so-called stability margin is a commonly used measure. This margin is defined to be the shortest distance in between the projected COM and the edges of the polygon spanned by all feet in contact. To give an example, Fig. 4.7(a) shows the gait diagram and the resulting stability margin for forward walking at 40 mm/s. Therein, the red bars mark the supporting legs that span the corresponding edge with the smallest distance to the projected COM. To further visualize the concept using the above example, Fig. 4.7(b) schematically indicates the stability margin and the related edge of the support polygon at different times of the gait cycle. Those sketches clearly show that lifting either a front or a hind leg reduces the stability margin most, and thus results in the least stable configurations. However, with all legs functional, the stability margin is large enough across the complete gait cycle and grants static stability for the anticipated range of walking velocities. Due to the comparatively small velocities, dynamic stability measures do not have to be considered. With respect to the static stability margin itself, there are a few underlying assumptions that simplify its computation at this stage. First, the ideal case is considered where all mass is concentrated within the body, which results in a fixed COM position. In contrast, the COM of the real robot



(a)



(b)

Figure 4.7: a) Gait simulation for the DLR Crawler walking forward at $v_x = 40\,\text{mm/s}$ and the related stability margin over time; red bars indicate the stance legs that span the edge with the smallest distance to the COM projection (white/red: stance phase; dark/light grey: swing phase left/right side); b) Schematic sketches of the foot configuration, stability margin and the corresponding edge at distinct times marked in (a)

moves within small bounds due to the motion of the legs and their associated mass. Second, only horizontal plane motions are considered at this time, but extensions to slopes and uneven terrain are straightforward.

Returning to the advantages of forward-directed waves of protractions, the related gaits show larger stability margins than those with a reversed stepping sequence, such as given in Fig. 4.8. A comparison of the stability margins across the range of feasible forward velocities yields the results depicted in Fig. 4.9. While the stability margin for forward-directed waves of protractions at low walking speeds is significantly larger than for rearward-directed ones, the difference decreases with increasing speed. Starting at a 45 % larger stability margin at 10 mm/s, it decreases via 25 % at 50 mm/s towards similar values at the highest walking velocities. This convergence at high speeds is an obvious behaviour since forward- as well as backward-directed stepping sequences produce tripod gaits at those velocities.

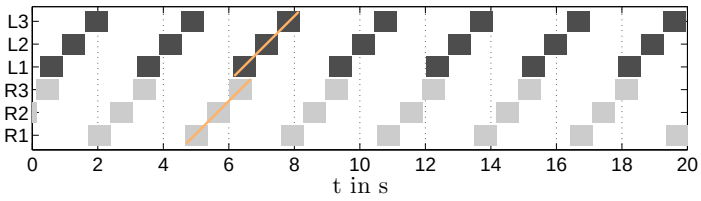


Figure 4.8: Gait simulation for the DLR Crawler walking forward at $v_x = 30\,\text{mm/s}$ with front-to-back waves of protractions; only four mechanism weights are changed: $k_{2,\text{R1L1}} = 5\,\text{mm}$, $k_{2,\text{L1R1}} = 5\,\text{mm}$, $k_{2,\text{R3L3}} = 10\,\text{mm}$, $k_{2,\text{L3R3}} = 10\,\text{mm}$; (white: stance phase; dark/light grey: swing phase left/right side)

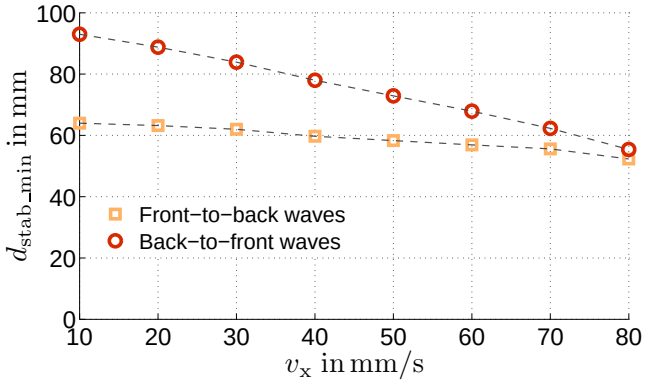
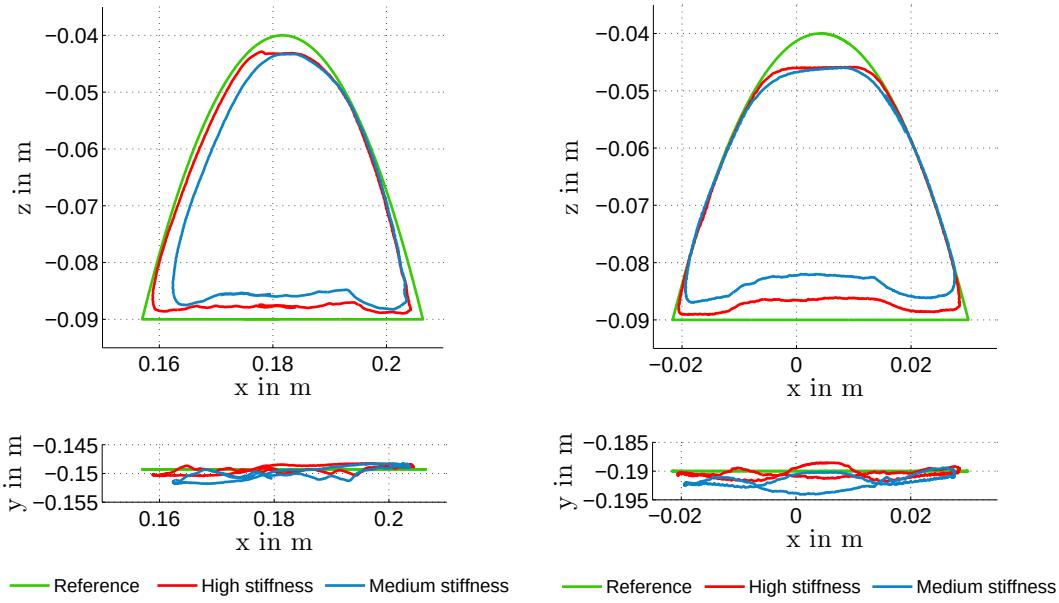


Figure 4.9: Minimum stability margins of the DLR Crawler for walking forward at various velocities with forward- and rearward-directed waves of protractions (orange square: front-to-back waves; red circle: back-to-front waves)

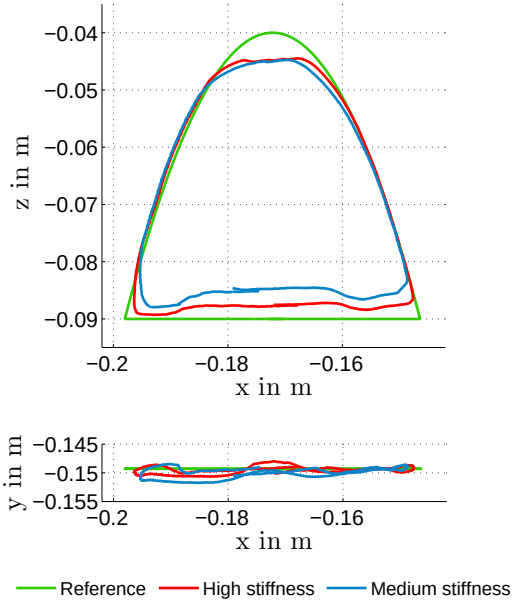
4.3 Leg Reference Trajectories

So far, the gait coordination algorithm only considers the Cartesian positions of the feet with respect to a body-centred coordinate system. While the high-level motion commands determine the stance-phase trajectories of the feet, the swing-phase trajectories are individually calculated by second-order polynomials. Those polynomials connect the starting point of a foot located on the PEP circle and its target point on the AEP circle while meeting the desired step height as well as the preset protraction speed with respect to the horizontal plane. The inverse kinematics algorithm presented in Section 3.4 computes joint angle reference trajectories from the desired Cartesian positions of the feet. Those joint angles then serve as desired values for the joint compliance controller presented in Section 3.2. To give an example, Fig. 4.10 shows the Cartesian reference trajectories for the right-side feet of the DLR Crawler while walking with a tripod gait on flat lab floor. Those trajectories are compared to the realised Cartesian foot trajectories, that are computed from joint angle measurements using the forward kinematics presented in Section 3.3. Obviously, the joint compliance controller shows good tracking performance for high and medium stiffness settings along the rising and falling edges of the swing trajectories. Only the right front leg experiences larger deviations along the rising edge of the medium stiffness trial. However, this behaviour is most likely caused by an inaccurate friction estimate. Furthermore, in this example none of the feet shows good trajectory tracking at the peak of its step, which does not necessarily indicate bad controller performance. The explanation of this behaviour is straightforward and shows another feature of the controller implementation. In order to avoid hitting the mechanical end stops of the joints, the joint compliance controller simply limits the possible range of desired joint angles and guides the foot along the work space boundary in this case. In addition to these deviations during the swing phase, each of the three legs shows deviations from their reference trajectories during their stance phase. These are a direct result from the joint compliance control, as the virtual joint springs deflect to produce the torques that support the body weight. As expected, the deviations from the reference trajectory are smaller for higher joint stiffness settings. Another interesting observation is the larger mid-step tracking error of the real foot trajectories which indicates higher leg loads during this phase. However, the explanation of this behaviour is simple again. Partially overlapping tripods, and thus short phases with more than three legs in stance, reduce the individual leg load at the beginning and the end of a step.



(a) Right front leg (R1)

(b) Right middle leg (R2)

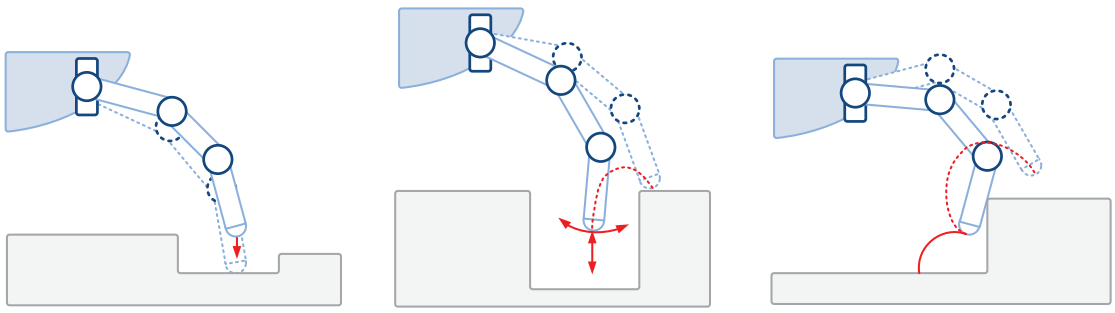


(c) Right hind leg (R3)

Figure 4.10: Cartesian trajectories of the right-side feet of the DLR Crawler with respect to the body coordinate system for walking on flat lab floor at 60 mm/s - each diagram shows the reference trajectory and the trajectories for medium as well as high joint stiffness settings that were computed by forward kinematics from Hall-sensor-based joint angle measurements

4.4 Leg Reflexes and Safety Value

While the gait coordination algorithm influences the phase relations in between the legs, and thus allows for temporal adaptations, it does not account for spatial variations of the terrain. However, moving across natural terrain requires the robot to constantly adapt its configuration to accommodate height changes of the ground surface. Therefore, each leg has to adjust its extension during the stance phase as well as its step height during the swing phase. In case of having perfect a priori knowledge of the terrain, the required adaptations could be preplanned. Since such detailed knowledge is not always available or difficult to attain, another approach is chosen within this work. Herein, each leg reactively adapts its step height as well as its extension by utilizing three different types of reflexes. Combined with the flexible gait coordination, these reflexes enable the robot to negotiate obstacles within the walking height autonomously.



(a) Stretch reflex (b) Search reflex (c) Elevator reflex

Figure 4.11: Leg reflexes

The three well-known reflexes employed are: the stretch reflex, the search reflex, and the elevator reflex, as depicted in Fig. 4.11. For proper function, each of them requires the detection of contacts with the environment. This contact detection is realised by comparing joint torque measurements to state dependent torque thresholds. The reflex that is responsible to enforce the ground contact of a supporting leg is the stretch reflex. Since the nominal swing phase ends at a predefined leg extension, the stretch reflex is triggered if a leg does not detect ground at the anticipated height. Furthermore, it is activated if the leg loses ground contact during the stance phase. Being active, the stretch reflex extends the leg with respect to the negative z-direction of the body coordinate system until the foot detects ground contact or the leg has reached a preset maximum extension. If the foot has found ground contact, the additional leg extension is kept until the end of the current stance phase or until reaching a high

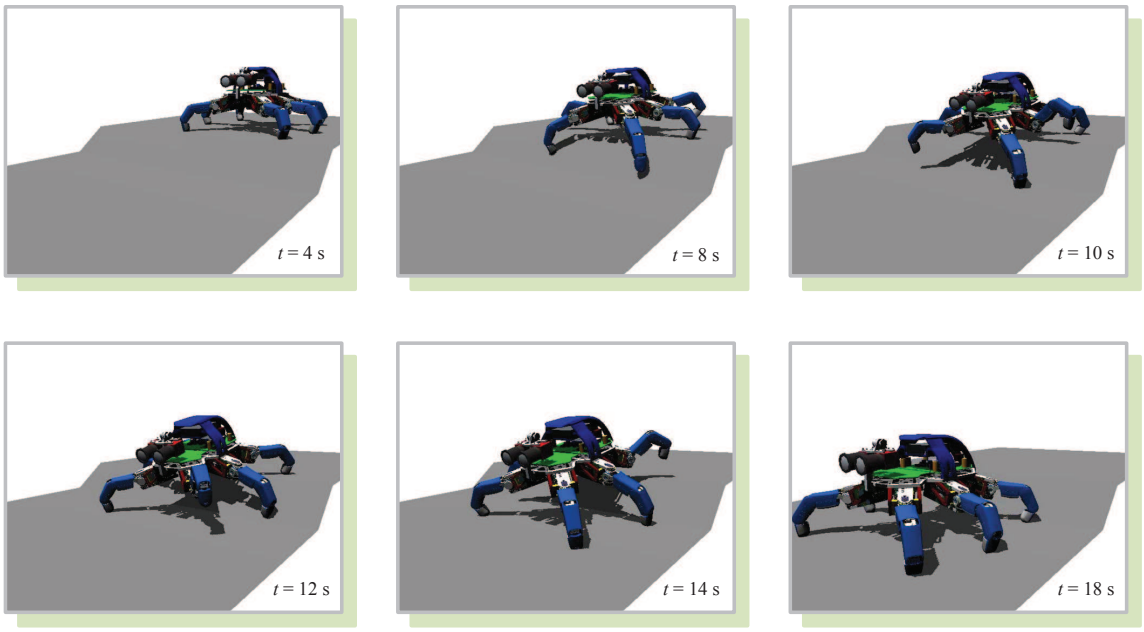
torque threshold that triggers its reduction. With the onset of the next swing phase, any reflex-based extension will be quickly reduced to zero. If the stretch reflex was not successful in establishing ground contact, the search reflex is triggered. This reflex initiates short, successive stepping motions into the walking direction in order to find a support within the vicinity. Without success, those exploratory steps are followed by stepping motions into orthogonal directions. If the leg does still not find any support, it triggers a central stop of the robot and returns to its last foothold. At this time a higher-level algorithm gets activated in order to search for a solution. This algorithm, for example, starts an extended search that includes coordinated body motions to increase the reach of the leg, or it plans a new path for the whole system. The last reflex employed within this work is the elevator reflex which is triggered once a leg hits an obstacle during its swing phase. In this case, the elevator reflex retracts and raises the leg about a fixed amount before it proceeds with its stepping motion. If necessary, this procedure is repeated up to a preset maximum step height. If the elevator reflex is not successful in clearing the obstacle, a higher-level algorithm is activated as well. First, the clearance of the leg is increased by equally extending all supporting legs. If this does not succeed, then the algorithm initiates a new planning phase to find a path around the obstacle. Remarkably, this small set of reflexes in combination with the flexible gait coordination is sufficient to master a large variety of terrains and obstacles. Thereby, the basic approach does not utilize any active posture control but lets the robot follow the inclination of the terrain by the interplay of reflexes. While this approach clearly simplifies the control, it is only suitable for smaller robots that have legs strong enough to temporarily sustain larger unbalanced loads. In contrast, large and heavy robots require the careful distribution of contact forces in order to avoid overloads within single legs. However, compared to small robots such large systems usually carry enough computational power to do so.

In addition to the reflexes presented above, this work introduces a binary "safety value". This new parameter enables the navigation layer to influence the obstacle-crossing behaviour of the robot. Set to 1, the robot walks more carefully, while it takes greater risks at a safety value of 0. The functional principle of this approach is the following. A safety value of 1 requires each leg to provide proper support before its state is allowed to switch from swing to stance. Thus, being not successful in making ground contact, a leg cannot enter its stance phase and has to trigger a central stop for the complete robot. This, in turn, activates higher-level algorithms which try to find a solution to the problem. In contrast, if the safety value is set to 0, then each leg immediately switches to stance at the end of its nominal swing trajectory, independent of its contact state. Being in stance, each leg without sufficient ground support then

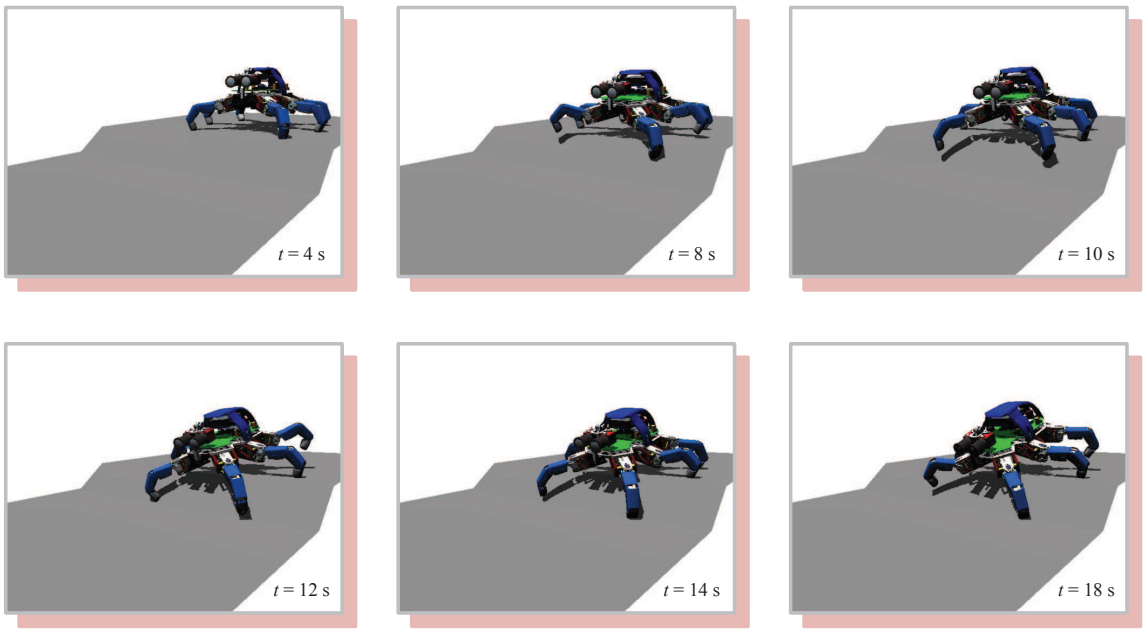
triggers the stretch reflex and extends while the gait coordination smoothly proceeds. Even though this approach seems contradictory to the original ideas for the walking layer, such behaviour is very useful under certain conditions. To give an example, the risky mode enables the robot to descend large downward steps which it could not traverse otherwise. The explanation is as follows: Within the safe mode, the front legs of the robot trigger the stretch reflex at the downward step and extend in order to find a support on the lower plane. If this attempt is not successful, they will trigger the search reflex which might also fail in establishing ground contact. Even though only few centimetres might be missing, the robot would stop and the walking layer would consider the downward step to be non-traversable. Nevertheless, a very different behaviour emerges if the safety value is set to 0. In this case, each leg immediately switches to stance at the end of its nominal swing trajectory, not considering its contact state. Thus, after crossing the upper edge, the front legs stretch to their maximum extent in trying to find ground support. Even though they do not reach the ground, the gait proceeds and the middle and hind legs push the robot across the edge. At some point, the body pitches downwards and the stretched front legs contact the lower plane. There, they provide the necessary support while the middle and hind legs follow across the edge in a similar way. Hence the safety value enables an informed navigation layer to influence the walking layer such that it is able to descend large steps.

Next, several simulations demonstrate the effectiveness of the safety value as well as the reflexes introduced above. Utilizing a Matlab/Simulink implementation of the simplified dynamics model presented in Section 3.5 as well as an Inventor-Graphics-based visualisation, the simulated robot climbs and descends slanted steps of different heights. The first set of simulations tests the reflex behaviours. For this purpose, the safety value is set to zero and the robot traverses an 8 cm step twice in both directions, once with all reflexes enabled and once with all reflexes disabled. Thereafter, the robot descends a 5 cm step to demonstrate the effects of the safety value. In this case, all reflexes remain enabled while the safety value is changed from 0 to 1 in between the trials. As listed in Table 4.1, four gait parameters have been changed for all of the simulations to improve the obstacle crossing behaviour of the simulated robot by shorter steps ($r_{\text{aep\_nom}}$, $r_{\text{pep\_nom}}$ and $r_{\text{pep\_max}}$) and reduced ground impacts (v_{swing}).

Considering the 8-cm-downward step first, Fig. 4.12(a) displays a series of images from the trial with all reflexes enabled. As shown in the middle image of the top row, both front legs activate the stretch reflex and extend downwards in order to find a foothold along the slope. Due to the active compliance, the resulting configuration causes increased flexion of the middle legs as they carry now larger parts of the body weight. Moving the body further forward, the loading of the hind legs constantly



(a) All reflexes enabled



(b) All reflexes disabled

Figure 4.12: Simulation results for crossing an 8-cm-downward step with the DLR Crawler (using the simplified dynamics model presented in Section 3.5)

Table 4.2: Modified gait parameters

Parameter	Value	Parameter	Value
$r_{\text{pep\_nom}}$[mm]	30	$r_{\text{pep\_max}}$[mm]	45
$r_{\text{aep\_nom}}$[mm]	30	v_{swing}[mm/s]	80

decreases up to a value below the stance phase threshold of the stretch reflex. Still being in their stance phase, the hind legs trigger this reflex to enforce the ground contact. As a result, the hind legs lift the posterior part of the robot. This causes a downward tilted body as indicated by the positive pitch angle β (green graph) shown in Fig. 4.14. At this time, the shortest distance from the robot body to the ground is in between the center of its bottom side and the upper edge of the slope. If a middle leg steps now, it will surely hit the edge and subsequently trigger its elevator reflex to overcome this obstacle. Once the hind legs are placed on the slope and the front legs have reached the lower plane, the middle legs are required to execute the stretch reflex as shown in the middle image of the bottom row in Fig. 4.12(a). Especially, crossing such edges requires the legs to activate the elevator as well as the stretch reflex. Figures 4.12(b) and 4.13 clearly demonstrate that a robot without reflexes is not able to cross the downward step. The reason for the observed behaviour is twofold. First, the front legs do not find sufficient support on the slope due to the missing stretch reflex. Furthermore, the middle legs do not step high enough to cross the upper edge due to the missing elevator reflex. Thus, without reflexes, the robot is caught in a bouncing and rocking motion about the upper edge which is indicated by the oscillating pitch, roll, and yaw angles presented in Fig. 4.14.

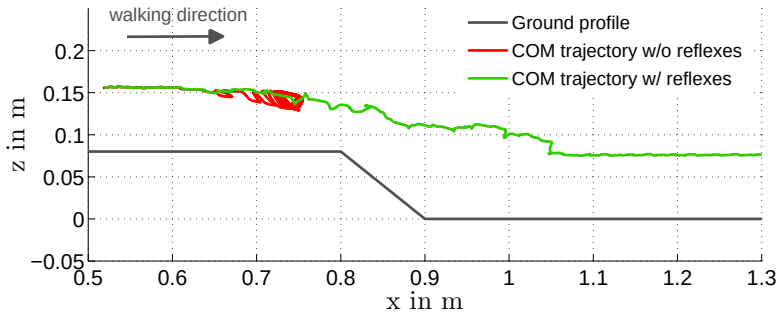


Figure 4.13: Centre of mass trajectories of the simulated DLR Crawler for crossing an 8-cm-downward step at a forward velocity of $v_x = 40\,\text{mm/s}$: reflexes enabled (green) / disabled (red)

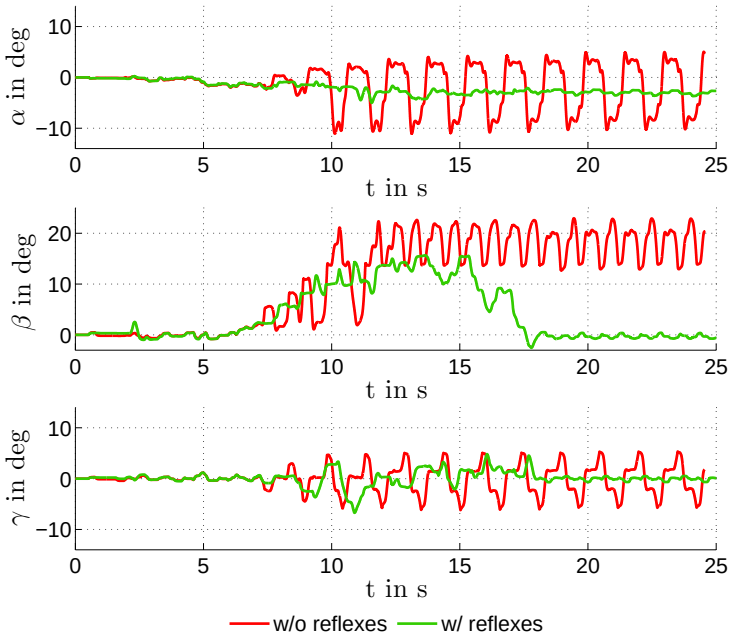


Figure 4.14: Yaw (α), pitch (β) and roll (γ) angles of the simulated DLR Crawler for crossing an 8-cm-downward step at a forward velocity of $v_x = 40\,\mathrm{mm/s}$: reflexes enabled (green) / disabled (red)

Returning to the reflex activity while crossing the downward step, Fig. 4.15 displays the joint angle trajectories of the left front leg as an example. Therein, the coloured, dashed ellipses mark the activation of either a stretch reflex (red) or an elevator reflex (brown). As already mentioned above, each front leg predominantly triggers the stretch reflex upon entering the slope and the elevator reflex after reaching the lower plane. Within the present example, the elevator reflex is activated each time early in the swing phase. At this point, the left front leg raises and unloads, which causes the robot body to tilt towards its left front. This results in a collision of the raising leg with the ground, which triggers the elevator reflex. Thereafter, the foot rapidly raises, and thus provides sufficient ground clearance to avoid stumbling. While this section only displays exemplary downward step joint trajectories for the left front leg, Appendix B presents the corresponding results for the other legs.

Similar to this example, the Figs. 4.16 and 4.19 demonstrate the importance of reflexes in crossing the same step in an upwards direction. First, the elevator reflex allows the front legs to master the slope and to clear the upper edge. Next, as shown in Fig. 4.18, the stretch reflex enables the left front leg to find a support on the upper plane. This, in turn, helps the Crawler to maintain its beneficial upward-tilted configuration, which redirects the thrust of the backward moving hind legs to push the

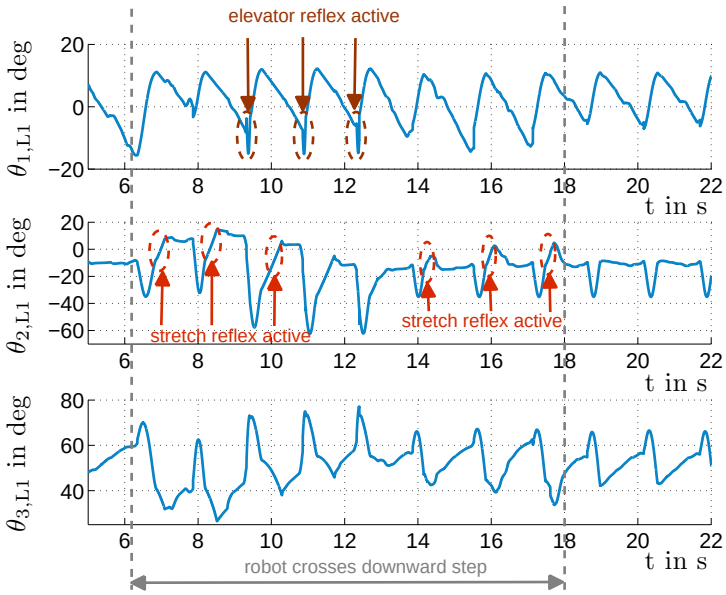


Figure 4.15: Joint trajectories of the left front leg (L1) of the simulated DLR Crawler for crossing an 8-cm-downward step; all reflexes are enabled; red ellipses mark stretch reflex activity; brown ellipses mark elevator reflex activity

robot upwards. Furthermore, the stretch reflex is crucial to enforce the ground contact of the middle legs while they transition from the lower plane to the slope as shown in Fig. 4.19(a). Once the front and middle legs are placed on the upper plane, the loading of the hind legs decreases. Upon reaching the respective torque threshold, the stretch reflex gets active and each of the hind legs extends. This behaviour induces a transition of the body inclination such that the negative pitch angle of the upward-tilted robot decreases to zero, as shown in Fig. 4.17.

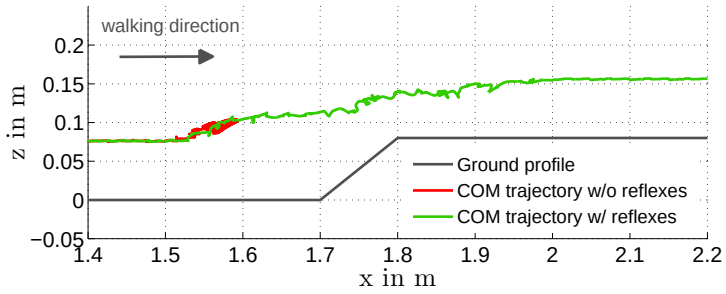


Figure 4.16: Centre of mass trajectories of the simulated DLR Crawler for crossing an 8-cm-upward step at a forward velocity of $v_x = 40\,\mathrm{mm/s}$: reflexes enabled (green) / disabled (red)

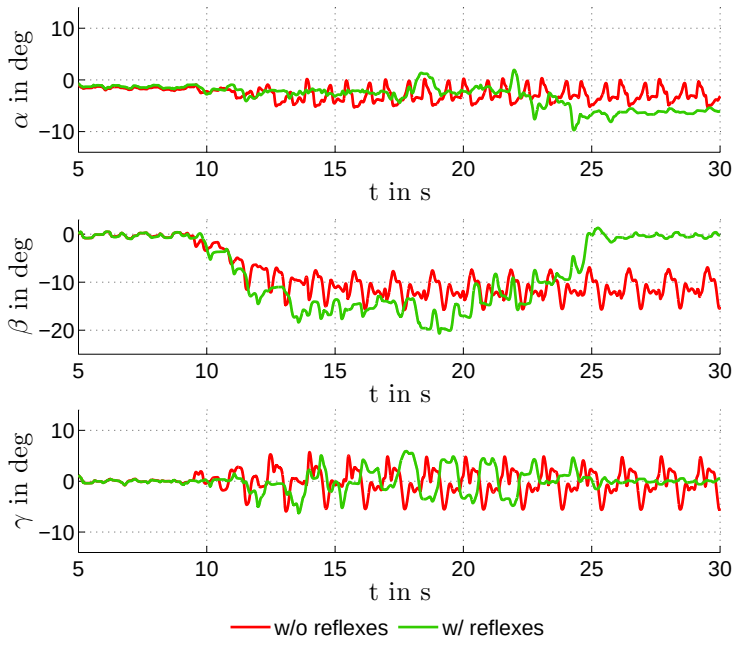


Figure 4.17: Yaw (α), pitch (β) and roll (γ) angles of the simulated DLR Crawler for crossing an 8-cm-upward step at a forward velocity of $v_x = 40\,\mathrm{mm/s}$: reflexes enabled (green) / disabled (red)

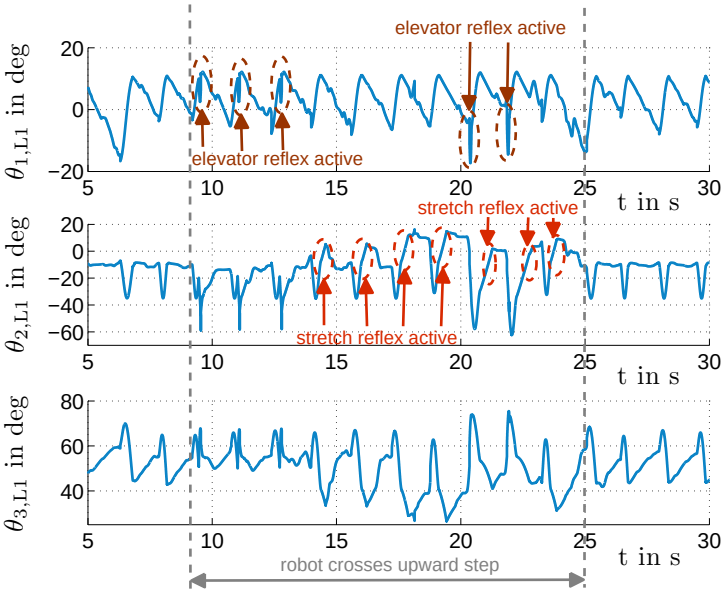
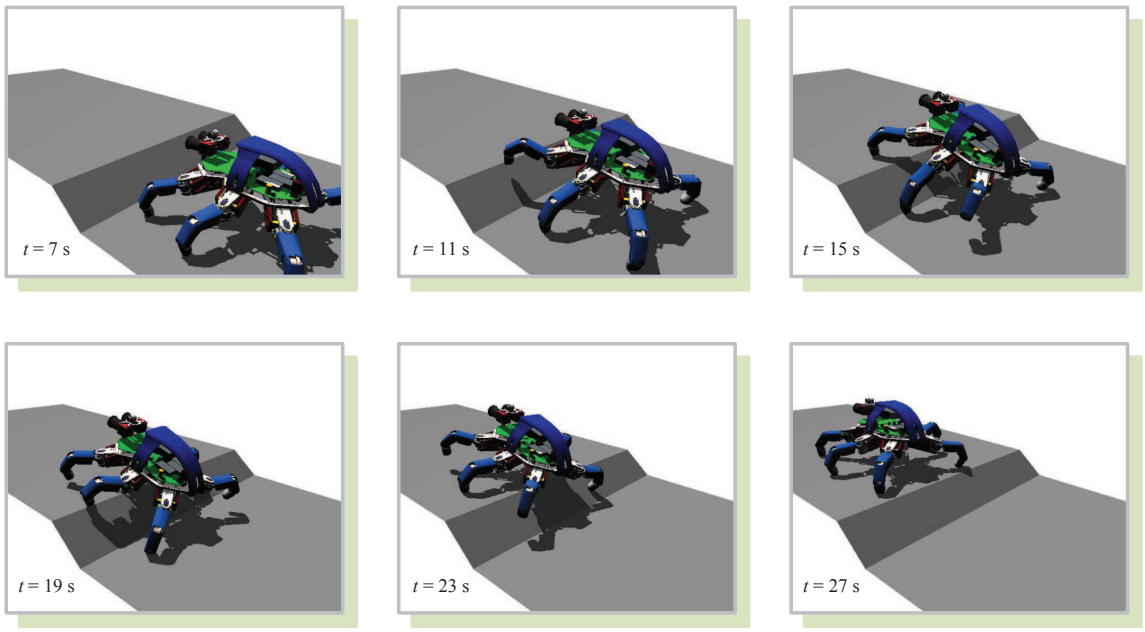
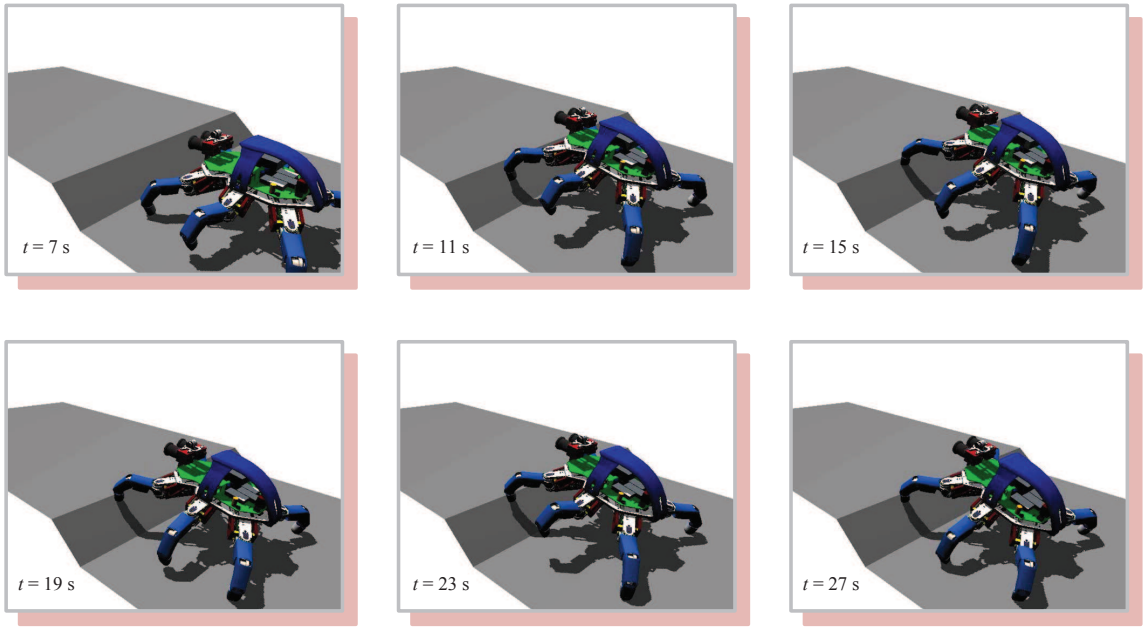


Figure 4.18: Joint trajectories of the left front leg (L1) of the simulated DLR Crawler for crossing an 8-cm-upward step; all reflexes are enabled; red ellipses mark stretch reflex activity; brown ellipses mark elevator reflex activity



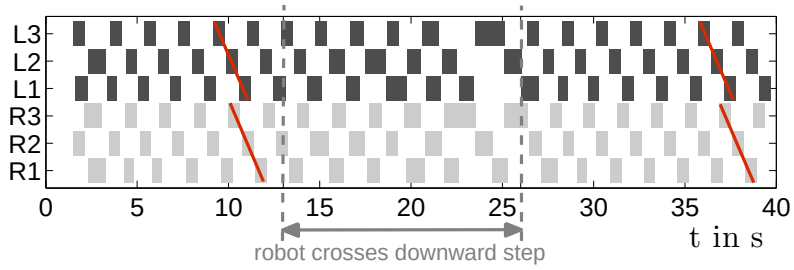
(a) All reflexes enabled



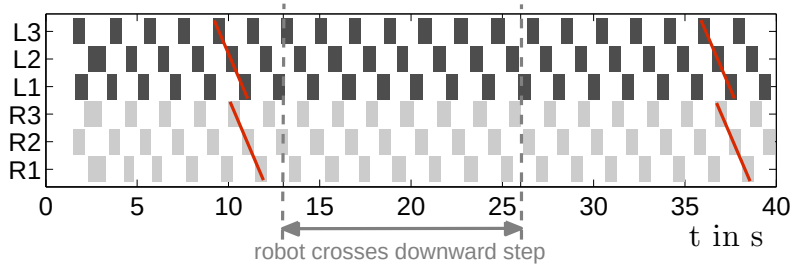
(b) All reflexes disabled

Figure 4.19: Simulation results for crossing an 8-cm-upward step with the DLR Crawler (using the simplified dynamics model presented in Section 3.5)

Next, this section proceeds to evaluate the influence of the safety value on the gait of the robot. For this purpose, the simulated robot descends a 5 cm step twice with all reflexes enabled but different safety value settings. At first, Fig. 4.20 compares the gait patterns that emerge for a safety value of 1 and a safety value of 0. While those gaits are similar for flat ground, they differ substantially when the robot crosses the step. Obviously, the walking layer produces a less regular pattern while being in the safe mode, i. e. the safety value is set to 1. The reason for this is that each leg requires proper ground contact before it switches into the stance phase. Thus, each stance leg delays the onset of its swing phase as long as its stepping neighbour is searching for support. This, in turn, induces changes of the relative timing in between the legs, which might even stop the complete robot for a short period. In contrast, the resulting gait pattern for the risky mode, i. e. the safety value is set to 0, is quite regular and shows only small changes of the leg cycle. Those changes mostly trace back to elevator reflex activity and the related delays. Nevertheless, the safety value does not only affect the gait pattern but also the overall motion of the robot. This becomes apparent in the pitch angle trajectories shown in Fig. 4.22. Therein, for a safety value of 0, the robot experiences stronger pitch oscillations while crossing the step. In addition, the joint torques are less smooth in this case, which is shown in Fig. 4.23 for the left front leg. Again, the corresponding trajectories of the other legs are presented in Appendix B.



(a) Safety value: 1



(b) Safety value: 0

Figure 4.20: Gait diagrams for descending a 5 cm step with the simulated DLR Crawler at a forward velocity of $v_x = 40\,\mathrm{mm/s}$ and different safety value settings

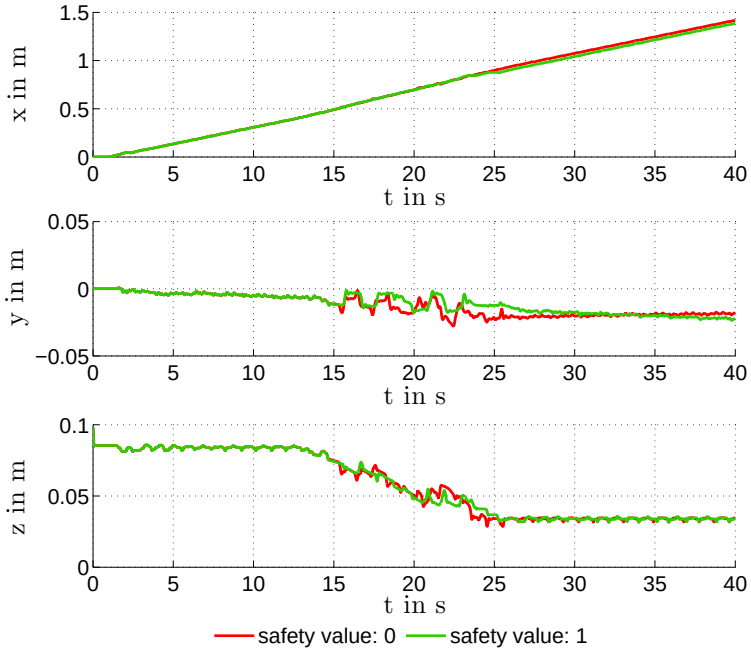


Figure 4.21: Centre of mass trajectories of the simulated DLR Crawler for crossing a 5-cm-downward step at a forward velocity of $v_x = 40\,\mathrm{mm/s}$: all reflexes enabled; safety value = 1 (green) / safety value = 0 (red)

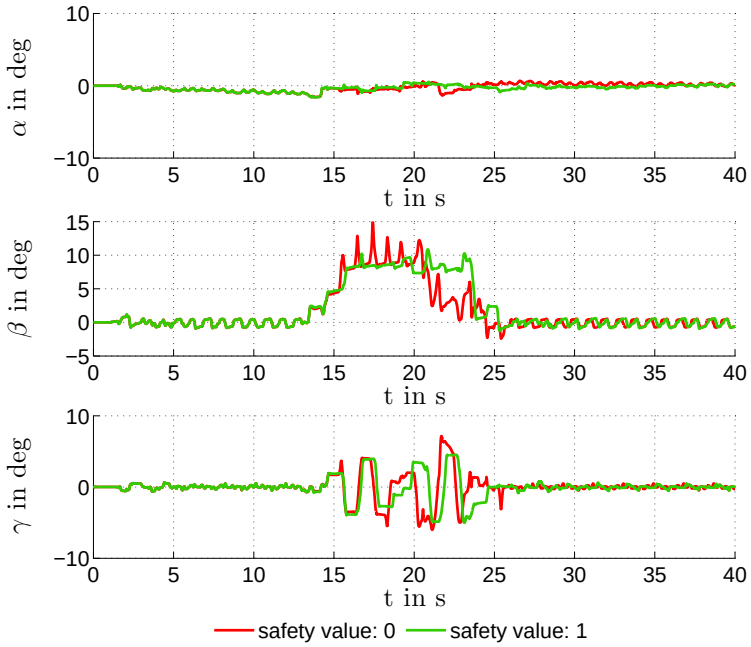
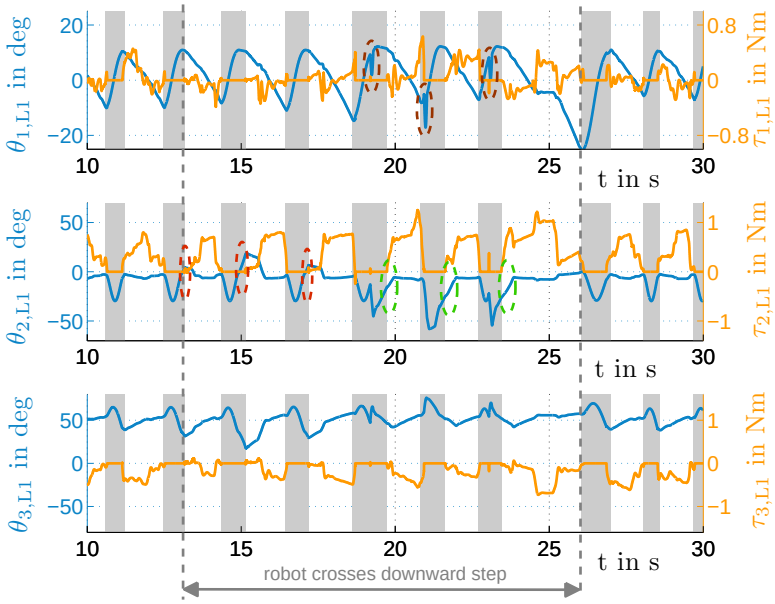
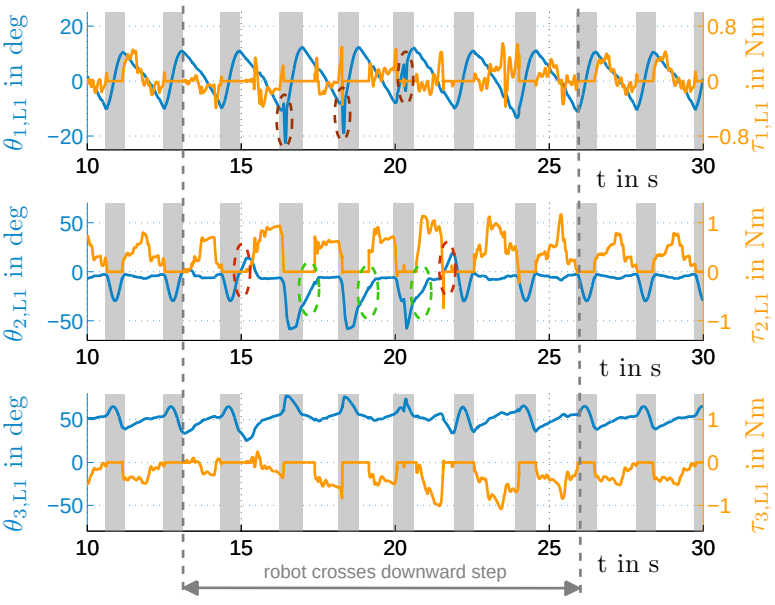


Figure 4.22: Yaw (α), pitch (β) and roll (γ) angles of the simulated DLR Crawler for crossing a 5-cm-downward step at a forward velocity of $v_x = 40\,\mathrm{mm/s}$: all reflexes enabled; safety value = 1 (green) / safety value = 0 (red)



(a) Safety value: 1



(b) Safety value: 0

Figure 4.23: Joint angles (blue) and joint torques (orange) of the left front leg (L1) of the simulated DLR Crawler for crossing a 5-cm-downward step: reflexes enabled; stretch reflex (red ellipse); elevator reflex (brown ellipse); reflex height reduction (green ellipse)

4.5 Experiments

Following the previous sections on the development of gait and control algorithms for six-legged robots, this section summarizes the experimental validation. For this purpose, the DLR Crawler serves as a test platform throughout various trials within laboratory and outdoor testbeds. Recorded trajectories as well as series of images extracted from video material document the results. During most of the experiments, an operator manually steered the robot by velocity commands, using a Spacemouse as input device. However, this section also presents two experiments wherein the developed gait algorithm is part of a stereo-vision-based autonomous navigation framework.

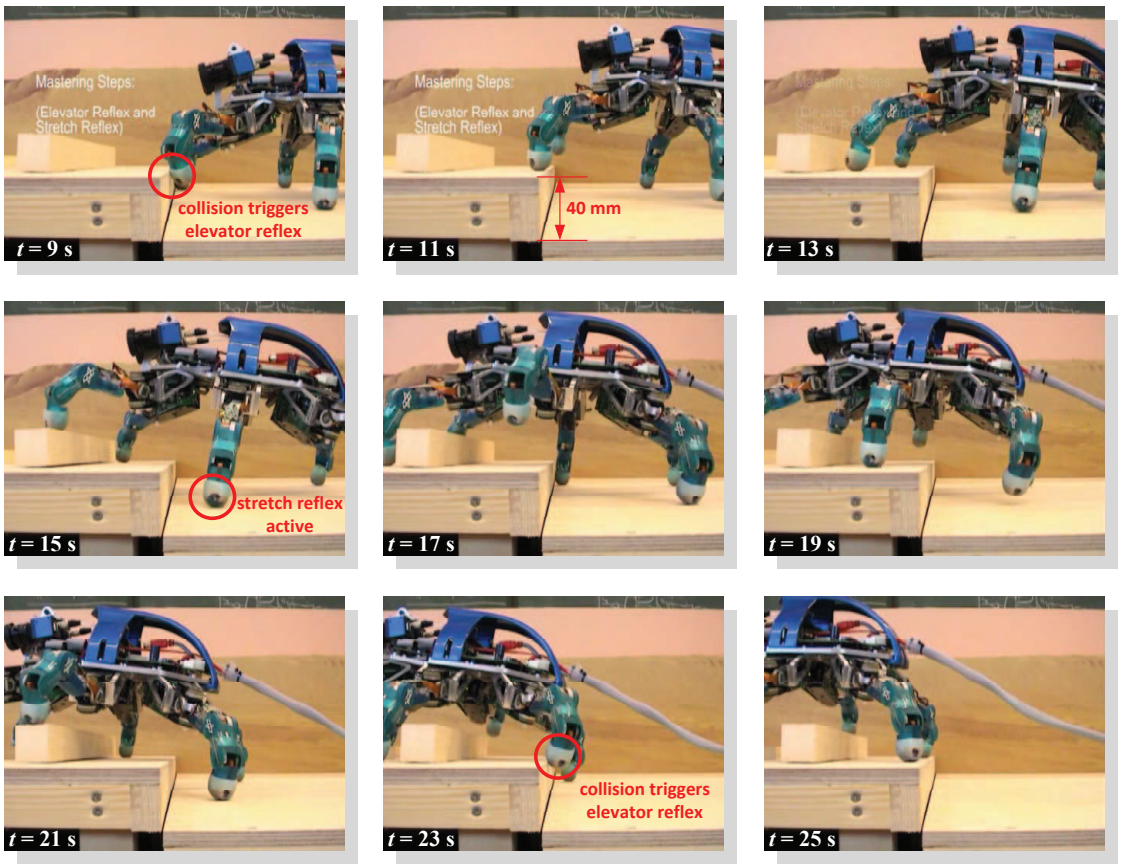


Figure 4.24: The DLR Crawler climbs a 4 cm step

Within the first experimental trial shown in Fig. 4.24, the DLR Crawler climbs a 4 cm step. The height difference is large enough to cause collisions of the front legs which then have to execute the elevator reflex in order to reach the upper platform. Being on top, the front legs extend towards their nominal height, and thus push the robot upwards. This, in turn, requires the middle legs to execute the stretch reflex

to provide sufficient support on the lower platform. Thus, the interplay of reflexes induces a terrain-following behaviour of the robot body which does not require any active posture control.

Figure 4.25: The DLR Crawler crosses stones within the indoor gravel testbed

The next experiment, shown in Fig. 4.25, presents a scenario wherein the DLR Crawler has to cross a group of larger stones within the indoor gravel testbed. As for the previous trial, those obstacles require the robot to exert multiple reflexes, and thus to permanently adapt the gait coordination. Therefore, the experiment demonstrates that the robot effectively walks on gravel, easily handles a change of the substrate, and is able to negotiate a larger obstacle within its walking height. In addition, the images clearly show that the body inclination follows the terrain instead of remaining horizontal.

During the next trial, presented in Fig. 4.26, the robot walks across an outdoor gravel field. This terrain comprises gravel with diameters up to 10 cm, which creates multiple large steps and ditches, and therefore requires the use of all reflexes. Furthermore, loose stones challenge the joint controllers and induce slippage. However, throughout all trials, the compliance controller remained stable and the robot always was able to accommodate to the height differences. In addition to crossing gravel surfaces, the robot easily walks on normal soil or cut grass. However, sandy terrain was not tested due to the open leg structure of the robot and the potential damage of the gears.

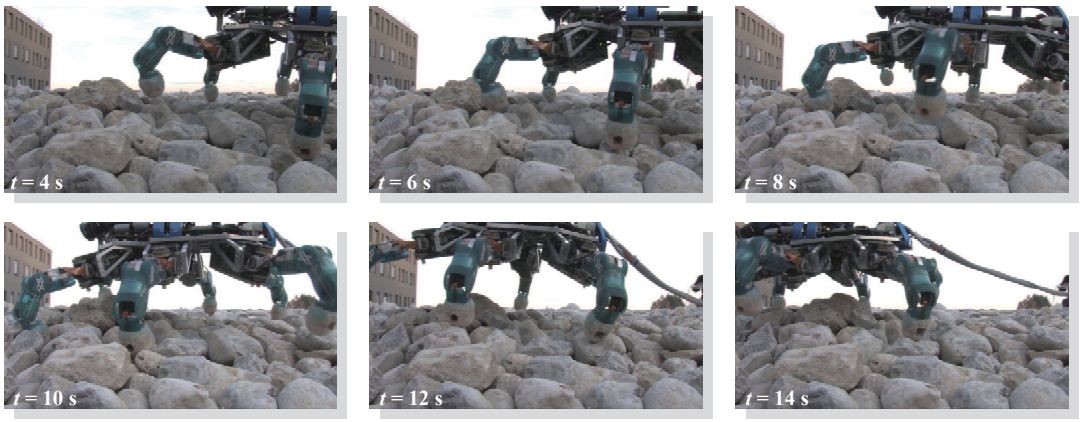


Figure 4.26: The DLR Crawler crosses a gravel field within the DLR outdoor testbed

Figure 4.27 presents the last series of images that visualizes an experimental trial. It demonstrates the ability of the robot to climb slopes that consist of mixed substrates, such as the combination of soil, gravel, and large stones. Using the gait and control algorithms developed within this work, the DLR Crawler easily handles such challenging conditions up to an inclination of approximately 35°. However, for walking along steeper slopes ($> 35°$) or slopes with very loose substrates, the robot should actively adjust its ground reaction forces to avoid slippage.

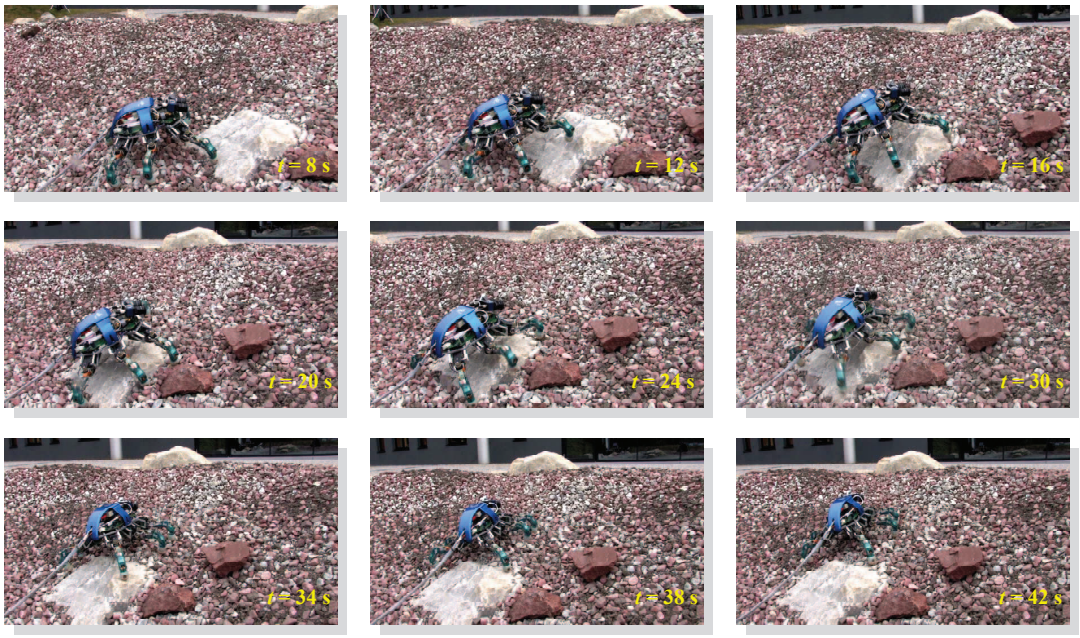


Figure 4.27: The DLR Crawler climbs a 30° slope within the DLR outdoor testbed

The next experiment with the DLR Crawler combines several different locomotion tasks while walking along the indoor test track shown in Fig. 4.28. First, the robot has to climb a short 15° slope onto a small platform. There, it slightly adjusts its position prior to turning 90° to the left. Next, it crosses a 5-cm-downward step and climbs another short 15° slope until it reaches a second platform where it stops.

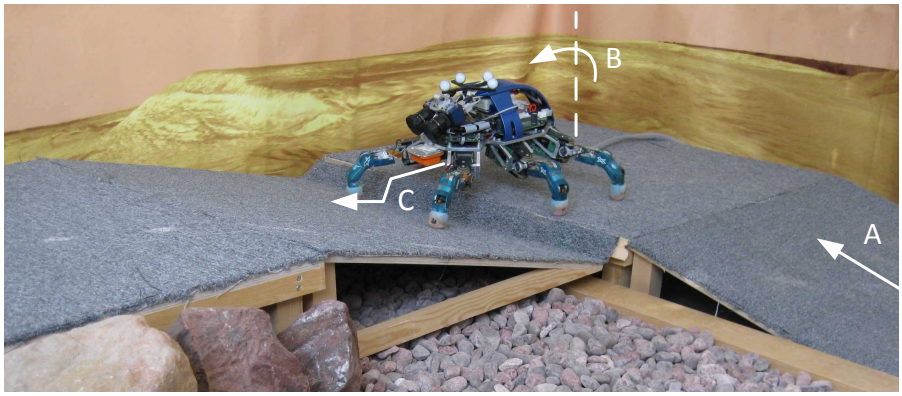
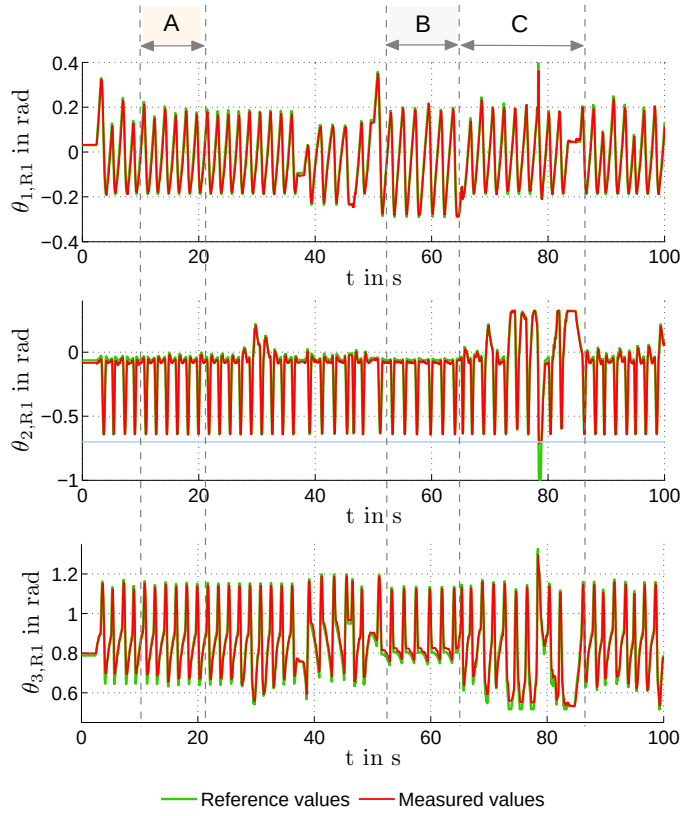


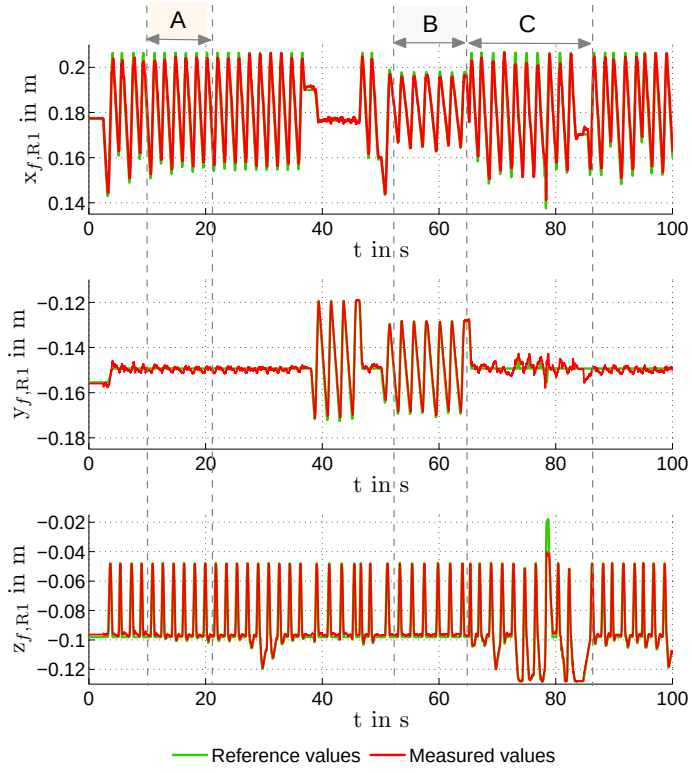
Figure 4.28: The DLR Crawler walks along a test track which comprises two slopes and a 5-cm-downward step

Along this track, there are several challenging passages that require spatio-temporal adaptation of the gait, i. e. the execution of reflexes as well as phase adjustments of the coordination. Those passages are, for example, the negative and positive edges at the beginning and the end of each slope as well as the 5-cm-downward step. For each case, height differences have to be accommodated as well as early or late ground contacts. To exemplify the leg activity along this track, Fig. 4.29 displays the joint trajectories of the right front leg (R1) as well as the Cartesian trajectories of the respective foot. The time intervals “A”, “B” and “C”, marked within those diagrams, correspond to the three equally labelled passages shown in Fig. 4.28, i. e. walking along the slope, turning to the left, and crossing the downward step. Furthermore, the z-trajectory of the foot is shown in greater detail for those intervals in Fig. 4.30, which also presents the related Cartesian foot trajectories with respect to the body frame.

As shown in Fig. 4.29(a) the overall tracking performance of the joint compliance controller is good. The largest errors occur for the third degree of freedom, which comprises the coupled medial and distal joints. In addition, a large error is apparent for the second degree of freedom in interval “C”. However, this does not result from poor controller performance. Instead, a reflexive elevation of the leg causes the desired joint angle to exceed a protective, software-set limit that is indicated by the light blue line in Fig. 4.29(a). At this limit, the software saturates the joint angle command, which is apparent for the measured value depicted within the same diagram.



(a) Joint trajectories of the right front leg (R1)



(b) Foot trajectories of the right front leg (R1) in body frame

Figure 4.29: The DLR Crawler walks along a test track - trajectories of leg R1

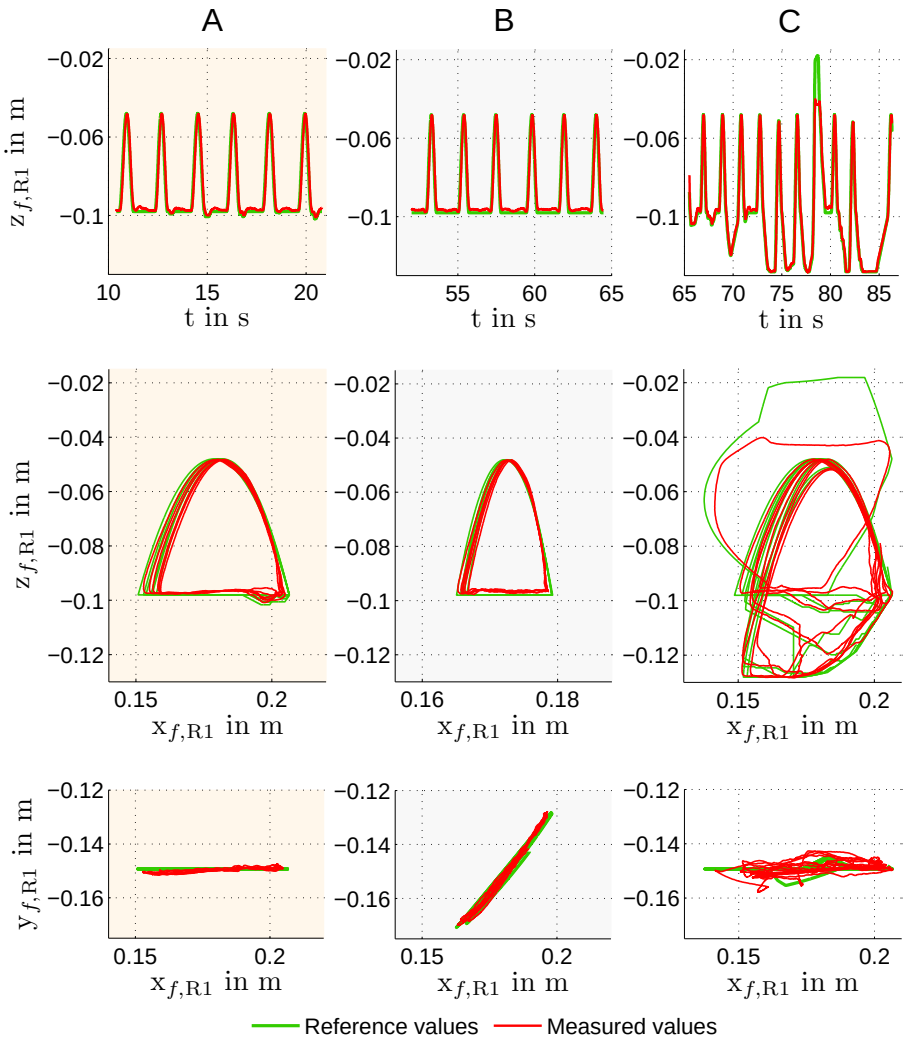
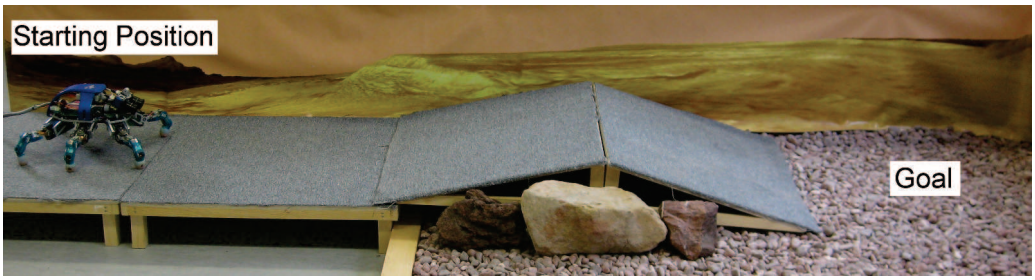


Figure 4.30: The DLR Crawler walks along a test track - foot trajectory of leg R1 with respect to the body coordinate system at specific time intervals (A, B, C): top row - z-position of the foot with respect to time; middle row - xz-trajectory of the foot; bottom row - xy-trajectory of the foot

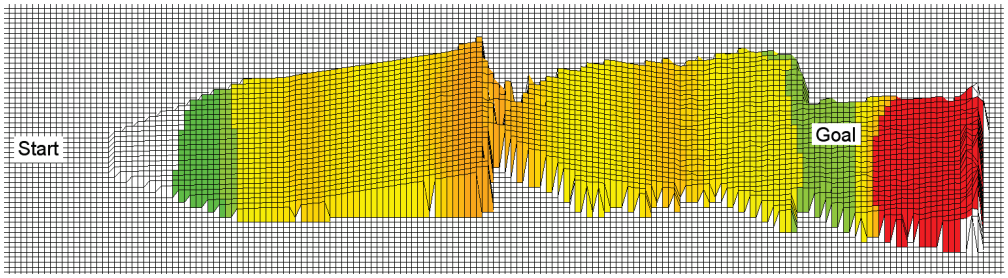
In Fig. 4.29(b), the first 20 seconds of the x-trajectory of the foot clearly indicate temporal adaptations of the leg cycle. Those have two main reasons: starting from rest and crossing the negative edge of the first slope. Nevertheless, during this time, the leg does not invoke the elevator reflex, as it contacts the ground late in its swing phase when the reflex is already inactive. Along the slope, the leg always extends to its nominal length which pushes the front of the robot upwards to follow the inclination of the terrain. In addition, the leg executes its stretch reflex for a short period following the swing-to-stance transition in order to achieve sufficient loading. This is clearly shown by the xz-trajectory of the foot displayed in the first column of Fig. 4.30. Around the time of 30 seconds, the robot transitions from the first slope onto the first platform. Crossing this positive edge invokes the stretch reflex multiple times, which is evident from the bottom row of Fig. 4.29(b). Once the robot is completely on top of the first platform, it moves sideways. Then it turns 90° to the left as shown within the time interval labelled "B". During the next time interval, labelled "C", the robot crosses the downward step that causes most of the reflex activity within this experiment. As shown in the third column of Fig. 4.30, the stretch reflex is activated several times. Each time, this reflex almost fully extends the leg to establish sufficient ground contact. In contrast, the elevator reflex is triggered only once. In this case, the leg retracts and raises to recover from a collision with the second slope early in its swing phase. As previously discussed, the commanded reflex step height of the foot is truncated due to the software-set limit of the second degree of freedom. However, this has no negative consequences and the leg returns to its regular cycle afterwards. The trajectories of all other legs are omitted in this place, as they show similar results.

The next two experiments exemplify the autonomous operation of the DLR Crawler in a priori unknown terrain, and thus the interaction of its walking layer with the stereo-vision-based navigation framework introduced in [Stelzer et al., 2012]. Within the first experiment, the DLR Crawler changes its gait generation method in dependence of the terrain. On flat ground without obstacles it utilizes a very simple, preprogrammed gait pattern but switches to the adaptive gait on challenging terrain. The reason for implementing such a behaviour is to save or redistribute computational resources whenever they are not required for locomotion. Of course, this is not important for the cable-bound DLR Crawler but of great relevance for future power autonomous systems with on-board computation. Figure 4.31 shows the setup of the experiment. The test track therein comprises two flat sections with different substrates as well as a bump that provides negative edges at its bottom and a positive edge at its top. As for the previous example, passing such edges requires spatio-temporal adaptation of the gait, and thus a flexible gait coordination in combination with reflexes. To

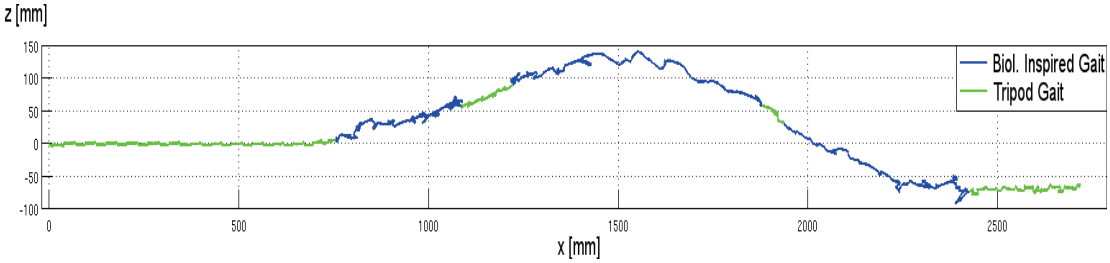
initiate the gait switch, a stereo-vision-based danger value is utilized. According to [Stelzer et al., 2012], this value is computed from slope, roughness and step height of the local terrain while an appropriate switching threshold is experimentally determined for each robot. Figure 4.31 shows that the DLR Crawler uses a preprogrammed tripod gait within the green and yellow sections of the test track while it switches to the adaptive gait for the more dangerous orange areas, i. e. the negative and positive edges of the bump.



(a) Test setup



(b) Traversibility map of the terrain

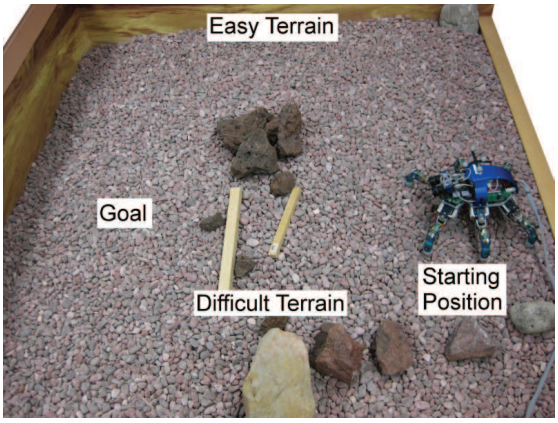


(c) xz-trajectory of the DLR Crawler

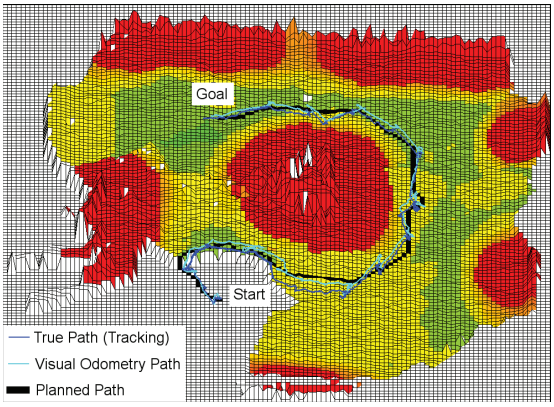
Figure 4.31: The DLR Crawler walks across a bump and switches its gait in dependence of the traversability of the terrain

The final experiment within this section demonstrates the adaptation of the path planning process to hardware- or task-related constraints. If the robot, for example, has to finish a time critical task, taking a short but risky path may be justified. The same is true, if the battery is low and the robot has to reach its base station within limited

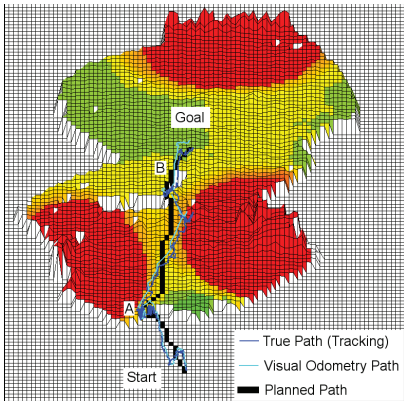
time. On the other hand, if the robot is damaged or carries large loads, the reduced mobility might require taking a longer but safer path. A simple way to adjust the path planning process is to change the danger value threshold that separates passable from impassable regions. Depending on the situation, this could be either done by high-level software such as mission planning, or by the walking layer of the robot. Figure 4.32 presents the setup of the experiment and two exemplary runs with different danger value thresholds. The diagram in Fig. 4.32(b) shows the trial with a low threshold. In this case, all grid cells with danger values higher than 0.2 are marked to be not traversable and the path planner chooses the longer but safer way. Furthermore, the robot uses the preplanned tripod gait due to the low danger value along its path. In the other case, shown in Fig. 4.32(c), the threshold is set to 1. Therefore, the path planner chooses the direct but more difficult way across the obstacles, which also induces a switch to the adaptive gait in between the points “A” and “B”.



(a) Test setup



(b) Traversibility map and path for $d_{\max} = 0.2$



(c) Traversibility map and path for $d_{\max} = 1$

Figure 4.32: The DLR Crawler navigates with different danger value thresholds $d_{\max}$

5 | Adaptation to Leg Loss

5.1 Introduction

Autonomous operation within scenarios such as extra-terrestrial exploration or terrestrial disaster response requires robotic systems to be highly robust. However, the robustness should not only be provided by the mechatronic hardware but also on the algorithmic level. Regarding hexapedal walking robots, the loss of a single leg is a likely source of error, which must not result into failure of the mission. Despite being impaired, a six-legged robot should be able to exploit its inherent redundancy in order to compensate for the loss and to proceed with its task. Several methods of varying complexity are reported in literature that adapt the gait of a six-legged robot to leg loss. Some of them simply use pre-computed gaits while others employ structural changes of the gait coordination network or learning-based approaches. [Ferrell, 1993], for example, reports on developments for the hexapod Hannibal that uses a damaged leg as a switchboard to transmit coordination signals to the next functional leg. In a similar way, the hexapod Oscar adapts to leg loss by removing a damaged leg from the active gait coordination network [El Sayed Auf et al., 2008]. Within a very recent example a learning algorithm retunes the oscillation frequencies of a synchronised network of multiple chaotic CPGs to account for a damaged leg [Ren et al., 2015]. The T-resilience algorithm [Koos et al., 2013], in contrast, combines a learning-based approach with self-modelling to cope with malfunctions of the legs. Therein, the robot uses the self-model, i. e. a dynamics model, to find new behaviours that do not use the damaged part, even though this is not known to the model. For this purpose, the algorithm chooses the behaviour that produces the least deviations in between the intact model and the impaired real system. A very interesting recent approach is presented in [Cully et al., 2015]. Herein, the robot pre-computes a behaviour-performance map that stores a large number of effective motion behaviours. In case of some damage, the robot applies a fast trial-and-error search based on this map and selects an appropriate motion for the specific error case. Also incorporating dynamic behaviours this method

Table 5.1: Mechanism weights for the selectively activated inter-leg couplings

Parameter	Value	Parameter	Value	Parameter	Value
$k_{1,\mathrm{L3L1}}$[mm]	5	$k_{2,\mathrm{L3L1}}$[mm]	5	$k_{2,\mathrm{R3R1}}$[mm]	5
$k_{1,\mathrm{R3R1}}$[mm]	5	$k_{2,\mathrm{L2R1}}$[mm]	10	$k_{2,\mathrm{R1L2}}$[mm]	5
$k_{2,\mathrm{L1R2}}$[mm]	5	$k_{2,\mathrm{R2L1}}$[mm]	10	$k_{2,\mathrm{L3R2}}$[mm]	10
$k_{2,\mathrm{R2L3}}$[mm]	10	$k_{2,\mathrm{L2R3}}$[mm]	10	$k_{2,\mathrm{R3L2}}$[mm]	10

is highly successful in generating effective locomotion. However, the approach does currently not consider the induced loads that might lead to further damage.

Returning to the approach for adaptation to leg loss presented within this thesis, the chapter proceeds as follows. First, the gait coordination algorithm introduced in Chapter 4 is extended in order to generate stable gait patterns for a five-legged robot. Next, a simple method is developed that increases the stability margin of the polygon of support, and thus smooths the motion of the impaired robot. Finally, the chapter concludes with the presentation of simulations and experimental results for the DLR Crawler being subject to leg loss.

5.2 Extended Gait Coordination

To enable proper adaptation to leg loss, the gait coordination algorithm introduced in Section 4.2 is extended by adding selectively activated inter-leg couplings, as shown in Fig. 5.1, and the associated mechanism weights given in Table 5.1. This assignment of new neighbourhood relations is the key to the immediate adaptation of the gait. While those additional inter-leg couplings remain inactive for the fully operational robot, they get instantaneously activated once a leg is labelled “damaged”, and thus removed from the network, as shown in Fig. 5.2. An important difference of these new couplings is the exclusion of mechanism 3, which did not show any beneficial influence on the resulting gaits in this case. Furthermore, only the loss of single legs is considered, while the approach also applies to the loss of two.

In addition to the immediate adaptation of the gait coordination, the emerging gait pattern should also preserve its beneficial properties such as forward-directed waves of protractions. The gait diagrams presented in Fig. 5.3 clearly demonstrate the effectiveness of the algorithmic extension in achieving those requirements. Since the results are similar for both sides, the figure only presents gait diagrams related to the loss of a left-side leg while those for the loss of a right-side leg are omitted

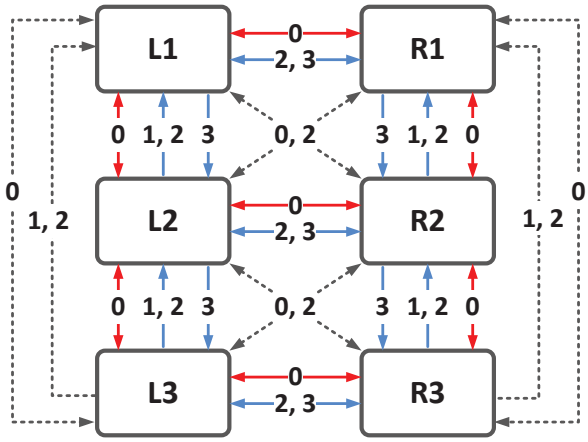
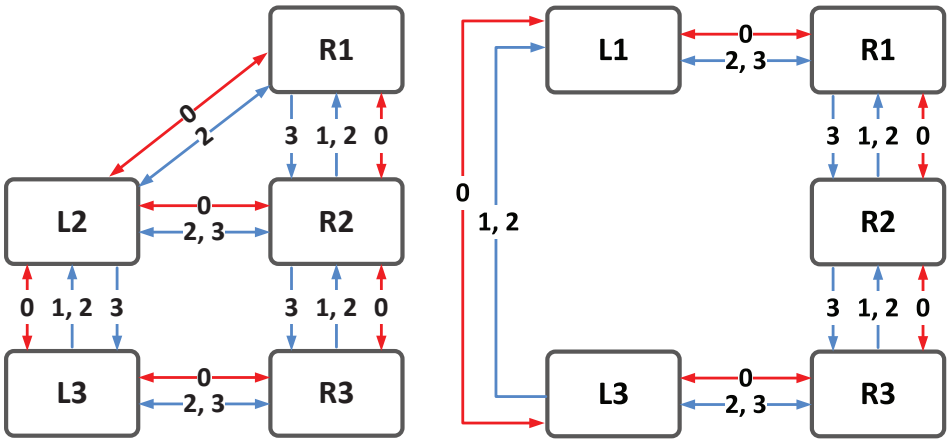


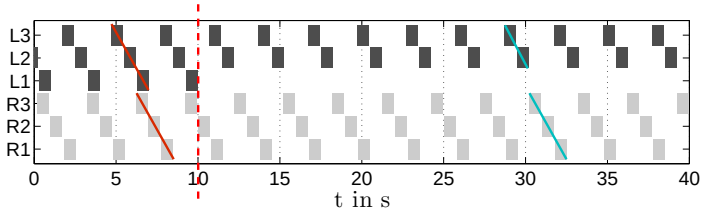
Figure 5.1: Extended coordination rules to adapt to leg loss/damage



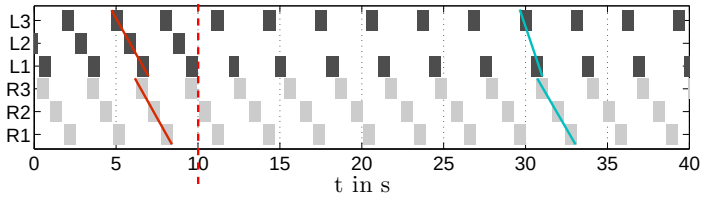
(a) (b)

Figure 5.2: Extended coordination rules: a) left front leg (L1) disabled, b) left middle leg (L2) disabled

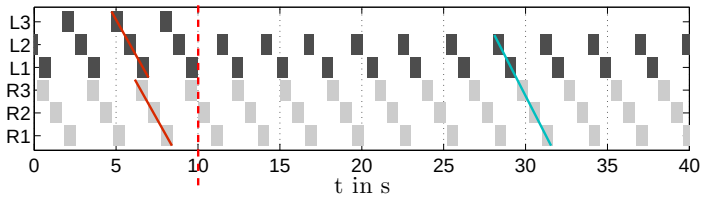
in this place. As desired, each of the diagrams shows immediate adaptation of the gait pattern and sustained forward-directed waves of protractions. These go along with changing phase relations that are established during several step cycles after the leg-loss incident. This behaviour is clearly apparent in Fig. 5.3(b), which shows a decreasing time gap in between the steps of the left hind leg and the left front leg.



(a) L1 disabled at $t = 10\,\mathrm{s}$



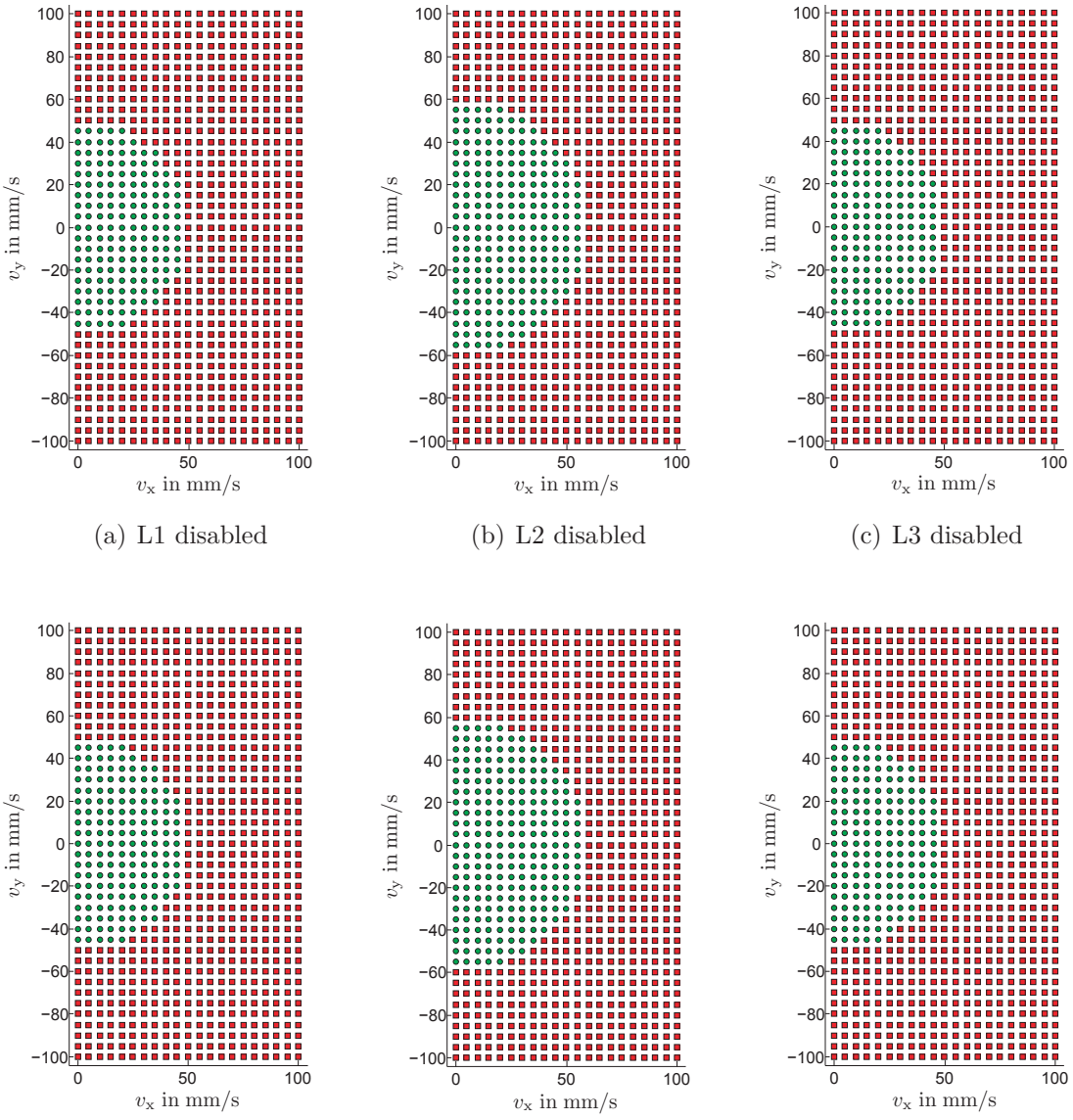
(b) L2 disabled at $t = 10\,\mathrm{s}$



(c) L3 disabled at $t = 10\,\mathrm{s}$

Figure 5.3: Gait diagrams of the DLR Crawler walking forward with single disabled legs at $v_x = 30\,\mathrm{mm/s}$ (white: stance phase; dark/light grey: swing phase left/right side)

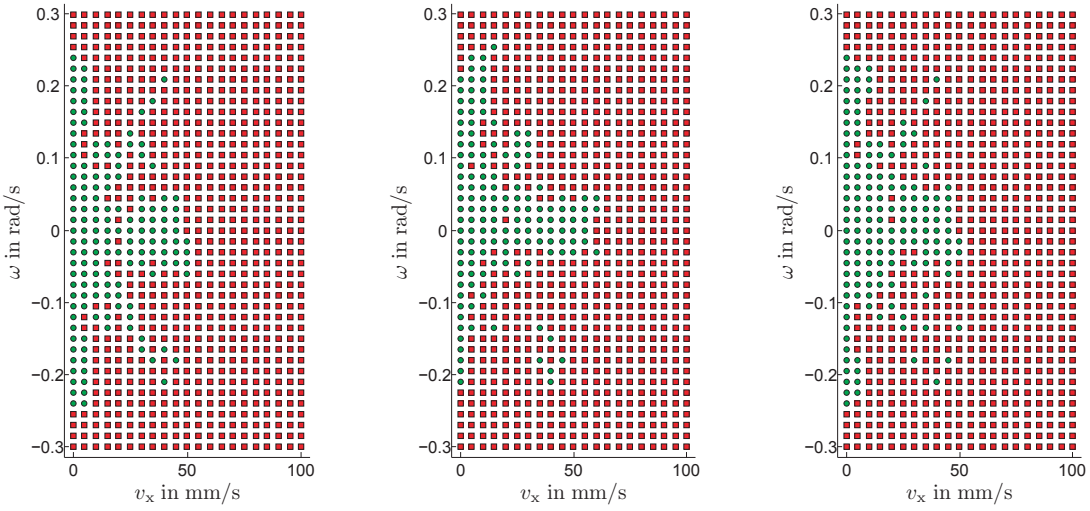
In order to assess the feasible range of velocity commands that produce well coordinated gaits for the five-legged robot, a large number of kinematic simulations has been performed. The resulting data for combined forward and sideways walking as well as for curve walking is presented in Fig. 5.4 and Fig. 5.5, respectively. In comparison to the fully functional six-legged configuration, the loss of a front or hind leg reduces the maximum velocity for straight-line walking approximately to one half, while the loss of a middle leg only results in reductions of one third. Furthermore, the cluster for curve walking with radii above 2 m is reduced to smaller speeds by 40 to 50 % while turning on the spot is less impaired. Based on these results, a higher-level planning algorithm is now able to adjust its commands to account for the changed capabilities of a five-legged robot.



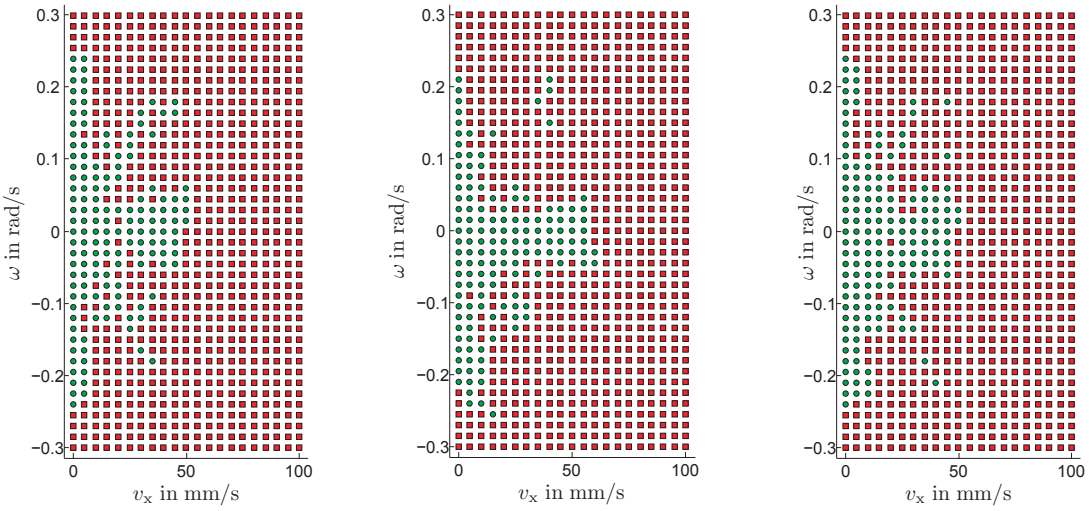
(a) L1 disabled (b) L2 disabled (c) L3 disabled

(d) R1 disabled (e) R2 disabled (f) R3 disabled

Figure 5.4: Stability assessment of the gait coordination of the DLR Crawler for various combinations of forward and lateral velocities in case of leg loss; green dots indicate stable gaits and red squares unstable gaits



(a) L1 disabled (b) L2 disabled (c) L3 disabled



(d) R1 disabled (e) R2 disabled (f) R3 disabled

Figure 5.5: Stability assessment of the gait coordination of the DLR Crawler for various combinations of forward velocity and yaw rate in case of leg loss; green dots indicate stable gaits and red squares unstable gaits

5.3 Improving Stability and Performance

Knowing about the capability of the gait coordination to produce well-coordinated gaits in case of leg loss, there is another important issue to be considered. While the coordination algorithm guarantees that neighbouring legs do not step at the same time, there is no guarantee for static stability of the five-legged robot. This fact is visualized by Fig. 5.6 which shows strongly reduced or even negative stability margins for the impaired robot. Since such unstable configurations only occur for short periods of time, they do not necessarily result in falls but cause undesirable rocking motions. In order to increase the static stability margin, this work introduces a very intuitive solution to the problem. As shown in Fig. 5.7, the projected center of gravity is simply shifted away from the damaged leg by adding a fixed offset to the desired position of each foot. Based on a set of kinematic simulations, Fig. 5.8 proves the effectiveness of this approach for the DLR Crawler. The diagrams therein compare the cyclic minima of the static stability margin in case of leg loss with and without COG shift applied. Following the loss of a left-side leg, the COG is shifted 20 mm to the right and additionally 20 mm along the longitudinal axis if the loss concerns a front or a hind leg. Obviously, these adjustments are sufficient to produce positive minima of the static stability margin for most of the feasible walking velocities.

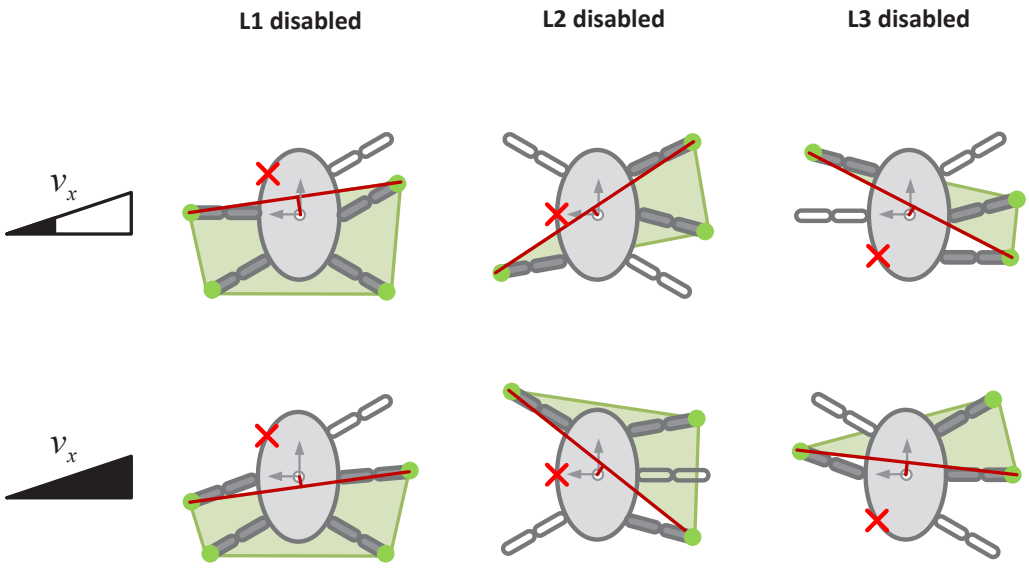
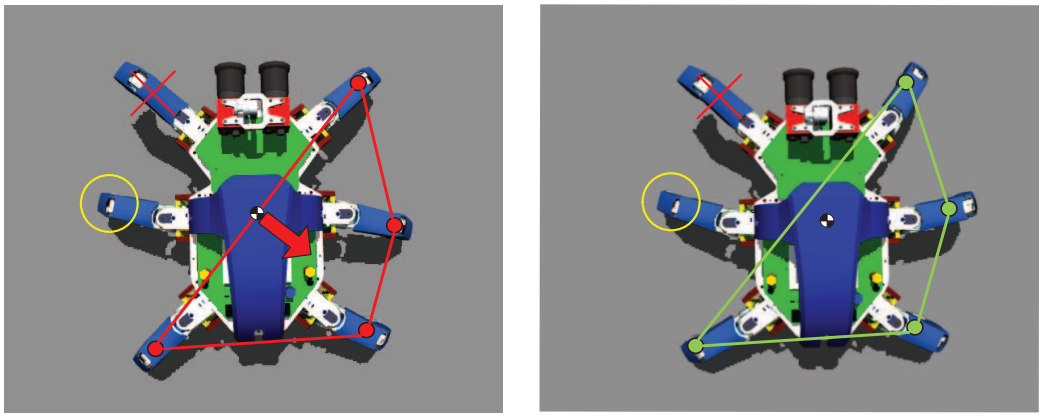


Figure 5.6: The sketches show the minimal stability margins and the respective configurations for different leg-loss scenarios and forward velocities; the red cross marks the damaged leg and the green area the polygon of support



(a) Stability margin **without** COG shift (b) Stability margin **with** COG shift

Figure 5.7: The DLR Crawler increases its stability margin by shifting its COG into the polygon of support, i. e. away from the damaged left front leg (L1); the red cross marks the damaged leg and the yellow circle a stepping leg

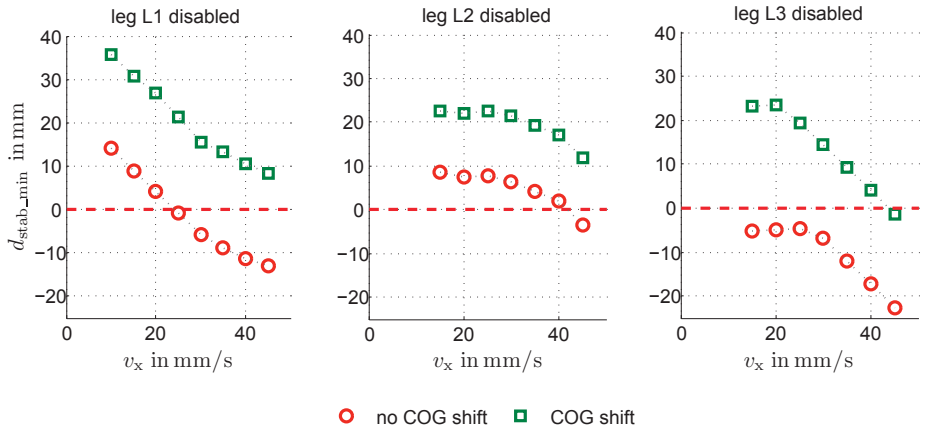
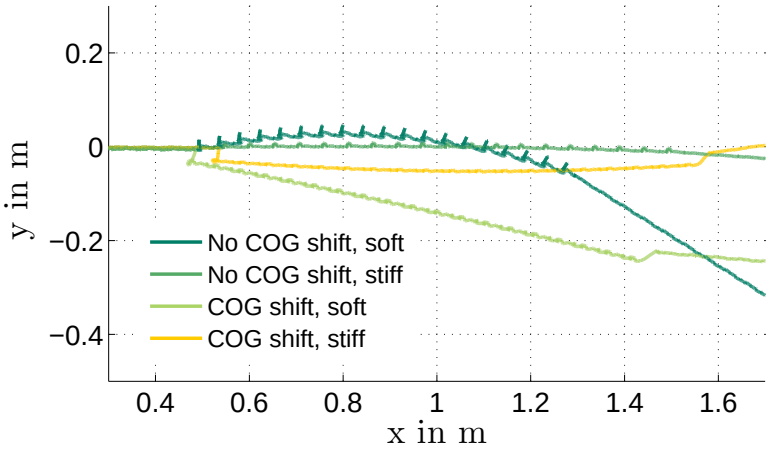


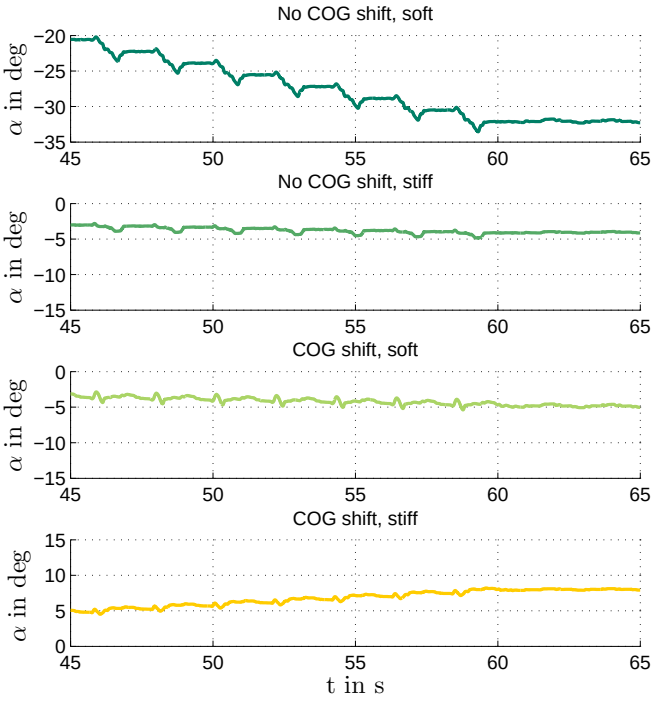
Figure 5.8: Comparison of the stability margin minima of the five-legged DLR Crawler while walking at different forward velocities with and without COG shift

Based on kinematics only, those simulations do not yet give the full picture. For this reason, additional simulations of the dynamics were performed that employ the simplified model introduced in Section 3.5 and the set of parameters given in Appendix A. The results provide an overview of the actual locomotive performance of the impaired robot, i. e. the rocking motions of the body and the yaw angle drift caused by unbalanced propulsive forces. In order to improve this performance, different measures were tested in addition to or in combination with the proposed COG shift. At first, the step size of the robot was reduced by changing the nominal AEP and PEP radii from 40 mm to 30 mm each. This adjustment already decreased the rocking motions of the robot body, while it had negligible influence on the gait coordination stability. Next, the joint stiffness was doubled, which significantly reduced the pitch and roll angle oscillations. Furthermore, it turned out that some of the reflexes have a deteriorating influence in case of leg loss. To give an example, the loss of a front leg causes the robot to tilt towards this lost leg. Therefore, the hind legs unload and trigger their stretch reflex. The resulting extension of the hind legs then further increases the downward pitch of the body. Being cyclic, this behaviour induces strong pitch and roll oscillations. To investigate such reflex influences on the overall motion of the impaired robot, 27 different sets of simulations were run. These sets comprise walking without a left front, middle, or hind leg at three different velocities with three different reflex settings: all reflexes disabled, all reflexes enabled, and the stretch reflex enabled only. Furthermore, each of these simulation sets includes four individual runs with different settings of the joint stiffness as well as the applied COG shift. This amounts to a total number of 108 simulations. Within the 80 seconds of each run the "lost" left-side leg was disabled at $t = 20\,s$ and recovered its function $t = 60\,s$.

Next, this section presents the results of one exemplary set of simulations in greater detail. Having lost its left front leg, the robot walks with all reflexes disabled at a velocity of 30 mm/s. At first, Fig. 5.9(a) compares the xy-trajectories of four different runs with respect to the influence of COG shift and leg compliance. In each case, the robot experiences a lateral drift which is most severe for walking at low joint stiffness settings. Without COG shift, this lateral motion results from the yaw angle drift shown in the top diagram of Fig. 5.9(b). With COG shift, the lateral drift mainly stems from an imbalance of propulsive forces. Nevertheless, doubling the joint stiffness by control strongly reduces the lateral drift as well as the amplitude of pitch and roll angle oscillations. For the present example, this effect does not depend on a COG shift, as shown Fig. 5.10(a) and Fig. 5.10(b). Therefore, the increased joint stiffness is the most important adjustment to improve the locomotive performance of a six-legged robot suffering the loss of a front leg.

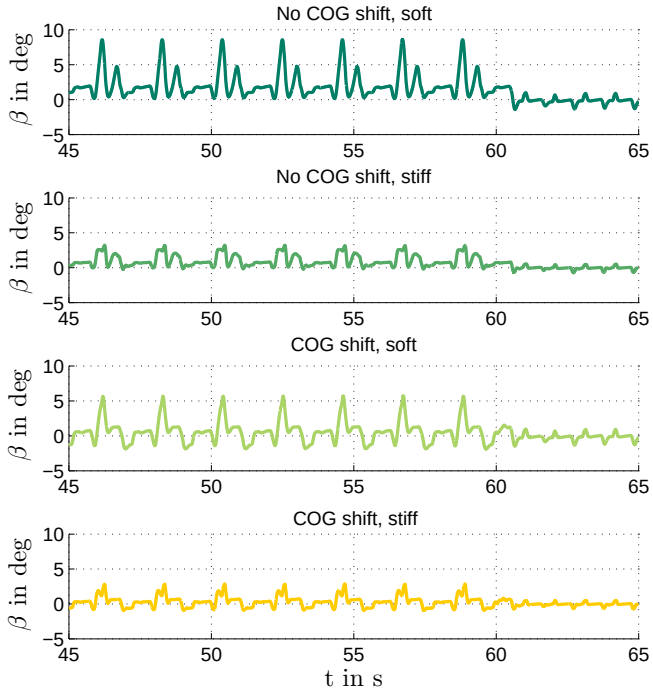


(a) xy-trajectories

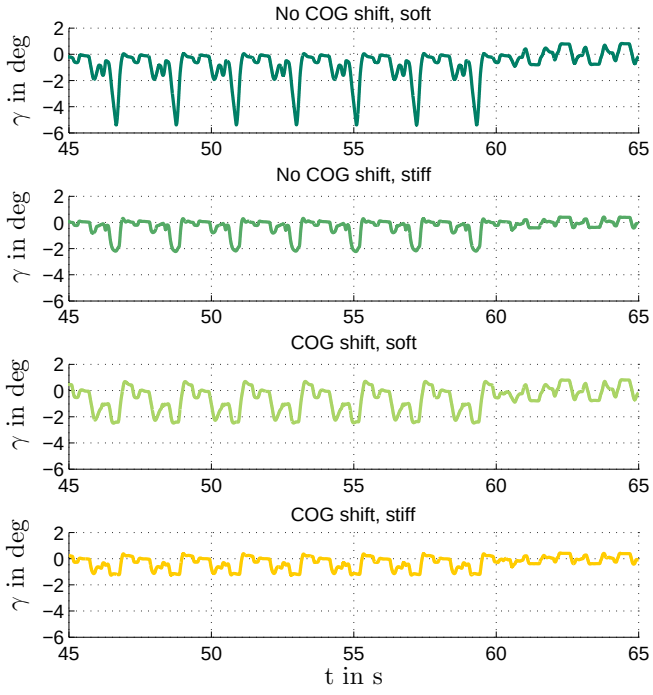


(b) Yaw angle

Figure 5.9: xy- and yaw angle trajectories of the simulated DLR Crawler for different stiffness and COG shift settings: all reflexes disabled, $v_x = 30\,\mathrm{mm/s}$, loss of the left front leg (L1) in between $t = 20\,\mathrm{s}$ and $t = 60\,\mathrm{s}$



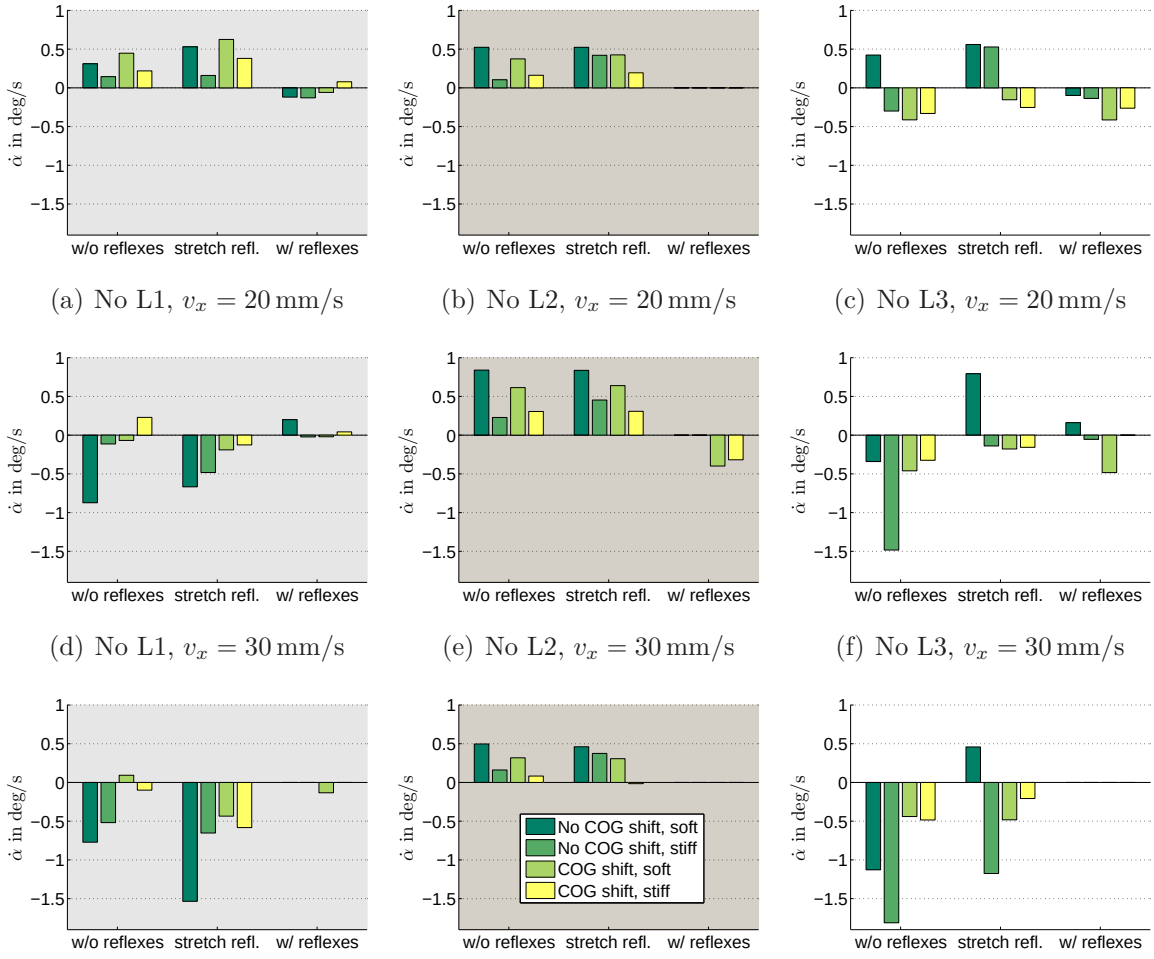
(a) Pitch angle



(b) Roll angle

Figure 5.10: Pitch and roll angle trajectories of the simulated DLR Crawler for different stiffness and COG shift settings: all reflexes disabled, $v_x = 30\,\mathrm{mm/s}$, loss of the left front leg (L1) in between $t = 20\,\mathrm{s}$ and $t = 60\,\mathrm{s}$

Following this detailed example, Fig. 5.11, Fig. 5.12 and Fig. 5.13 summarize the results of the 108 individual simulation runs. First, Fig. 5.11 compares the drift rate of the yaw angle at three forward walking velocities that is caused by the loss of a single left-side leg. Within each of these diagrams four different combinations of COG shift and leg compliance are evaluated for a robot that has either all reflexes enabled, all reflexes disabled, or the stretch reflex enabled only.



(a) No L1, $v_x = 20\,\mathrm{mm/s}$ (b) No L2, $v_x = 20\,\mathrm{mm/s}$ (c) No L3, $v_x = 20\,\mathrm{mm/s}$

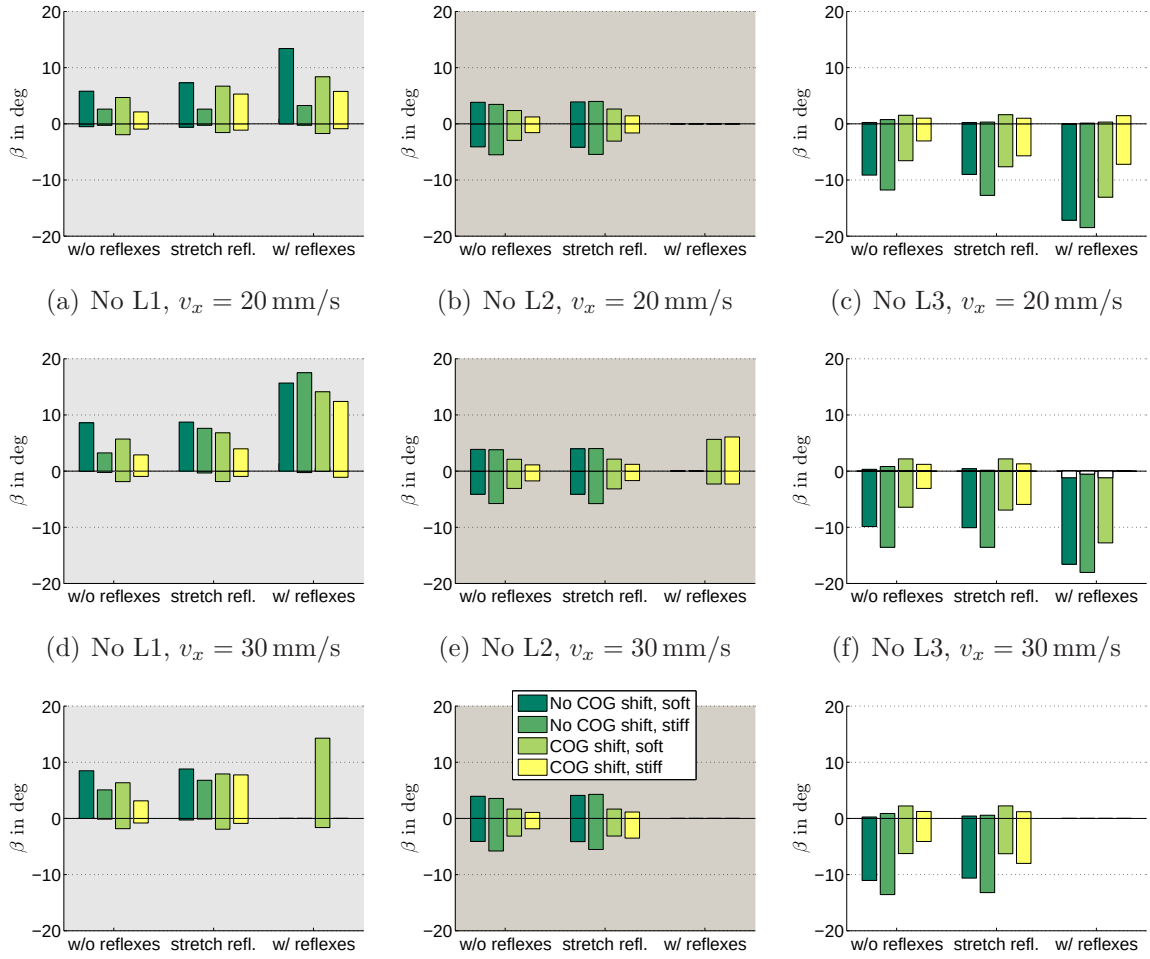
(d) No L1, $v_x = 30\,\mathrm{mm/s}$ (e) No L2, $v_x = 30\,\mathrm{mm/s}$ (f) No L3, $v_x = 30\,\mathrm{mm/s}$

(g) No L1, $v_x = 40\,\mathrm{mm/s}$ (h) No L2, $v_x = 40\,\mathrm{mm/s}$ (i) No L3, $v_x = 40\,\mathrm{mm/s}$

Figure 5.11: Leg-loss-induced yaw angle drift rate of the simulated DLR Crawler while walking at different velocities

Next, Fig. 5.12 and Fig. 5.13 follow a similar scheme in comparing the minima and maxima of the leg-loss-induced pitch and roll oscillations. For some of the configurations, the diagrams show no result. This is either caused by failing simulations or due to the robot getting stuck within a reflex loop. Obviously, the trials with all reflexes enabled most often failed. Especially, the loss of a middle leg caused strong pitch-

ing and rolling motions provoking repeated ground collisions of the adjacent front or hind legs. While stepping, these legs then repeatedly triggered their elevator reflex and got caught in reflex loops. To resolve this issue, different reflexes were disabled. In combination with an increased joint stiffness this approach yields the best overall performance after the loss of a front leg. In contrast, disabling reflexes and shifting the COG is the method of choice to reduce the pitch and roll oscillations induced by the loss of a middle or hind leg.



(a) No L1, $v_x = 20\,\mathrm{mm/s}$ (b) No L2, $v_x = 20\,\mathrm{mm/s}$ (c) No L3, $v_x = 20\,\mathrm{mm/s}$

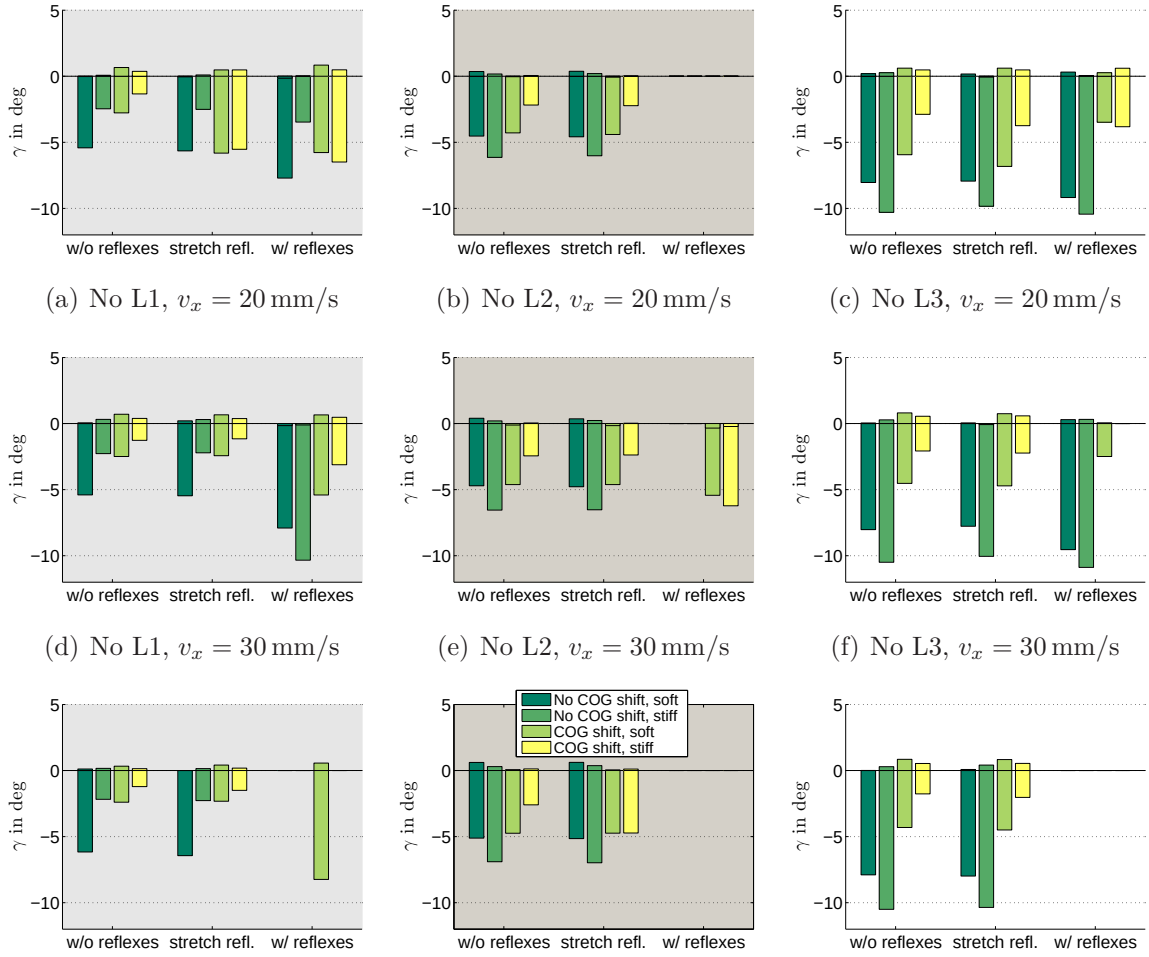
(d) No L1, $v_x = 30\,\mathrm{mm/s}$ (e) No L2, $v_x = 30\,\mathrm{mm/s}$ (f) No L3, $v_x = 30\,\mathrm{mm/s}$

(g) No L1, $v_x = 40\,\mathrm{mm/s}$ (h) No L2, $v_x = 40\,\mathrm{mm/s}$ (i) No L3, $v_x = 40\,\mathrm{mm/s}$

Figure 5.12: Leg-loss-induced cyclic pitch angle minima and maxima of the simulated DLR Crawler while walking at different velocities

In summary, disabling reflexes, increasing the joint stiffness, and simultaneously shifting the COG of the impaired robot always reduces the pitch and roll oscillations as well as the associated yaw angle drift. Often, but not always, this combination also yields the best result. Nevertheless, in uneven terrain it might not be possible to

turn off the reflexes. In this case, a higher-level algorithm should limit the velocity command following the loss of a front or hind leg while keeping the reflexes enabled. Since this approach does not apply to the loss of a middle leg, as shown in Fig. 5.11, Fig. 5.12 and Fig. 5.13, a high-level strategy has to be developed to resolve situations that require the full reflex capabilities.



(a) No L1, $v_x = 20\,\mathrm{mm/s}$ (b) No L2, $v_x = 20\,\mathrm{mm/s}$ (c) No L3, $v_x = 20\,\mathrm{mm/s}$

(d) No L1, $v_x = 30\,\mathrm{mm/s}$ (e) No L2, $v_x = 30\,\mathrm{mm/s}$ (f) No L3, $v_x = 30\,\mathrm{mm/s}$

(g) No L1, $v_x = 40\,\mathrm{mm/s}$ (h) No L2, $v_x = 40\,\mathrm{mm/s}$ (i) No L3, $v_x = 40\,\mathrm{mm/s}$

Figure 5.13: Leg-loss-induced cyclic roll angle minima and maxima of the simulated DLR Crawler while walking at different velocities

5.4 Simulations and Experiments

To complete the chapter, this section presents a visual impression of the results. For this purpose, various image series were extracted from video recordings of simulation runs and experimental trials. At first, Fig. 5.14 visualises the simulation results of the

DLR Crawler experiencing the loss of its left front leg. At t = 4 s this leg is disabled by control and the foot is lifted to a resting position. Therein it stays throughout the remainder of the trial. Following this "loss" event, the gait coordination instantaneously adapts to the new five-legged configuration. In the meantime, the walking layer enhances the stability as it initiates a COG shift, stiffens the joints and disables the reflexes of the robot. Thus, after an initial downward pitch of its frontal part, the robot quickly recovers its stable and smooth locomotion. Similar results are obtained for the loss of the left middle and the left hind leg, as shown in Appendix C

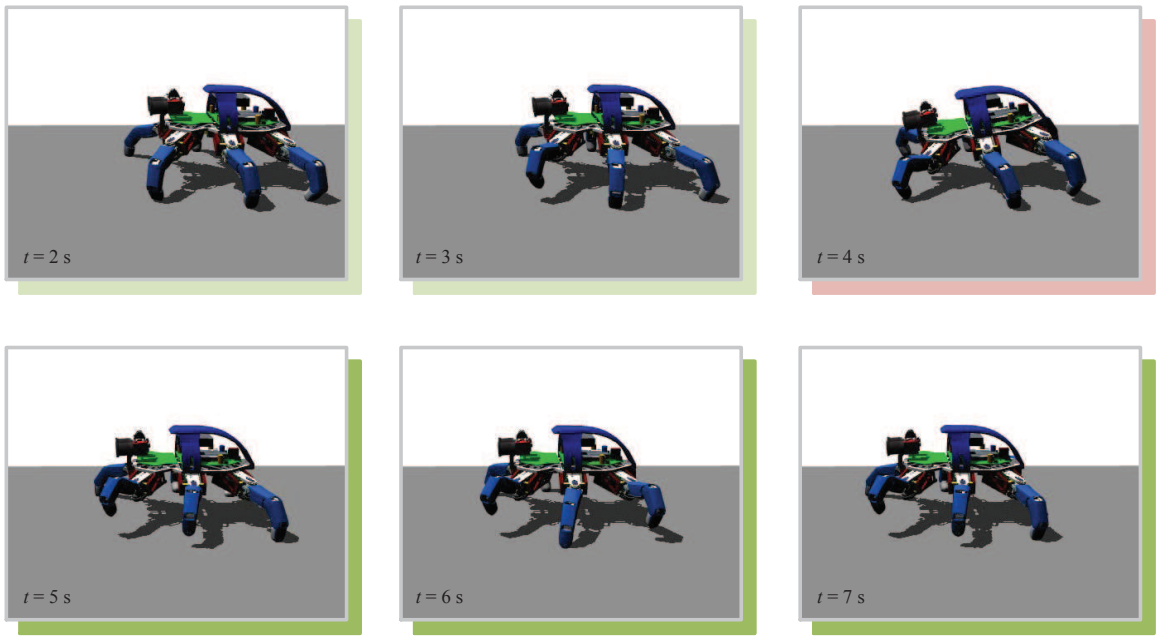
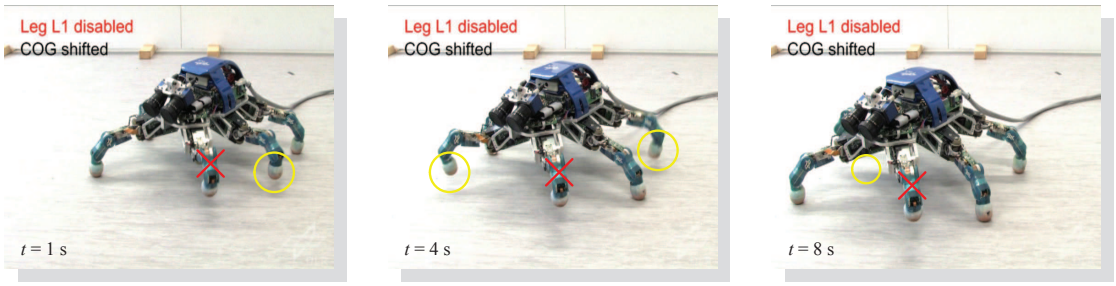
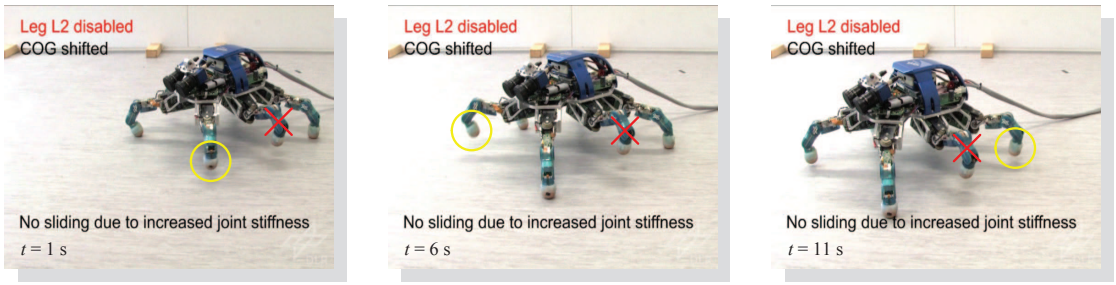


Figure 5.14: Simulation of the DLR Crawler experiencing the loss of its left front leg (L1); light green frames: the robot is fully functional, red frame: the left front leg is lost (i. e. lifted and hold in this position), green frames: the robot instantaneously adapts its gait, shifts its COG away from L1, and increases its joint stiffness settings

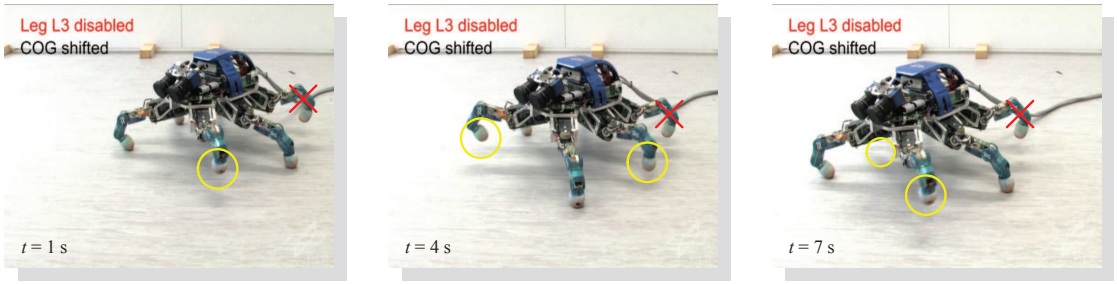
Of course, the presented gait adaptation scheme also works with the real system which is shown in Fig. 5.15. The scenario is equal to the simulation shown above. The robot experiences the loss of a left-side leg and instantaneously adapts its gait. In addition, it increases its joints stiffness, shifts its COG, and disables all of its reflexes. The results are similar to those of the simulation as the robot is able to smoothly walk with five legs.



(a) Failure of the left front leg (L1)



(b) Failure of the left middle leg (L2)



(c) Failure of the left hind leg (L3)

Figure 5.15: Experimental trials for the loss of a left front, middle, or hind leg of the DLR Crawler; red crosses indicate lost/damaged legs, yellow circles indicate stepping legs

6 | Leg Odometry

6.1 Introduction

Within future exploration scenarios, legged robots will serve as the rough terrain specialists for short range tasks. Having neither a prior knowledge of the environment nor reliable absolute references, like GPS or an external magnetic field, they have to collect all of the relevant information by themselves. For this reason, they need to exploit the measurements of their on-board sensors to a maximum extent and should not discard any useful information. Very important to exploration is the ability of each robot to localise itself with respect to its environment. This means, the robot has to know its current pose, i. e. the position and orientation of its body, in relation to a fixed world frame. However, as there is no absolute external reference available, it has to rely on relative methods, such as dead reckoning, to compute a pose estimate from IMU, visual or joint sensor data.

Therefore, this chapter presents the development and test of a leg-proprioception-based odometry algorithm for statically stable, torque-controlled walking robots. Applicable to four-, six- or eight-legged systems, the algorithm returns a full 6-DOF pose estimate. Thus, the leg odometry constitutes a "pose sensor" that is solely based on proprioceptive data from the legs, i. e. joint angles and joint torques. The key feature of the algorithm is the stabilisation of a kinematics-based 6-DOF estimate with absolute pitch and roll angles that are computed from joint torque measurements. The chapter proceeds as follows. Section 6.2 introduces the overall method and describes the three individual stages of the algorithm. Thereafter, Section 6.3 presents experimental results for the DLR Crawler and evaluates the performance of the leg odometry algorithm with respect to walking on different substrates, at different velocities, and with different joint stiffness settings. In the end of this chapter some experimental results demonstrate the successful integration of the leg odometry within a multisensor data fusion scheme for robust pose estimation [Chilian et al., 2011], which is an integral part of the visual navigation framework of the DLR Crawler [Stelzer et al., 2012].

6.2 Method

In comparison to any wheel odometry, the computation of a 6D leg odometry for multi-legged robots is a complex task. It has to consider a large number of sensors, which often results in an over-constrained problem. To cope with this challenge, the presented method borrows from computer vision. It treats the supporting feet at two consecutive time steps as point clouds and tries to match them by finding a minimizing transformation. Referenced to a body-fixed coordinate system, this transformation encodes the relative change of the robot pose within the time interval. Summing up all relative motions then results in a full pose estimate for the robot with respect to its initial body frame. To obtain proper results, the algorithm requires at least three non-collinear feet in contact with the ground. Furthermore, it assumes rigid point clouds, which implies a no slip condition. Hence slipping legs have to be detected and the related measurements have to be treated adequately. However, as it is true for any dead reckoning method, small errors accumulate over time. In the present case, the pitch and roll angle estimates are most susceptible to errors which propagate to severe deviations of the overall position estimate. Thus, these two angle estimates have to be stabilised by some additional measurements. For this purpose, an error-state Kalman filter fuses the kinematics-based pitch and roll angles with two absolute values. Those are derived from joint torque measurements based on the assumption that the main loads at slow speeds result from gravitational forces. As modern walking robots increasingly employ joint torque sensors for control reasons [Ott et al., 2011], [Boaventura et al., 2012], they do not involve any additional hardware effort just for the odometry.

6.2.1 Joint-Angles-Based Pose Estimate

The first stage of the algorithm is purely based on kinematics and calculates the relative change of the robot pose from two consecutive stance feet configurations. The resulting incremental motions are summed up over time, yielding the pose of the robot with respect to the world frame. This so-called world frame is defined to be the gravity-aligned body frame of the robot at its start position. The body frame, in turn, is oriented as follows. The positive x-direction points from the back to the front of the robot, while the positive z-direction points upwards from its bottom to its top. The positive y-direction completes a right hand system and the origin is placed at the center of the robot body. All rotations follow the xyz-convention with yaw angle α defined about the z-axis, pitch angle β about the y-axis and roll angle γ about the x-axis.

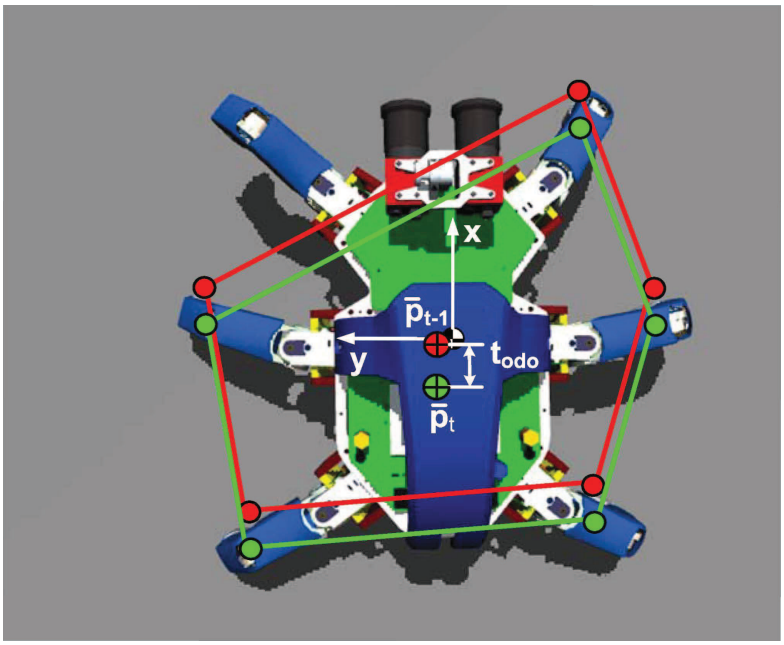


Figure 6.1: Foot contacts and their centroids for two consecutive time steps with respect to the body frame

For proper operation, the first stage of the algorithm requires at least three non-collinear legs in contact with the ground. This condition is true as long as the robot stays within a statically stable regime. Having more than three legs in contact with the ground, the system is over-constraint and the odometry algorithm has to account for this situation. Furthermore, due to rolling ground contacts of the feet and small errors within the kinematics, it is unlikely that the stance feet point clouds perfectly match. Thus, a rigid body transformation with the rotation matrix $\mathbf{R}_{\text{odo}}$ and the translation vector ${}^{b}\mathbf{t}_{\text{odo}}$ has to be found that minimizes the matching error. The detailed approach to solve this problem, including the complete derivation, was initially presented in a computer vision article [Haralick et al., 1989]. In the following, the necessary equations are given for the purpose of completeness. The algorithm aims to minimize ϵ, which is the sum of squared errors of the rigid body transformation augmented by constraints $f(\boldsymbol{\lambda}, \mathbf{R}_{\text{odo}})$ that enforce an orthogonal rotation matrix, $\mathbf{R}_{\text{odo}}\mathbf{R}_{\text{odo}}^{T} = \mathbf{I}$.

$$\epsilon = \sum_{i=1}^{n} w_i(\|{}^{b}\mathbf{p}_{i,t} - \mathbf{R}_{\text{odo}} \cdot {}^{b}\mathbf{p}_{i,t-1} - {}^{b}\mathbf{t}_{\text{odo}}\|_2)^2 + f(\boldsymbol{\lambda}, \mathbf{R}) \tag{6.1}$$

Herein, the vectors ${}^{b}\mathbf{p}_{i,t-1}$ and ${}^{b}\mathbf{p}_{i,t}$ are the positions of a foot at two consecutive time steps with respect to the body frame b. These foot positions are calculated from joint angle measurements using forward kinematics. The parameter n is the number of legs of the robot, which is usually 4, 6 or 8. The parameters w_i are weights that are 1 if

a foot is in a valid contact state or 0 otherwise. A valid contact state means that the foot has contact at both time steps and does not slip severely. The vector $\boldsymbol{\lambda}$ consists of Lagrange multipliers for the six constraints.

Taking the partial derivative of ϵ with respect to the translation vector ${}^{b}\mathbf{t}_{\text{odo}}$ and setting it equal to zero results into the following equation.

$$\begin{aligned} {}^{b}\mathbf{t}_{\text{odo}} &= {}^{b}\bar{\mathbf{p}}_{t} - \mathbf{R}_{\text{odo}}\,{}^{b}\bar{\mathbf{p}}_{t-1} \\ &= \frac{\sum_{i=1}^{n} w_i\,{}^{b}\mathbf{p}_{i,t}}{\sum_{i=1}^{n} w_i} - \mathbf{R}_{\text{odo}}\frac{\sum_{i=1}^{n} w_i\,{}^{b}\mathbf{p}_{i,t-1}}{\sum_{i=1}^{n} w_i} \end{aligned} \tag{6.2}$$

The terms ${}^{b}\bar{\mathbf{p}}_{t}$ and ${}^{b}\bar{\mathbf{p}}_{t-1}$ can be considered as the centroids of the contact point clouds at two consecutive time steps, as depicted in Fig. 6.1. Inserting ${}^{b}\mathbf{t}_{\text{odo}}$ into (6.1) leaves ϵ as a function of the elements of the rotation matrix and the Lagrange multipliers $\boldsymbol{\lambda}$. Taking now the partial derivative of ϵ with respect to each element of the rotation matrix, setting it equal to zero, $[\frac{\partial \epsilon}{\partial r_{\text{odo},(m,n)}}]_{3\times 3} = 0_{3\times 3}$, and rearranging terms results into the following equation.

$$\mathbf{A}\mathbf{R}_{\text{odo}}^{T} + \mathbf{R}_{\text{odo}}^{T}\boldsymbol{\Lambda} = \mathbf{B} \tag{6.3}$$

Herein, $\boldsymbol{\Lambda}$ is a symmetric matrix consisting of the six Lagrange multipliers and $\mathbf{A}$ as well as $\mathbf{B}$ are defined as follows.

$$\mathbf{A} = \sum_{i=1}^{n} w_i({}^{b}\bar{\mathbf{p}}_{t-1} - {}^{b}\mathbf{p}_{i,t-1})({}^{b}\bar{\mathbf{p}}_{t-1} - {}^{b}\mathbf{p}_{i,t-1})^{T} \tag{6.4}$$

$$\mathbf{B} = [\ \mathbf{b}_x \quad \mathbf{b}_y \quad \mathbf{b}_z\ ]_{3\text{x}3} \tag{6.5}$$

$$\mathbf{b}_x = \sum_{i=1}^{n} w_i({}^{b}\bar{p}_{x,t} - {}^{b}p_{i,x,t})({}^{b}\bar{\mathbf{p}}_{t-1} - {}^{b}\mathbf{p}_{i,t-1}) \tag{6.6}$$

$$\mathbf{b}_y = \sum_{i=1}^{n} w_i({}^{b}\bar{p}_{y,t} - {}^{b}p_{i,y,t})({}^{b}\bar{\mathbf{p}}_{t-1} - {}^{b}\mathbf{p}_{i,t-1}) \tag{6.7}$$

$$\mathbf{b}_z = \sum_{i=1}^{n} w_i({}^{b}\bar{p}_{z,t} - {}^{b}p_{i,z,t})({}^{b}\bar{\mathbf{p}}_{t-1} - {}^{b}\mathbf{p}_{i,t-1}) \tag{6.8}$$

Multiplying (6.3) with $\mathbf{R}_{\text{odo}}$ from the left leaves an equation with symmetric left- and right-hand sides since $\mathbf{A}$ and $\boldsymbol{\Lambda}$ are symmetric matrices.

$$\begin{aligned} \mathbf{R}_{\text{odo}}\mathbf{B} &= \mathbf{R}_{\text{odo}}\mathbf{A}\mathbf{R}_{\text{odo}}^{T} + \boldsymbol{\Lambda} \\ &= (\mathbf{R}_{\text{odo}}\mathbf{B})^{T} \end{aligned} \tag{6.9}$$

Thus, a singular value decomposition of $\mathbf{B}$ into the orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ and the diagonal matrix $\mathbf{D}$ allows the calculation of $\mathbf{R}_{\text{odo}}$, which can be verified by inserting (6.10) and (6.11) into (6.9).

$$\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{6.10}$$

$$\mathbf{R}_{\text{odo}} = \mathbf{V}\mathbf{U}^T \tag{6.11}$$

One important property of the solution for $\mathbf{R}_{\text{odo}}$ is that the rotation matrix is only valid if its determinant is positive. If this is not true, the last column of the matrix $\mathbf{V}$ has to be multiplied by -1 to deliver a valid result. Inserting $\mathbf{R}_{\text{odo}}$ into (6.2) gives the relative translation of the robot. Propagating the relative rotation and translation, the pose of the robot can be determined with respect to the world frame, i. e. relative to the gravity-aligned frame at the start point of the robot.

In order to reduce errors accumulated over time, leg slip should be detected and the slipping leg should be discarded from the calculations, if possible. To assess the severity of the point cloud deformation due to slip, the quadratic error of the rigid body transformation for two consecutive time steps is calculated. If this error is higher than an acceptable threshold, the algorithm will try to identify the leg that causes the strongest distortion. For this purpose, the relative distance of each leg in contact to each other stance leg is calculated and compared to the distances of the previous time step. If more than three legs are in contact, the leg with the largest change of distance to all other legs in contact is removed from the calculation by setting its weight equal to zero. In most cases this approach already reduces the squared error of the transformation such that it is smaller than the acceptable threshold. If it does not suffice or only three legs are in contact, the remaining error will be compared to a second threshold and the rotation and translation during this time interval are either neglected or accepted. If many odometry calculations are neglected the odometry is invalidated. Nevertheless, slippage of the complete robot on a slope or on icy ground cannot be detected by this approach, and thus remains a limitation of the leg odometry.

6.2.2 Joint-Torques-Based Pitch and Roll Angle Estimates

Within its second stage, the leg odometry algorithm computes absolute pitch and roll angle estimates for the robot which are based on joint torque measurements. The approach assumes that the joint loads of slowly walking robots, e.g. $< 10\,\text{cm/s}$ for the DLR Crawler, mainly originate in gravitational forces that act on the body of the robot. Following this assumption of quasi-static behaviour, the ground contact forces

of each leg ${}^{B}\mathbf{f}_{l,i}$, $i = 1 \ldots 6$, with respect to the leg base frame B are calculated from its measured joint torque vector $\boldsymbol{\tau}_{l,i}$ and its individual leg Jacobian $\mathbf{J}_{l,i}$.

$$\boldsymbol{\tau}_{l,i} = \mathbf{J}_{l,i}^{T}\,{}^{B}\mathbf{f}_{l,i} \tag{6.12}$$

$${}^{B}\mathbf{f}_{l,i} = (\mathbf{J}_{l,i}^{T})^{-1}\boldsymbol{\tau}_{l,i} \tag{6.13}$$

After transformation from the leg base frame into the body frame, the sum of these individual ground contact forces ${}^{b}\mathbf{f}_{l,i}$ yields the total ground contact force of the robot ${}^{b}\mathbf{f}$ that is assumed to be mostly caused by gravity. Given with respect to the body frame, the total ground contact force vector allows for calculating the pitch angle β_{abs} as well as the roll angle γ_{abs} of the robot.

$$\gamma_{\text{abs}} = \text{atan2}({}^{b}f_y, {}^{b}f_z) \tag{6.14}$$

$$\beta_{\text{abs}} = \text{atan2}(-{}^{b}f_x, {}^{b}f_y \sin(\gamma_{\text{abs}}) + {}^{b}f_z \cos(\gamma_{\text{abs}})) \tag{6.15}$$

If there are no torque sensors available, foot force sensors are equally useful to compute absolute pitch and roll angle estimates. In this case, the total ground reaction force vector is the sum of the individual foot force measurements that are rotated from their sensor frame into the body frame.

6.2.3 Error-State-Kalman-Filter-Based Data Fusion

To improve the overall pose estimate, the joint-angles-based pitch and roll angle estimates of the first stage are fused with the joint-torques-based pitch and roll angle estimates of the second stage by an error-state Kalman filter in feedback configuration. The fusion process shown in Fig. 6.2 combines the fast components of the joint-angles-based estimates with the slow components of the joint-torques-based estimates. Consequently, it removes the drift of the pitch and roll angles from the first stage of the algorithm as well as the ground-impact-related peaks from the second stage. Since both stages employ the same joint angle measurements, the results are not optimal in the sense of Kalman filter theory. Although the measurements are not independent, the correlation of errors is expected to be small. The reasons for this expectation are the different sources of errors. For the first stage, the drift results from the calculation of the transformation rather than from the joint angles themselves, while the errors in the second stage originate in the ground impacts measured by the joint torque sensors. Thus, not being optimal, the filter is still a good method to stabilises the leg

odometry using leg sensors only. In addition, the chosen error-state formulation of the filter has the advantage that it only estimates the error of the states but not the states themselves. Therefore, the filter does not require a complex motion model of the robot which eases the computation. Furthermore, due to the feedback formulation the error is corrected at each step and the assumption of a perfect correction allows to simplify the prediction stage of the filter as well. In the following, the general equations of a Kalman filter, as given in [Welch and Bishop, 2006], are reduced to the error-state formulation.

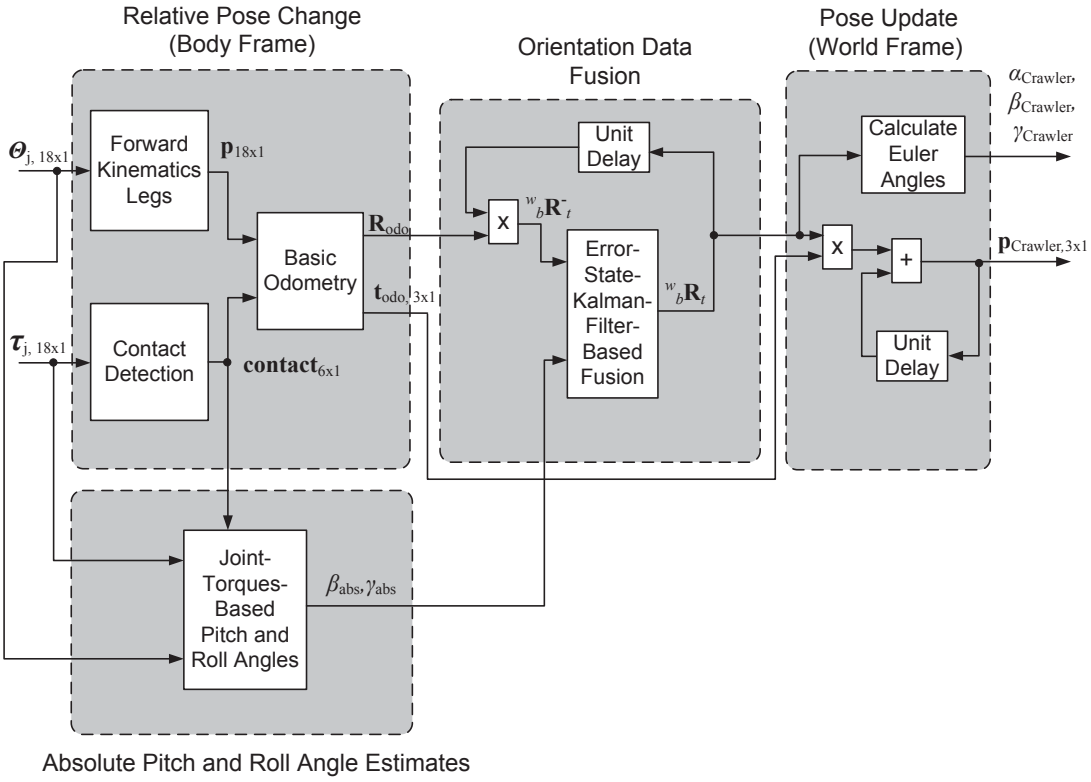


Figure 6.2: Signal flow diagram for the complete joint-torque-aided 6-DOF odometry implemented on the DLR Crawler

The first part of the filter is the prediction or time update step. Herein, $\Delta\hat{\mathbf{x}}_t = [\Delta\beta_{\text{cor}}, \Delta\gamma_{\text{cor}}]^T$ represents the rotation angle error estimates and $\Delta\mathbf{z}_t = [\Delta\beta, \Delta\gamma]^T$ the rotation angle error measurements, which are derived later in this section following the Kalman filter description.

$$\Delta\hat{\mathbf{x}}_t^- = \mathbf{A}_t\Delta\hat{\mathbf{x}}_{t-1} = 0 \tag{6.16}$$

$$\mathbf{P}_t^- = \mathbf{A}_t\mathbf{P}_{t-1}\mathbf{A}_t^T + \mathbf{Q}_p \tag{6.17}$$

In this first stage of the filter, (6.16) presents an error propagation model, wherein the terms $\Delta\hat{\mathbf{x}}_t^-$ and $\Delta\hat{\mathbf{x}}_{t-1}$ are the predicted error estimate at time t and the corrected error estimate at time $t-1$, respectively. Due to the feedback formulation of the filter, the error estimate of the previous time step $\Delta\hat{\mathbf{x}}_{t-1}$ is set to zero. Therefore, the predicted error estimate of the current time step $\Delta\hat{\mathbf{x}}_t^-$ is also zero. The process matrix $\mathbf{A}_t$ is set to be a 2×2 identity matrix, $\mathbf{A}_t = \mathbf{I}_{2\times2}$. (6.17) gives the predicted estimate of the error covariance matrix $\mathbf{P}_t^-$ based on the previous estimate of the error covariance matrix $\mathbf{P}_{t-1}$ and the process noise covariance matrix $\mathbf{Q}_p$.

The second part of the filter is the correction or measurement update step. Here the Kalman gain matrix $\mathbf{K}_t$ is calculated based on the predicted estimate of the process error covariance matrix $\mathbf{P}_t^-$, the matrix $\mathbf{H} = \mathbf{I}_{2\times2}$ relating the measured errors $\Delta\mathbf{z}_t$ and the error-state estimates $\Delta\hat{\mathbf{x}}_t$ as well as the measurement noise covariance matrix $\mathbf{Q}_m$. Depending on the covariance matrices, the Kalman gain matrix adjusts the influence on the corrected error estimate either towards the predicted error estimate or towards the measurement. The last part of the correction step is the update of the estimated error covariance matrix $\mathbf{P}_t$.

$$\mathbf{K}_t = \mathbf{P}_t^-\mathbf{H}^T(\mathbf{H}\mathbf{P}_t^-\mathbf{H}^T + \mathbf{Q}_m)^{-1} \tag{6.18}$$

$$\Delta\hat{\mathbf{x}}_t = \mathbf{K}_t\Delta\mathbf{z}_t \tag{6.19}$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t^- \tag{6.20}$$

For the odometry, all calculations that involve rotation matrices and Euler angles follow the xyz-convention. The rotation matrices are calculated as shown below with

yaw angle α, pitch angle β, and roll angle γ, where "c" represents a cosine and "s" a sine.

$$\begin{aligned}\mathbf{R} &= \mathbf{R}_{\alpha} \cdot \mathbf{R}_{\beta\gamma} \\ &= \begin{bmatrix} \mathrm{c}\alpha & -\mathrm{s}\alpha & 0 \\ \mathrm{s}\alpha & \mathrm{c}\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathrm{c}\beta & \mathrm{s}\beta\mathrm{s}\gamma & \mathrm{s}\beta\mathrm{c}\gamma \\ 0 & \mathrm{c}\gamma & -\mathrm{s}\gamma \\ -\mathrm{s}\beta & \mathrm{c}\beta\mathrm{s}\gamma & \mathrm{c}\beta\mathrm{c}\gamma \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{c}\alpha\mathrm{c}\beta & \mathrm{c}\alpha\mathrm{s}\beta\mathrm{s}\gamma - \mathrm{s}\alpha\mathrm{c}\gamma & \mathrm{c}\alpha\mathrm{s}\beta\mathrm{c}\gamma + \mathrm{s}\alpha\mathrm{s}\gamma \\ \mathrm{s}\alpha\mathrm{c}\beta & \mathrm{s}\alpha\mathrm{s}\beta\mathrm{s}\gamma + \mathrm{c}\alpha\mathrm{c}\gamma & \mathrm{s}\alpha\mathrm{s}\beta\mathrm{c}\gamma - \mathrm{c}\alpha\mathrm{s}\gamma \\ -\mathrm{s}\beta & \mathrm{c}\beta\mathrm{s}\gamma & \mathrm{c}\beta\mathrm{c}\gamma \end{bmatrix}\end{aligned} \tag{6.21}$$

To fuse the orientation angles from the first stage of the algorithm with pitch and roll angles from the second stage, an orientation angle error vector is calculated that is based on the predicted rotation matrix ${}^{w}_{b}\mathbf{R}^{-}_{t}$ and the measured rotation matrix $\mathbf{R}_{\beta\gamma\mathrm{meas},t}$. Herein, the matrix ${}^{w}_{b}\mathbf{R}^{-}_{t}$ is calculated from the corrected rotation matrix of the previous time step ${}^{w}_{b}\mathbf{R}_{t-1}$ and the matrix $\mathbf{R}_{\mathrm{odo},t}$ that represents the actual incremental rotation returned by the first stage of the algorithm. With respect to the matrices, the subscript b and the superscript w refer to a rotation from the body frame into the world frame.

$${}^{w}_{b}\mathbf{R}^{-}_{t} = {}^{w}_{b}\mathbf{R}_{t-1}\mathbf{R}_{\mathrm{odo},t} \tag{6.22}$$

Since the joint torque measurements allow no yaw angle estimate, $\mathbf{R}_{\beta\gamma\mathrm{meas},t}$ only consists of pitch and roll terms. The predicted rotation matrix ${}^{w}_{b}\mathbf{R}^{-}_{t}$ is separated into a rotation matrix $\mathbf{R}^{-}_{\alpha,t}$ representing the yaw component and a rotation matrix $\mathbf{R}^{-}_{\beta\gamma,t}$ representing the pitch and roll components. Only $\mathbf{R}^{-}_{\beta\gamma,t}$ is used in the fusion process and is related to $\mathbf{R}_{\beta\gamma\mathrm{meas},t}$ by the following equation.

$$\mathbf{R}_{\Delta} = \mathbf{R}^{-}_{\beta\gamma,t} \cdot \mathbf{R}^{T}_{\beta\gamma\mathrm{meas},t} \tag{6.23}$$

Herein, $\mathbf{R}_{\Delta} = [r_{\Delta,(i,j)}]_{3\times3}$ is a matrix that represents the pitch and roll rotation error measurement and $\Delta\mathbf{z}_t = [\Delta\beta, \Delta\gamma]^T$ can be calculated as follows.

$$\Delta\beta = \arcsin(-r_{\Delta,(3,1)}) \tag{6.24}$$

$$\Delta\gamma = \mathrm{atan2}(r_{\Delta,(3,2)}, r_{\Delta,(3,3)}) \tag{6.25}$$

Applying both steps of the Kalman filter and using $\Delta\hat{\mathbf{x}}_t = [\Delta\beta_{\text{cor}}, \Delta\gamma_{\text{cor}}]^T$ to calculate a corrected rotation error matrix $\mathbf{R}_{\Delta,\text{cor}}$, the corrected rotation matrix of the current time step ${}^{w}_{b}\mathbf{R}_t$ can be calculated.

$$ {}^{w}_{b}\mathbf{R}_t = \mathbf{R}^{-}_{\alpha,t} \cdot \mathbf{R}^{T}_{\Delta,\text{cor}} \cdot \mathbf{R}^{-}_{\beta\gamma,t} \tag{6.26} $$

Using the above matrix, the position of the walking robot (the DLR Crawler in this case) with respect to the world frame ${}^{w}\mathbf{p}_{\text{Crawler},t}$ can be updated based on the relative position change ${}^{b}\mathbf{t}_{\text{odo}}$ computed by the first stage of the leg odometry algorithm.

$$ {}^{w}\mathbf{p}_{\text{Crawler},t} = {}^{w}\mathbf{p}_{\text{Crawler},t-1} + {}^{w}_{b}\mathbf{R}_t \cdot {}^{b}\mathbf{t}_{\text{odo}} \tag{6.27} $$

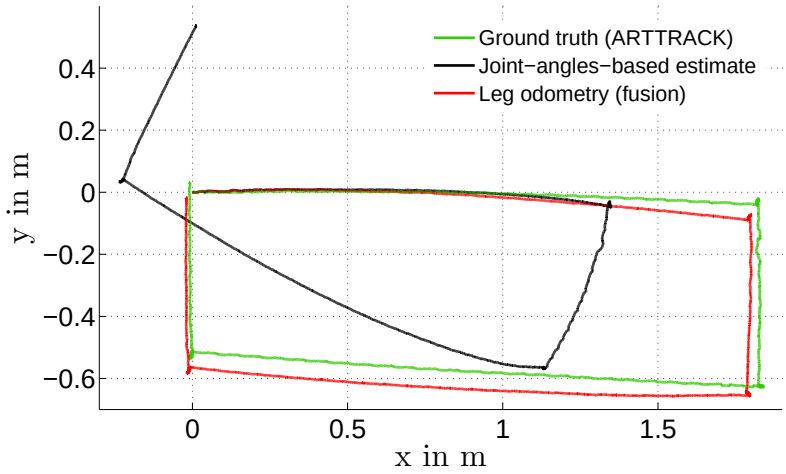
The measurement noise covariance matrix $\mathbf{Q}_m$ is significantly larger than the process noise covariance matrix $\mathbf{Q}_p$. The final settings have to be found by manual filter tuning and are further discussed in the following section.

6.3 Experiments

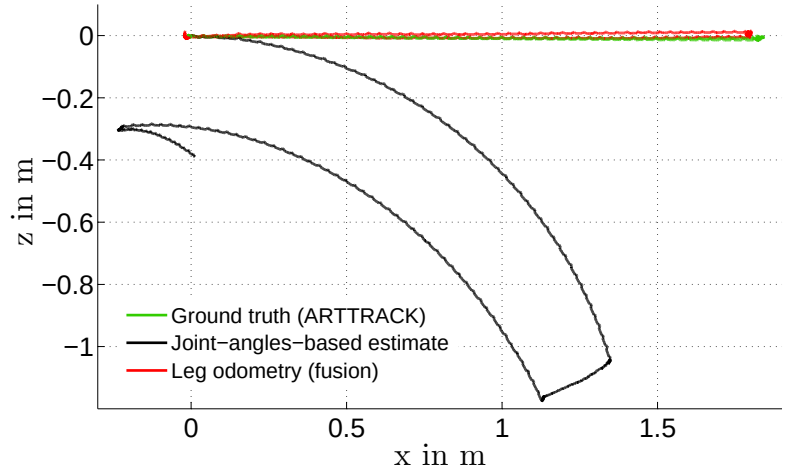
This section presents experimental results for the complete 6-DOF leg odometry algorithm. The DLR Crawler serves as test system and all runs were either performed on flat lab floor or within a 2 m × 2 m box filled with gravel 10 to 15 cm high. First, some examples illustrate the performance of each single stage of the algorithm in order to motivate the necessity of data fusion. Then, different error sources are identified and three parameters are presented that are used to attenuate the errors and to adjust the odometry algorithm for different conditions. In the following, the performance of the algorithm is evaluated for forward walking and turning on two different substrates, at two different joint stiffness settings, and at two different desired walking velocities. Furthermore, Section 6.3.5 presents some results for walking along a rectangular path on gravel as well as for walking along a path that includes uphill segments and a downward step. During all of the test runs the robot was steered manually and ground truth measurements were recorded using an A.R.T. tracking system. For this purpose, a target body was attached to the DLR Crawler that was tracked with four infrared cameras to obtain ground truth data of the 6-DOF pose with an average accuracy of 0.5 mm for the translational DOF and 0.12 ° for the orientation angles.

6.3.1 Behaviour of the Individual Stages: Drift and Error Sources

To give an example for the performance of the first stage of the leg odometry algorithm, Fig. 6.3 compares those estimates to the data of the ground truth measurement system (GTMS). The diagram clearly shows that the path estimated by the first stage of the algorithm is strongly bent and deviates from the ground truth measurements.



(a) Path of the DLR Crawler projected onto the xy-plane



(b) Path of the DLR Crawler projected onto the xz-plane

Figure 6.3: The DLR Crawler walks along a rectangular path on flat ground ($v_{des} = 6\,cm/s$, medium joint stiffness): Comparison of the COM position estimates of the first stage of the odometry algorithm, the complete leg odometry (data fusion) as well as the ground truth measurement system

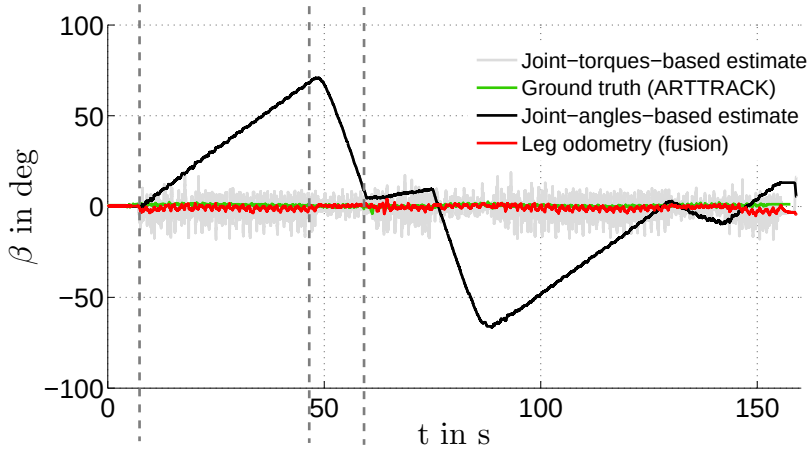
This behaviour is mainly attributed to the interaction of the odometry calculation and the active compliance, which results in an angular drift that is shown in the

"forward walking" marked region of Fig. 6.4. The drift of the pitch or roll angle depends on the direction of motion and strongly affects the absolute pose estimate. For forward or backward walking, the pitch angle is affected, while sideways walking leads to a drift of the roll angle. During pure turning this disturbing behaviour was not observed. In order to give an explanation, the case of forward walking is considered. Here, after touch down, a front leg moves towards the COG of the robot. During this motion the loading of the leg increases and causes its height to decrease due to the active compliance. Opposite to this behaviour, the loading of a hind leg decreases over the course of its stance motion since it moves away from the robot COG. Thus, the leg extends depending on the stiffness setting. To each calculation of the incremental pose change this behaviour appears to be a tilting motion that increases the pitch angle, summing up to the strong angular drift apparent in Fig. 6.4. Having built up a large pitch angle, each turning motion, i. e. increase or decrease of the yaw angle, transfers the pitch to a roll angle, which is apparent within the plots.

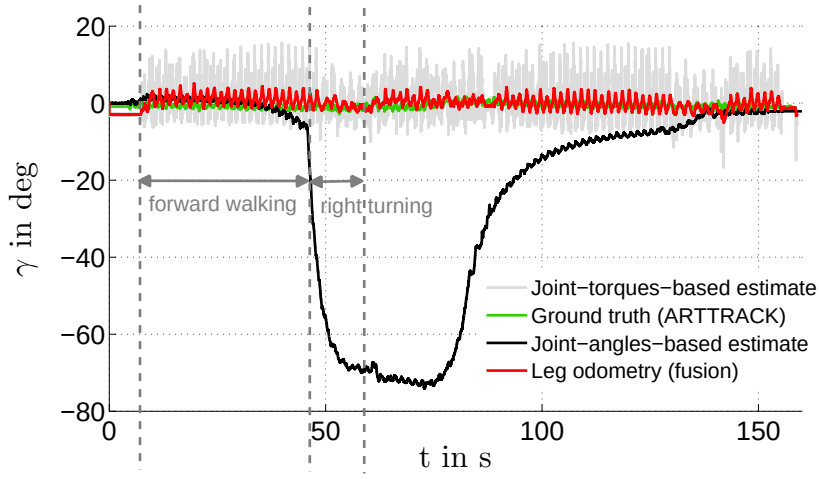
Another error is caused by the initial contact phase of the legs, especially during the execution of the stretch reflex while walking on uneven ground. The source of this error is that the algorithm considers larger parts of the downward motion at the beginning of the stance phase than of the upward motion at the end of the stance phase, where the leg quickly loses contact. The downward motion of a leg caused by the stretch reflex has an effect on the translation estimation and appears to the algorithm as upward motion of the robot body. In order to attenuate this behaviour, the ground contact has to be detected properly. For this purpose, contact thresholds are introduced that can be adjusted and influence the pose estimate depending on the walking speed and the terrain, which is discussed in further detail in the following subsection on parameter tuning.

The error caused by the stretch reflexes appears randomly and depends on the distribution of the height differences across the ground. The error due to the compliance is somehow systematic but depends on many parameters like joint stiffness, walking velocity, and ground properties. For this reason, an error prediction and correction without further sources of information is infeasible. Thus, absolute pitch and roll angles from a different source have to correct the odometry data. As introduced in Section 6.4, the joint torque measurements are sufficient to estimate the gravity vector with respect to the body frame.

To give an example for the performance of the second stage of the odometry algorithm, Fig. 6.5 shows joint-torque-measurement-based pitch angle estimates for the DLR Crawler. The upper figure refers to the Crawler standing on a slowly tilting board that is manually moved. It shows that the pitch angle estimate has an initial

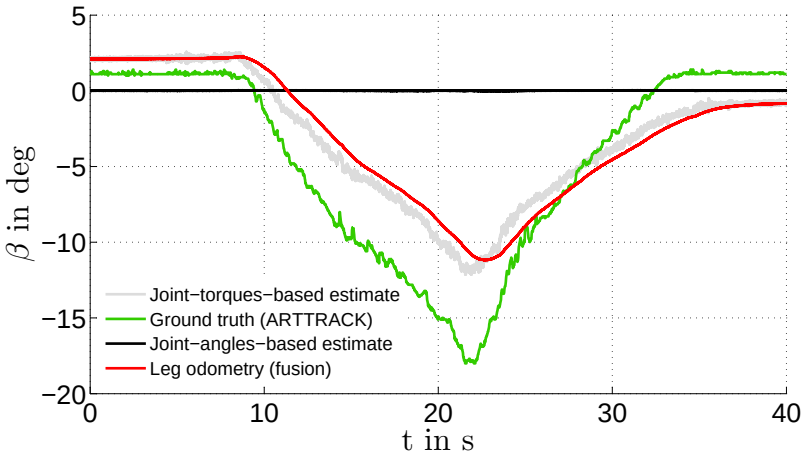


(a) Pitch angle of the DLR Crawler

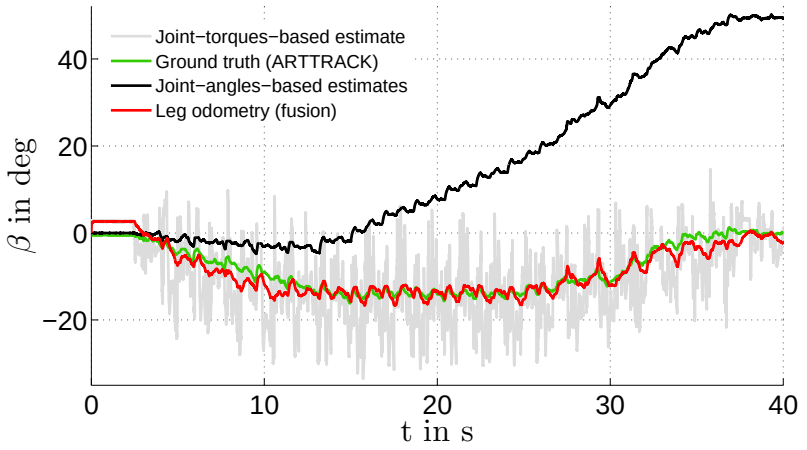


(b) Roll angle of the DLR Crawler

Figure 6.4: The DLR Crawler walks along a rectangular path on flat ground: Comparison of the pitch and roll angle estimates provided by the first and second stage of the odometry algorithm, the complete leg odometry (data fusion) as well as the ground truth measurement system



(a) Pitch angle of the DLR Crawler on a tilting board



(b) Pitch angle of the DLR Crawler for walking on a 15° slope

Figure 6.5: Comparison of different pitch angle estimates for the DLR Crawler

offset and does not closely track the angular motion. The lower figure depicts the pitch angle estimate for the Crawler walking uphill. It clearly shows that the ground impacts during walking cause strong force peaks that are translated to false peaks within the pitch angle estimates. It also shows that for walking uphill the pitch angle estimate of the first stage of the algorithm as presented in Section 6.2.1 is dominated by the compliance induced drift. Nevertheless, standing on the tilting board or walking uphill, the data contains information about the inclination of the robot. Thus, the joint torque sensors emulate an inclination sensor even though it is not a very accurate one. Since the joint-torque-based pitch and roll angle estimates are very noisy but free of drift and the joint-angle-based estimates include little noise but show strong drift, they are very well suited to be fused by a Kalman filter as presented in the previous section. The improvements gained by this fusion process with respect to each single stage become apparent in Fig. 6.3, Fig. 6.4 and Fig. 6.5. As can be seen, the large position errors caused by the compliance-induced pitch angle drift are completely removed by fusion with the joint-torque-based absolute pitch and roll angles. The remaining position error mainly originates in a small yaw angle drift during forward walking and slight underestimation of the yaw angle during turning. Due to a missing absolute reference for the yaw angle this error cannot be removed.

6.3.2 Tuning Parameters

The algorithm provides three parameters that are adjusted in order to increase the accuracy of the estimate and to account for different conditions. These parameters are two torque thresholds used for contact detection in the first stage of the algorithm and the process noise covariance matrix of the error-state Kalman filter in the third stage.

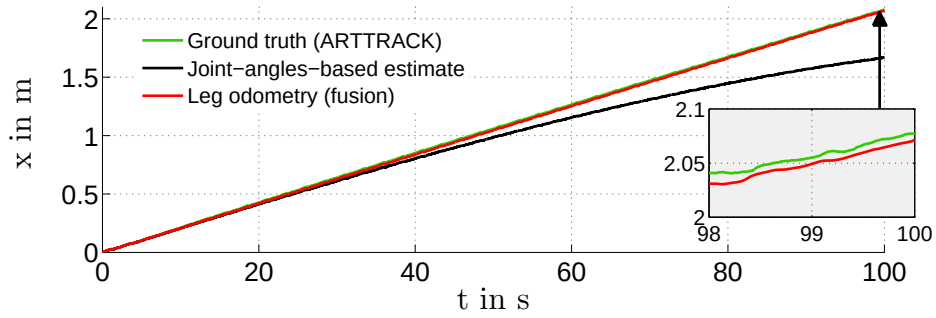
The first torque threshold is active during the swing phase of the leg and detects the initial contact that marks the onset of the stance phase. The second threshold becomes active once the stance is established. It is used to monitor if the leg loses contact during stance and to detect the onset of the next swing phase once the leg lifts. In all cases the first threshold is higher than the second and helps to discard the error-prone initial contact and loading phase of a leg that has strong influence on errors of the z-coordinate and especially the yaw angle. Thus, the first threshold reduces the error of the yaw estimate while the second mainly reduces the error of the z-coordinate estimate. During the experiments two distinct values were identified for the first threshold that are independent of the terrain or the stiffness setting and are only influenced by the walking speed. Here, the first constant value was used for slow walking dominated by emerging pentapod or tetrapod gaits. The threshold had to be increased to the second constant value for fast walking with dominating tripod gaits.

This is actually obvious since the leg load is higher during the tripod. Furthermore, the second torque threshold appears to be independent of the stiffness but needs to be adjusted for each terrain and once the gait changes from tetrapod to tripod.

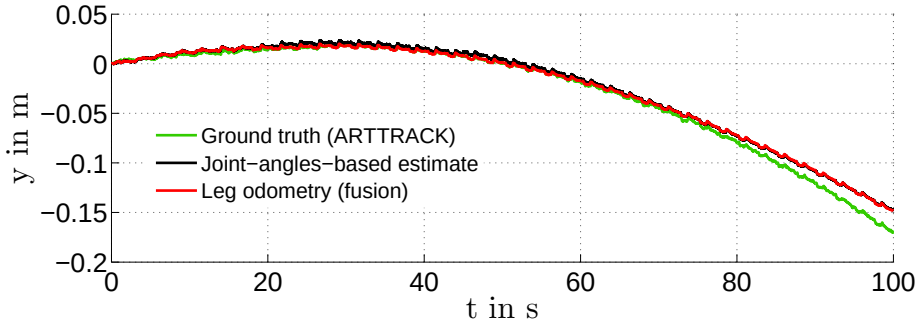
The last tuning parameter, the process noise covariance matrix, is used to remove the remaining z-coordinate drift. The measurement noise covariance matrix is kept equal for all settings and the process noise covariance matrix is assumed to change depending on the gait, the stiffness setting and the terrain. For walking on gravel the values were smaller than for walking on lab floor. Considering the fixed measurement noise covariance matrix, this means that for walking on gravel the joint-torque-based pitch and roll estimates are less trusted. Furthermore, the process noise covariance values are smaller for slow walking on gravel than for fast walking on gravel and did not change when the stiffness changed. For walking on lab floor the values had to be increased once the gait changed from tetrapod to tripod and also had to be increased once the stiffness was increased. Altogether, there is no single set of parameters that is valid for all combinations of gait, terrain, and joint stiffness. Nevertheless, different parameter sets can be identified and stored depending on the combination of the dominating gait, the stiffness setting, and the terrain. This identification is currently done manually but will be done automatically in future by calibrating the leg odometry on a new terrain type with respect to the IMU and visual odometry derived pose estimates. Each time a change of dominating gait or stiffness is initiated or a change of terrain is detected by visual cues or a change of reflex activation behaviour, the filter can be automatically adapted by loading the appropriate parameter set.

6.3.3 Forward Walking

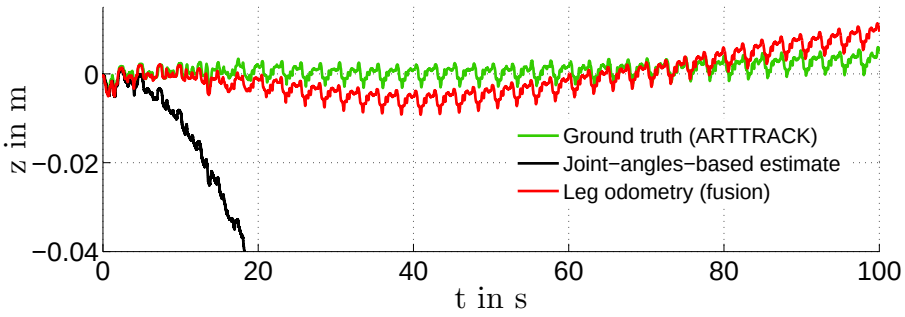
This set of experiments was conducted to evaluate the performance of the complete leg odometry and its associated errors under varying conditions. For this purpose, the robot walked forward at a certain velocity and the absolute translation and orientation errors were measured at the times when the ground truth measurement system (GTMS) indicated 0.5 m, 1 m, and 1.5 m travel in the x-direction of the local start frame. In order to evaluate the algorithm for different emerging gaits, two sets of trials at a lower and a higher velocity were performed. First, the robot walked at a commanded velocity of 3 cm/s to enforce a tetrapod gait while a velocity command of 6 cm/s mainly generated a tripod gait. For each velocity setting, runs at two different stiffness values were recorded: A medium joint stiffness setting and a stiff configuration with doubled joint stiffness values. Furthermore, each velocity-stiffness configuration was tested on laboratory floor as well as in the gravel testbed, which resulted in 8 different test conditions. For each of these 8 conditions, 10 separate runs were recorded and analysed.



(a) x-position of the DLR Crawler

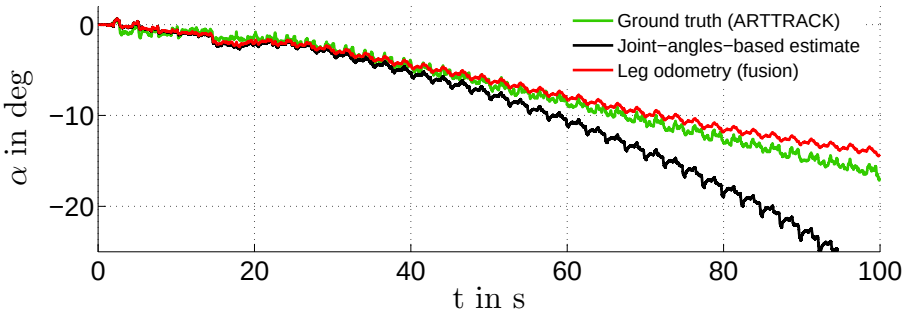


(b) y-position of the DLR Crawler

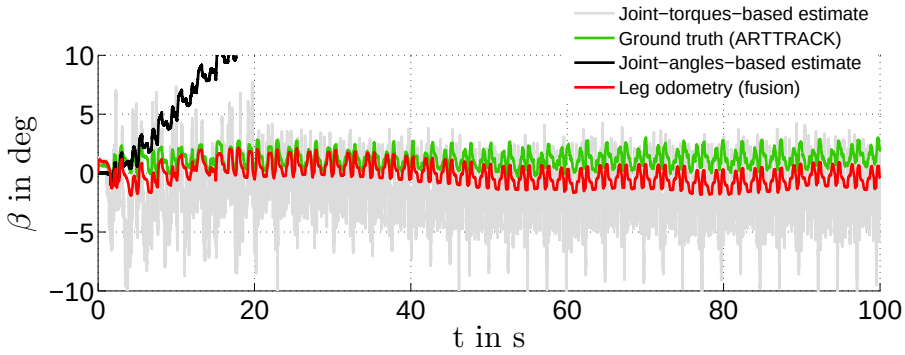


(c) z-position of the DLR Crawler

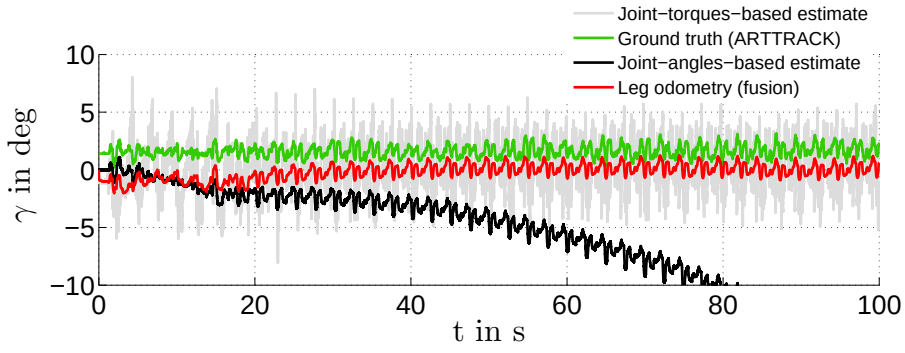
Figure 6.6: Exemplary Crawler odometry COM position estimates for forward walking on lab floor at a velocity of $v_{des} = 3\,cm/s$ and medium joint stiffness



(a) Yaw angle of the DLR Crawler

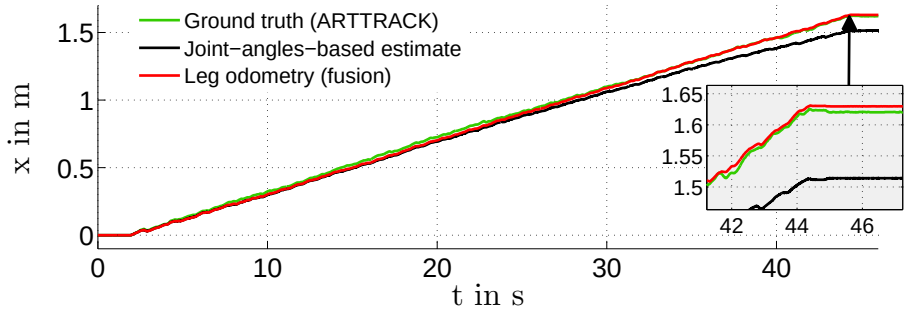


(b) Pitch angle of the DLR Crawler

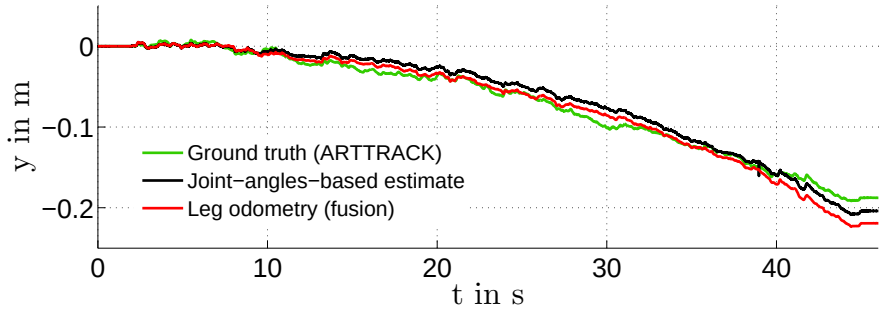


(c) Roll angle of the DLR Crawler

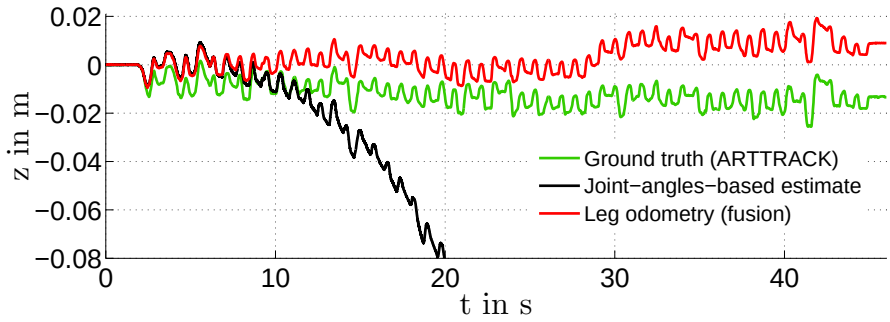
Figure 6.7: Exemplary Crawler odometry Euler angles estimates for forward walking on lab floor at a velocity of $v_{des} = 3\,cm/s$ and medium joint stiffness



(a) x-position of the DLR Crawler

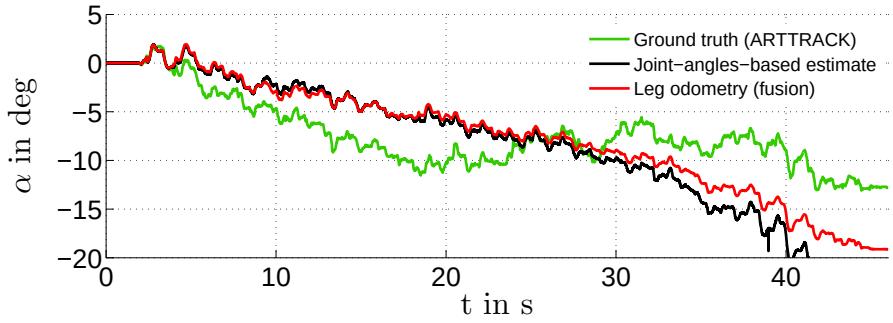


(b) y-position of the DLR Crawler

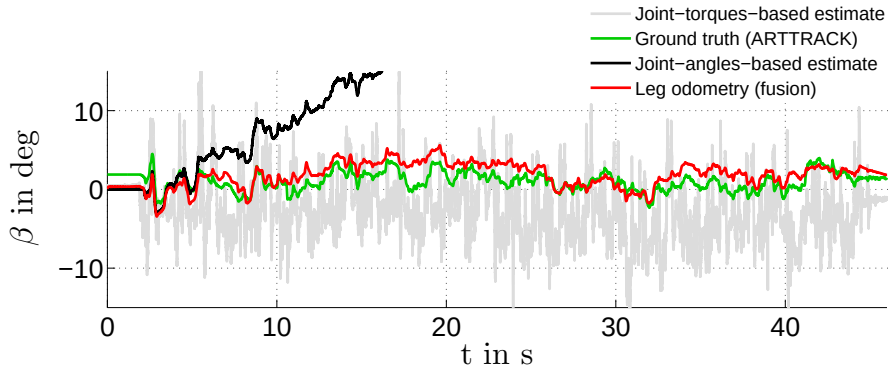


(c) z-position of the DLR Crawler

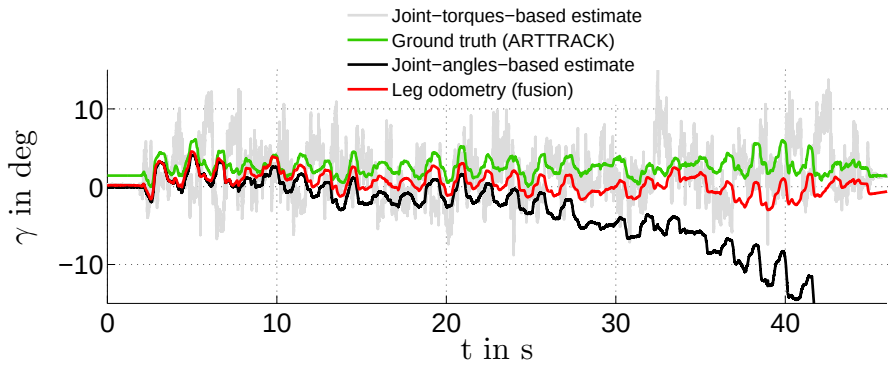
Figure 6.8: Exemplary Crawler odometry COM position estimates for forward walking on gravel at a velocity of $v_{des} = 6\,cm/s$ and medium joint stiffness



(a) Yaw angle of the DLR Crawler



(b) Pitch angle of the DLR Crawler



(c) Roll angle of the DLR Crawler

Figure 6.9: Exemplary Crawler odometry Euler angles estimates for forward walking on gravel at a velocity of $v_{des} = 6\,cm/s$ and medium joint stiffness

Figures 6.6 and 6.7 as well as Figs. 6.8 and 6.9 display the pose estimates for good exemplary runs at 3 cm/s on lab floor and at 6 cm/s on gravel, respectively. The graphs show that the actual walking velocity in both cases is slightly lower than the commanded velocity, which is attributed to the medium joint stiffness setting and the resulting reduced tracking accuracy of the joints. After walking a comparable distance of 1.5 m in the x-direction of the local start frame, as measured by the GTMS, the absolute Cartesian position errors are 1.74 cm for the run at 3 cm/s on lab floor and 3.21 cm for the run at 6 cm/s on gravel. The absolute Cartesian position errors at this distance with respect to the Cartesian path length are 0.82 % and 1.39 %, respectively. In both cases the dominant source of the endpoint errors is a drift of the yaw angle estimate. Additionally, the baselines of the z-coordinate estimates show deviations from the baselines of the GTMS data. Nevertheless, the oscillations around the baselines closely represent the observed z-coordinate variations that result from the change of stance configurations. The baseline deviations are caused by a combination of small remaining influences from the stretch reflex and the compliance induced z-coordinate drift as well as inaccuracies in the pitch angle estimate. Furthermore, the pitch and roll angle estimates show offsets that appear due to horizontal propulsion forces, which have been assumed to be negligible at low speeds.

Table 6.1 displays the means and standard deviations of the errors observed during the forward walking experiments. The computed errors for each single run are the Cartesian position errors in the x-, y-, and z-direction with respect to the start frame, the absolute Cartesian position error with respect to the travelled path length, and the root mean square (rms) errors of the yaw, pitch, and roll angles all measured after 0.5 m, 1 m, and 1.5 m travel in the x-direction of the start frame. Apparently, the odometry algorithm underestimates the distance travelled in the x-direction for all trials on lab floor. In contrast, it overestimates the travelled distance on gravel due to increased leg slip that is partially mistaken for forward body motion. In almost all cases the lateral motion in the y-direction is overestimated which is strongest on gravel and mainly caused by larger yaw angle errors. In all cases, the yaw angle rms error increases with distance, indicating a yaw angle drift. Pitch and roll angle rms errors show in most cases constant values independent of the distance travelled. Only on lab floor the pitch angle shows a slight drift but with errors being smaller than the ones for gravel. The z-coordinate estimates show the largest errors for slow walks on gravel. Considering the absolute Cartesian position error with respect to the path length, Δp, the smallest error on gravel is obtained with a tripod gait and medium joint stiffness settings, while on lab floor a tetrapod gait and medium joint stiffness is best. Nevertheless, with the right torque thresholds and process covariance matrices all

Table 6.1: Error behaviour of the complete Crawler odometry (with fusion of first and second stage) for **forward walking**: Average Cartesian position errors (Δx,Δy,Δz) and average orientation rms errors ($\Delta\alpha_{\text{rms}}$,$\Delta\beta_{\text{rms}}$,$\Delta\gamma_{\text{rms}}$) at different combinations of ground substrate, desired forward velocity, and joint stiffness at 50 cm, 100 cm, and 150 cm distance to the start point with respect to the x-direction of the start frame (10 experimental runs for each combination); **Δp is the absolute Cartesian position error with respect to the travelled Cartesian path length**

v_{des}	Stiffness	Δx in cm mean (std)	Δy in cm mean (std)	Δz in cm mean (std)	**Δp in % mean (std)**	$\Delta\alpha_{\text{rms}}$ in ° mean (std)	$\Delta\beta_{\text{rms}}$ in ° mean (std)	$\Delta\gamma_{\text{rms}}$ in ° mean (std)
$x_{tracking} = 50$ cm								
Lab floor								
3 cm/s	medium	-1.15 (0.38)	0.06 (0.30)	-0.18 (0.49)	**1.91 (0.44)**	0.55 (0.21)	0.33 (0.83)	2.28 (0.24)
3 cm/s	stiff	-1.49 (0.20)	0.61 (0.33)	-0.13 (0.34)	**2.48 (0.29)**	0.46 (0.26)	1.40 (0.30)	0.99 (0.40)
6 cm/s	medium	-0.78 (0.45)	0.03 (0.52)	-0.14 (0.15)	**1.6 (0.33)**	0.34 (0.18)	1.27 (0.1)	2.37 (0.17)
6 cm/s	stiff	-0.48 (0.21)	0.14 (0.15)	-0.29 (0.26)	**1.03 (0.31)**	0.31 (0.04)	1.30 (0.19)	2.46 (0.15)
Gravel								
3 cm/s	medium	0.80 (2.28)	0.15 (2.48)	2.47 (0.89)	**4.82 (2.28)**	1.79 (1.39)	1.79 (0.67)	1.26 (0.39)
3 cm/s	stiff	0.83 (1.70)	0.99 (1.65)	3.09 (1.82)	**4.70 (2.18)**	2.44 (1.02)	2.54 (0.80)	1.76 (0.81)
6 cm/s	medium	0.87 (1.37)	1.84 (1.96)	0.04 (1.04)	**3.86 (1.87)**	2.12 (1.23)	2.45 (0.58)	2.02 (0.85)
6 cm/s	stiff	0.07 (1.62)	1.00 (1.83)	0.29 (1.50)	**3.59 (1.53)**	2.01 (1.14)	2.16 (0.51)	2.60 (1.63)
$x_{tracking} = 100$ cm								
Lab floor								
3 cm/s	medium	-1.64 (0.59)	0.01 (0.55)	-0.36 (1.08)	**1.47 (0.48)**	0.59 (0.22)	1.16 (0.31)	2.05 (0.18)
3 cm/s	stiff	-2.13 (0.49)	1.12 (1.15)	0.07 (0.74)	**2.00 (0.41)**	0.76 (0.23)	1.23 (0.29)	0.83 (0.31)
6 cm/s	medium	-1.77 (0.59)	-0.08 (0.92)	-0.08 (0.46)	**1.63 (0.23)**	0.38 (0.24)	1.38 (0.2)	2.32 (0.08)
6 cm/s	stiff	-1.34 (0.40)	0.69 (0.40)	-0.46 (0.39)	**1.31 (0.33)**	0.39 (0.11)	1.50 (0.15)	2.16 (0.10)
Gravel								
3 cm/s	medium	2.33 (2.90)	1.97 (5.17)	2.15 (1.79)	**3.79 (2.29)**	2.47 (1.27)	2.31 (0.44)	1.68 (0.32)
3 cm/s	stiff	0.44 (2.22)	0.47 (2.89)	4.16 (3.10)	**3.37 (1.71)**	2.55 (1.42)	2.38 (0.28)	1.82 (0.54)
6 cm/s	medium	0.71 (1.98)	2.26 (2.40)	-0.25 (1.77)	**2.47 (1.25)**	2.29 (1.14)	2.63 (0.84)	2.27 (0.92)
6 cm/s	stiff	-0.58 (2.24)	3.39 (4.04)	-0.17 (1.52)	**3.37 (2.00)**	2.82 (1.24)	2.33 (0.50)	2.80 (1.04)
$x_{tracking} = 150$ cm								
Lab floor								
3 cm/s	medium	-1.74 (0.85)	0.86 (1.22)	-0.31 (1.73)	**1.30 (0.52)**	0.83 (0.36)	1.95 (0.58)	1.93 (0.20)
3 cm/s	stiff	-2.77 (0.89)	2.97 (2.18)	0.35 (1.24)	**2.24 (0.46)**	1.34 (0.54)	1.27 (0.32)	0.76 (0.23)
6 cm/s	medium	-2.69 (0.64)	-0.3 (1.47)	0.32 (0.68)	**1.63 (0.24)**	0.5 (0.2)	1.63 (0.17)	2.31 (0.06)
6 cm/s	stiff	-2.17 (0.45)	1.13 (0.86)	-0.55 (0.41)	**1.37 (0.27)**	0.43 (0.17)	1.71 (0.09)	1.97 (0.07)
Gravel								
3 cm/s	medium	3.88 (3.55)	1.08 (7.98)	2.57 (2.17)	**3.71 (1.79)**	3.13 (1.32)	2.42 (0.51)	1.88 (0.39)
3 cm/s	stiff	1.32 (3.43)	1.83 (5.57)	4.44 (4.46)	**3.26 (1.60)**	2.95 (1.52)	2.36 (0.43)	1.98 (0.41)
6 cm/s	medium	2.81 (2.4)	2.53 (5.05)	-0.57 (2.51)	**2.54 (1.69)**	2.51 (1.48)	2.59 (0.70)	2.30 (0.79)
6 cm/s	stiff	0.56 (2.79)	5.93 (6.20)	-0.57 (2.15)	**3.58 (1.90)**	3.22 (1.23)	2.21 (0.43)	2.94 (0.73)

configurations show comparable results. As expected the estimation errors are smaller on lab floor where Δp is within 1 % to 3 %, while on gravel it mainly lies in a range of 2 % to 6 %.

6.3.4 Turning

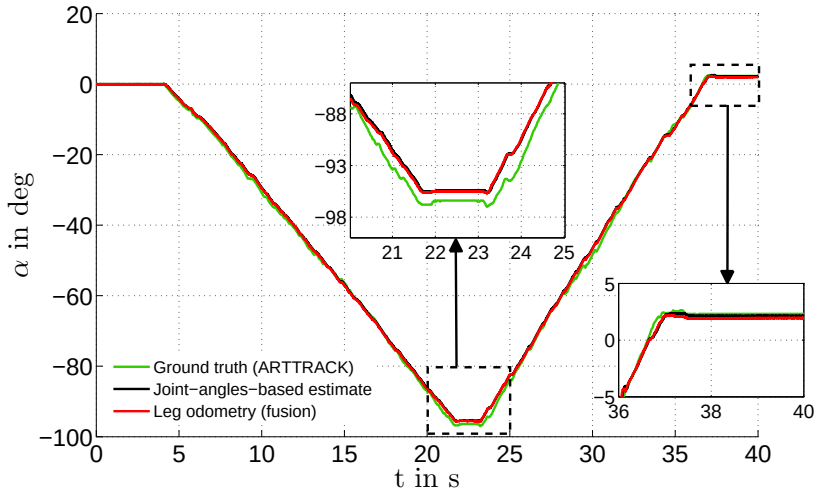


Figure 6.10: Yaw angle estimates for turning on gravel: $\dot{\alpha}_{\text{des}} = \pm 10\,^{\circ}/\text{s}$, medium joint stiffness

As for forward walking, a set of experiments was conducted to evaluate the performance of pure turning. To visualize an average result, Fig. 6.10 shows an exemplary yaw angle plot for a 90 ° right-left turn on gravel. Appropriate tuning parameters were manually identified for gravel as well as for lab floor and showed only very small dependence on the joint stiffness setting as well as the turning velocity. Thus, they are only adjusted for a change of substrate. Since the algorithm appears to be independent of velocity and stiffness setting for turning, Table 6.2 only displays the results for five trials of turning to the right on each substrate at 10 °/s and medium joint stiffness. The data shows that the yaw angle estimate on lab floor experiences a drift which amounts to 2 ° to 5 ° per 90 ° turn. For gravel, the error is smaller and shows no drift. The Cartesian position errors are very small for turning and the pitch and roll angle estimates show constant but smaller errors than for forward walking. Turning to the left shows similar error behaviour. Therefore, the performance evaluation is omitted in this place.

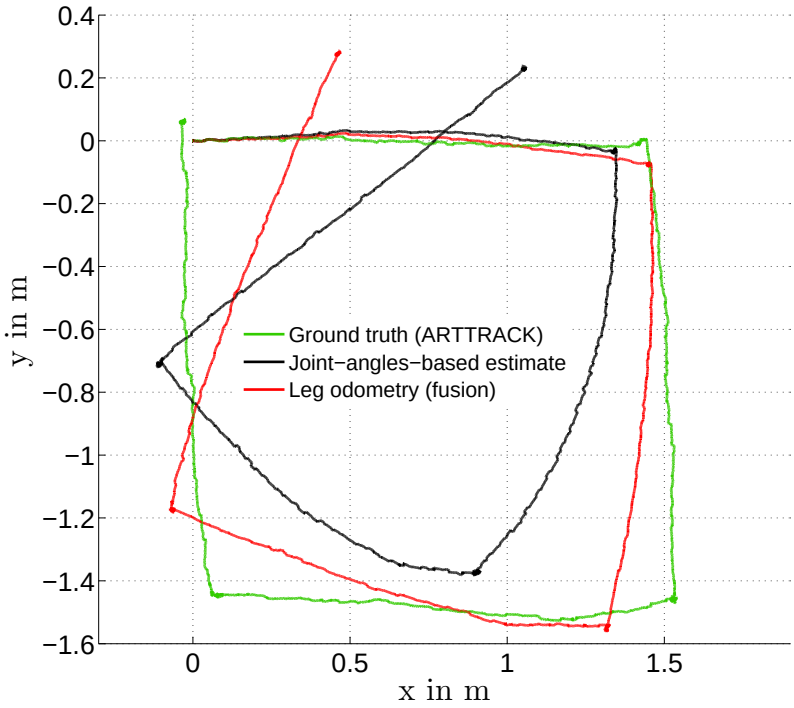
Table 6.2: Error behaviour of the complete Crawler odometry (with fusion of first and second stage) for **turning to the right**: Average translation rms errors (Δx_{rms},Δy_{rms},Δz_{rms}), **average absolute yaw angle errors ($\Delta\alpha$)** and average orientation rms errors ($\Delta\alpha_{rms}$,$\Delta\beta_{rms}$,$\Delta\gamma_{rms}$) on lab floor and gravel at $-30\,^\circ$, $-60\,^\circ$ and $-90\,^\circ$ (5 experimental runs for each ground substrate);

α_{ground_truth} in °	Δx_{rms} in cm mean (std)	Δy_{rms} in cm mean (std)	Δz_{rms} in cm mean (std)	**$\Delta\alpha$ in ° mean (std)**	$\Delta\alpha_{rms}$ in ° mean (std)	$\Delta\beta_{rms}$ in ° mean (std)	$\Delta\gamma_{rms}$ in ° mean (std)
Lab floor							
-30	0.22 (0.03)	0.10 (0.03)	0.14 (0.07)	**1.29 (0.37)**	0.92 (0.35)	1.05 (0.21)	1.47 (0.53)
-60	0.34 (0.04)	0.16 (0.04)	0.23 (0.10)	**2.48 (0.46)**	1.53 (0.31)	0.93 (0.11)	1.27 (0.37)
-90	0.37 (0.03)	0.31 (0.08)	0.31 (0.10)	**3.75 (0.51)**	2.22 (0.33)	0.97 (0.08)	1.22 (0.32)
Gravel							
-30	0.37 (0.25)	0.44 (0.27)	0.34 (0.26)	**0.73 (1.91)**	1.15 (0.86)	1.50 (0.36)	1.92 (0.43)
-60	0.48 (0.25)	0.65 (0.26)	0.52 (0.27)	**1.00 (2.15)**	1.33 (0.80)	1.31 (0.21)	2.07 (1.02)
-90	0.65 (0.36)	0.90 (0.35)	0.72 (0.39)	**0.90 (1.88)**	1.47 (0.92)	1.29 (0.18)	1.94 (0.90)

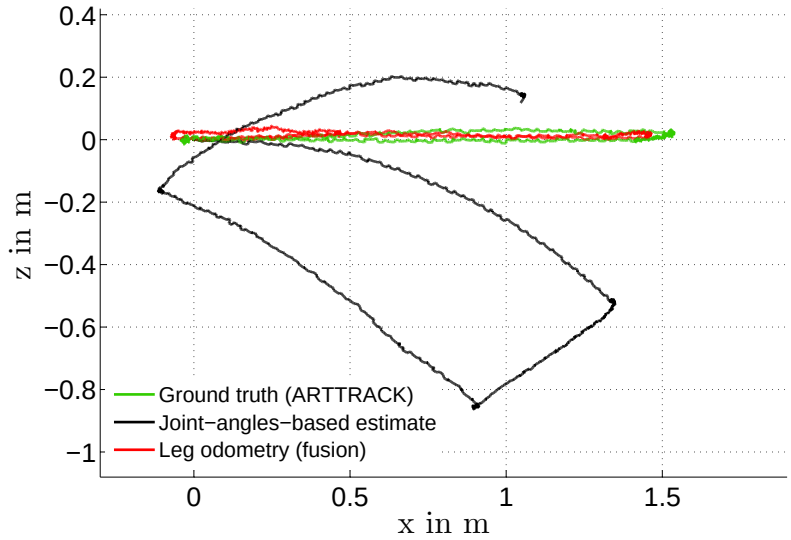
6.3.5 Combined Motions

To further evaluate the algorithm, this section gives a few examples of combined motions. Figure 6.11 shows an average result for walking a rectangular path on gravel. As can be seen, the absolute Cartesian position error on gravel is larger than on lab floor as shown in Fig. 6.3 above. In both cases the absolute Cartesian position error is mainly caused by errors of the yaw angle estimate. On lab floor the estimated path opens slightly outwards, which is a result of underestimating the yaw angle during the turning motions while it is estimated quite accurately for the forward walking parts. The opposite happens on gravel where the rectangular path is bent inwards. Here, the yaw estimates for the turning motions are quite accurate while the yaw angle estimate experiences a drift for the forward walking segments. In both cases, the behaviour is consistent with the performance results obtained for forward walking and turning. Even with the high number of steps that the robot takes while walking a distance of several meters, the z-coordinate estimate remains close to the GTMS value.

The final experiment shows an average result for a path that combines forward, upward, and turning motions, and includes a downward step as an additional challenge. In this case, the robot walks uphill along a short $15\,^\circ$ slope, moves sideways, and turns $90\,^\circ$ to the left on top of a plateau. Then it crosses a 5 cm edge and walks up a second $15\,^\circ$ slope onto a second plateau as depicted in Fig. 6.12. Along its path the odometry algorithm uses the torque thresholds and the process noise covariance matrix

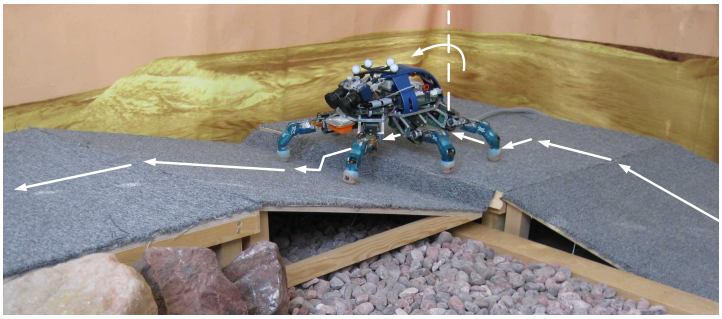


(a) Path of the DLR Crawler projected onto the xy-plane

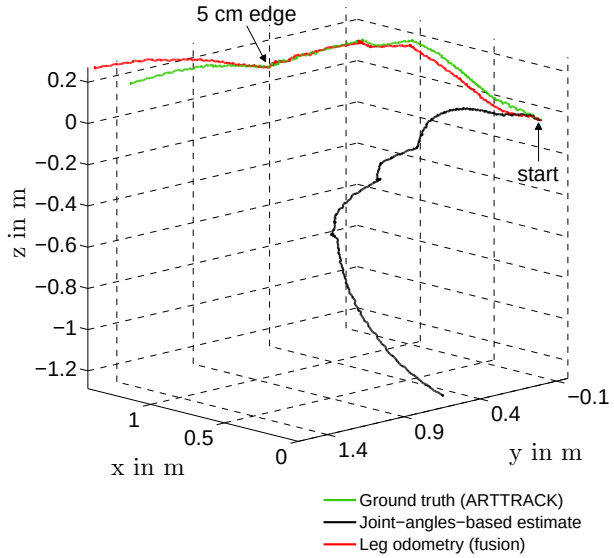


(b) Path of the DLR Crawler projected onto the xz-plane

Figure 6.11: The DLR Crawler walks along a rectangular path on gravel ($v_{des} = 6\,cm/s$, medium joint stiffness): Comparison of the COM position estimates of the first stage of the odometry algorithm, the complete leg odometry (data fusion) as well as the ground truth measurement system



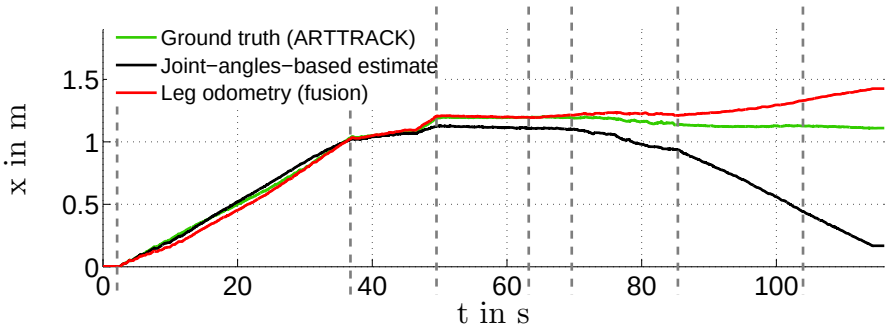
(a) DLR Crawler in the test area



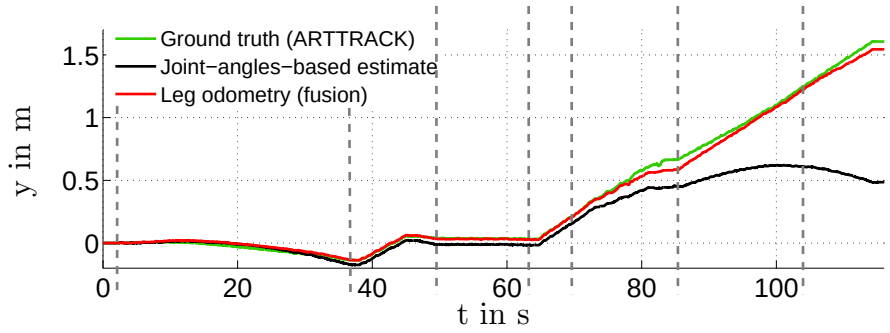
(b) Estimated and measured trajectories in xyz

Figure 6.12: Crawler odometry for walking uphill ($v_{des} = 6\,cm/s$, high joint stiffness)

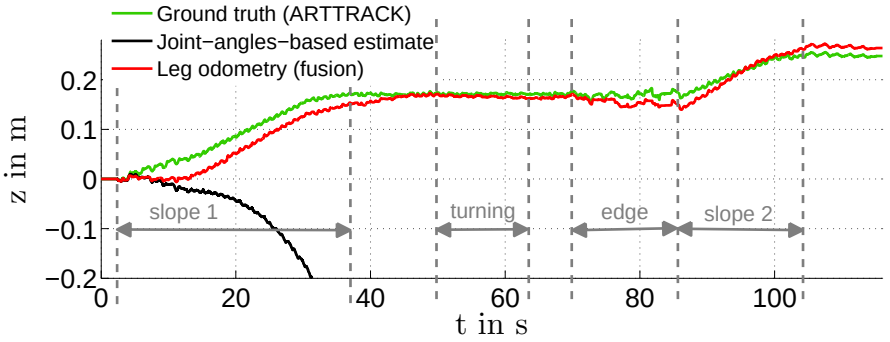
identified for walking with a tripod gait on lab floor and switches to the parameter set for walking with a tripod gait on gravel while crossing the edge. This switch is initiated automatically by stronger reflex activations and height differences within the legs and improves the estimate while crossing the edge. Again, the main source of the Cartesian position error is the yaw angle estimate that experiences a drift, especially while walking along the slopes. The overall z-coordinate estimate shown in Fig. 6.13 is quite good, even though it misses the onset of the upward motion and slightly overestimates it thereafter due to experiencing slip along the slope. The pitch angle estimate shown in Fig. 6.14 detects the slopes and closely represents the shaky motion while crossing the edge. In summary, the odometry results are very encouraging, especially since the algorithm only employs the proprioceptive sensors of the legs.



(a) x-position of the DLR Crawler

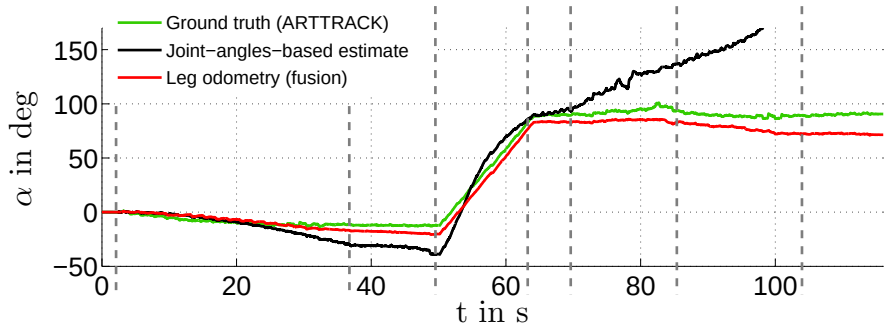


(b) y-position of the DLR Crawler

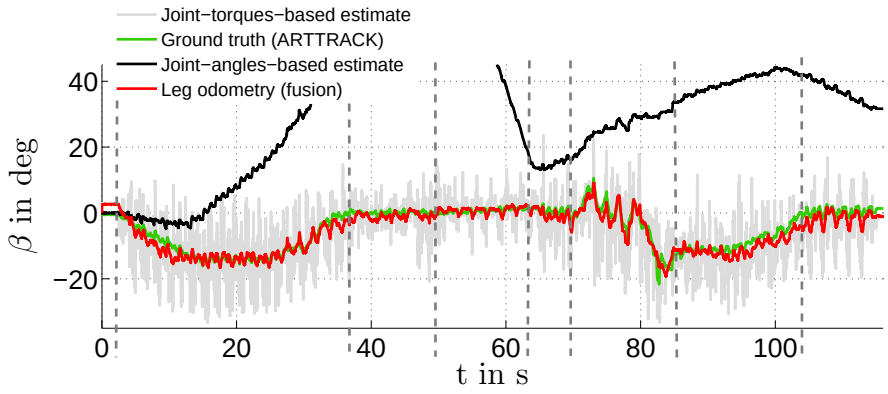


(c) z-position of the DLR Crawler

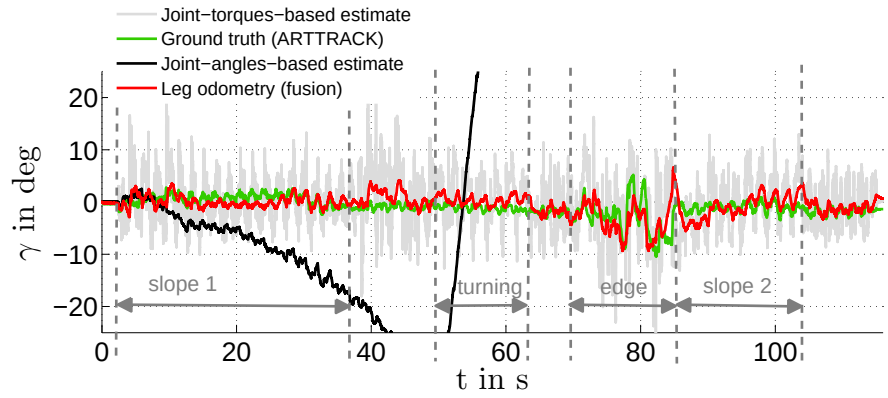
Figure 6.13: Crawler odometry COM position estimates for walking uphill ($v_{des} = 6\,cm/s$, high joint stiffness)



(a) Yaw angle of the DLR Crawler



(b) Pitch angle of the DLR Crawler



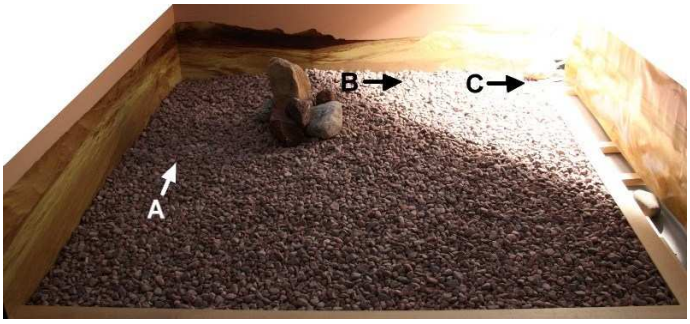
(c) Roll angle of the DLR Crawler

Figure 6.14: Crawler odometry Euler angles estimates for walking uphill ($v_{des} = 6\,cm/s$, high joint stiffness)

6.3.6 Leg-Odometry-Enhanced, Vision-Based Pose Estimation

The leg odometry algorithm presented within this chapter is a valuable source of information for any higher-level algorithm. However, being an incremental method, errors accumulate over time. Therefore, the approach is only useful within certain spatial and temporal bounds. For this reason, the leg odometry is integrated into the visual navigation framework of the DLR Crawler, wherein it serves as an aiding sensor for robust pose estimation. Since obstacle avoidance and navigation algorithms require accurate pose estimates, the fusion of pose data from multiple sources will improve their performance. In case of the DLR Crawler, an indirect feedback information filter was chosen to fuse the data provided by its IMU, its visual odometry, and its leg odometry [Chilian et al., 2011]. The information filter used is numerically equivalent to a Kalman filter but has inverse mathematical properties. Similar to the error-state Kalman filter chosen for the leg odometry, an indirect version of the information filter was selected. Working on an error model, this approach greatly simplifies the computation as the filter does not require a nonlinear dynamics model of the robot. Instead it works on linear equations that describe the state error propagation within the system. Considering leg and visual odometry as aiding sensors the filter is easily expandable to further sources, if they become available. As none of the used sensors is perfect, the pose estimation process tries to exploit their strengths while it eliminates their deficits. For example, an IMU usually has a high update rate and is independent of environmental influences, but its orientation and position output suffer a strong drift due to the double integration of acceleration signals. In contrast, visual odometry experiences only small drift but has low update rates and requires a sufficiently textured environment. The leg odometry is somewhere in between as it has a quite high update rate but is less accurate and experiences stronger drift. Comparing visual odometry to leg odometry, they are complementary. The visual odometry works well in rough terrain, which provides good texture but causes the legs to slip, while the leg odometry works best on smooth ground, which provides little texture for the visual odometry.

Within this multisensor fusion process developed in [Chilian et al., 2011], the leg odometry increases the robustness of the overall pose estimate with respect to visual disturbances. For this purpose, the visual odometry returns an error measure that influences the weights of the information filter. These adjust the filter such that it increasingly trusts the leg odometry in case of poor visual performance. On the other hand, the leg odometry invalidates its output if it detects severe slippage of single legs, and thus can be neglected by the information filter. The following experiment is an example that demonstrates the stabilising effect of the leg odometry on the overall pose estimate in the presence of visual disturbances. During this experiment,

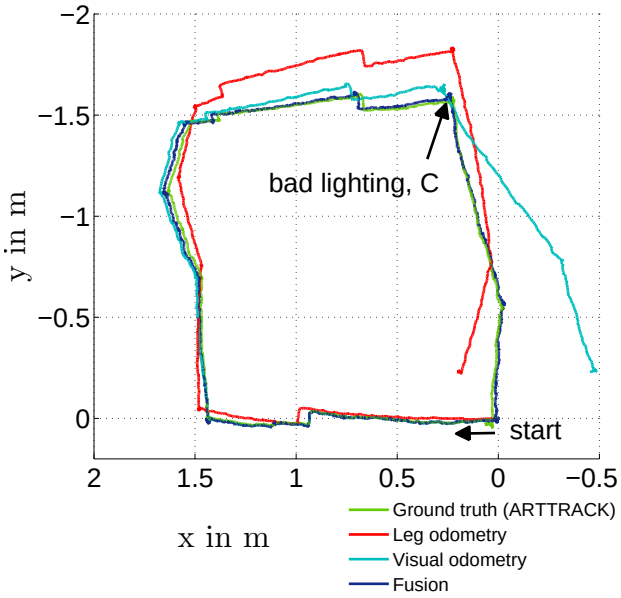


(a) Gravel testbed with one brightly illuminated area

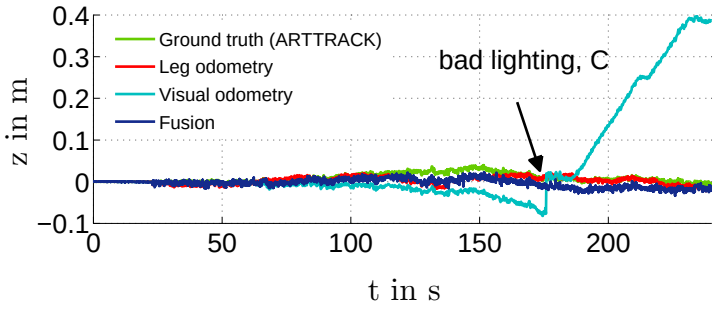
(b) Left camera view of the DLR Crawler at distinct locations

Figure 6.15: Test setup for the multisensor data fusion experiments with the DLR Crawler under varying visual conditions [Chilian et al., 2011]

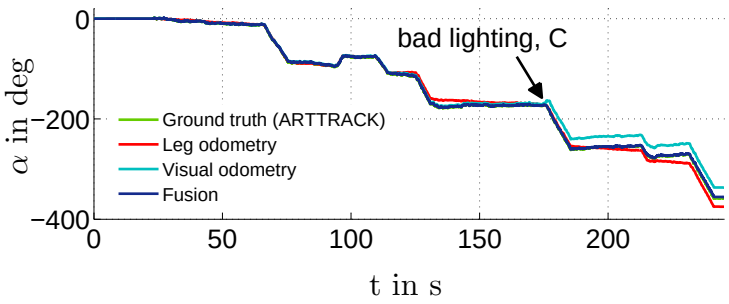
the DLR Crawler walked manually steered through the gravel testbed. Its cameras were set to fixed exposure, the lab was shaded, and a single corner of the testbed was illuminated by a bright, slightly downward-pointing light, as depicted in Fig. 6.15(a). Reaching this highly lit corner "C", the robot was blinded as shown by the exemplary left camera image in Fig. 6.15(b). Within this area, the camera images did not provide enough texture for proper visual odometry, which returned a high error value. This error value then caused the information filter to shift its weights towards an increased influence of the leg odometry. Therefore, the overall pose estimate was not affected by the strong errors of the visual odometry, as shown in Fig. 6.16(a). Once the visual conditions improved, the weights of the fusion process changed towards an increased influence of the visual odometry estimate. A comparison with the GTMS trajectories shows, that the fusion result was highly accurate throughout the entire trial, even though the robot experienced slippage and severe visual disturbances. This result demonstrates that multisensor data fusion strongly increases the robustness of the overall pose estimate, which is of great importance for navigating in environments without external references.



(a) Robot trajectory within the xy-plane



(b) Body height with respect to time



(c) Yaw angle with respect to time

Figure 6.16: Multisensor-data-fusion-based pose estimates of the DLR Crawler for an experimental run under varying visual conditions [Chilian et al., 2011]

7 Discussion

Aiming at the development of six-legged robotic scouts for disaster recovery and exploration missions, this thesis introduced a set of algorithms that constitute a self-contained walking layer for compliant, torque-controlled hexapedal robots. Furthermore, it proposed a novel 6-DOF leg odometry for these systems. The underlying idea of the presented work is to produce complex locomotive behaviours and sufficiently accurate pose estimates while only using simple algorithms that rely on the proprioceptive measurements provided by the legs, i. e. joint angles and joint torques.

7.1 Robust, Adaptive, and Leg-Loss-Tolerant Hexapedal Walking

With respect to hexapedal locomotion, this thesis demonstrates that such a set of simple algorithms is well suited to produce robust, adaptive, and leg-loss-tolerant walking in torque-controlled hexapods. The presented gait coordination adopts the first three inter-leg coordination mechanisms that Cruse and collaborators identified for the stick insect [Cruse, 1990]. Differing from the original findings as well as Espenschied's implementation of Cruse's rules [Espenschied, 1994, Espenschied et al., 1996], this work uses three important adjustments. First, similar to [Ferrell, 1995], it removes the timing component of mechanism 2. Therefore, each stance leg encourages its anterior neighbour throughout the complete stance phase to initiate a step. This strongly enforces forward-directed waves of protractions, which produce polygons of support that provide a large margin of stability across the entire range of walking speeds (see Chapter 4). Next, this thesis introduces a new coordination rule in order to prevent simultaneous stepping of adjacent legs, which is a rare but possible event when applying Cruse's rules. This simple rule, called mechanism 0, bidirectionally connects neighbouring legs and strictly inhibits the stance-to-swing transition of a leg as long as any of its neighbours has not finished its step. Independently of this thesis work, [El Sayed Auf, 2010] presents a similar inhibition rule for the six-legged

walking robot Oscar, where it serves as the only mean of gait coordination. While El Sayed Auf realises different gaits by adjusting the duty cycle of the legs, i. e. the relation of stance-to-swing time, the combination of such a rule with Cruse's first three mechanisms results in a higher variability of the emerging gait. The third difference of the coordination algorithm presented in this thesis is the novel binary safety value that implements either a timing- or a contact-force-induced swing-to-stance transition. This safety value enables higher-level algorithms to influence the walking layer such that the robot takes greater risks in uneven terrain. Consequently, the robot is able to descend steps that are larger than its walking height.

In addition to the gait coordination algorithm, the walking layer developed in this thesis employs three basic leg reflexes, the stretch reflex, the elevator reflex, and the search reflex. For these reflexes, the availability of joint torque measurements is one of the main advantages in comparison to related work presented in [Espenschied et al., 1996, El Sayed Auf, 2010, Bartsch et al., 2012, Rönnau et al., 2014, Goldschmidt et al., 2014]. While the use of reflexes is quite common in multi-legged robots, the use of joint torque measurements for triggering and monitoring them is not. Collision detection, for example, benefits of the high sensitivity that joint torque sensors provide along the entire leg in comparison to the limited capabilities of sensors located at the feet. Furthermore, based on multiple sensors within each leg, the collision detection gains redundancy with respect to sensor failure, and thus delivers more reliable results. Only in [Paskarbeit et al., 2015] the authors report a similar approach in employing joint torque estimates for collision detection. In their case the joint torque estimates are obtained from the deflection of elastomer springs integrated within the joints to provide passive compliance. While the gait coordination presented in this thesis is responsible for temporal adaptations of the stepping sequence, the reflexes mainly account for spatial variations of the terrain. However, extending the swing phase of individual legs, elevator and search reflex also influence the gait timing. Thus, the interaction with the environment couples reflex and gait coordination activity, and therefore produces complex spatio-temporal motion patterns. Another important aspect of the presented approach is the fact that torque-threshold-based stretch reflexes are sufficient to generate a terrain following behaviour in small robots without requiring active posture control (see Sections 4.4 and 4.5). This significantly reduces computational efforts while allowing the robot to effectively cross uneven terrain. Of course, there is a limit to this behaviour at which posture control or active force distribution control is inevitable to effectively master steep slopes ($> 35°$ in case of the DLR Crawler) or to prevent slippage in critical terrain where a fall might cause the loss of the robot.

In contrast to the common use of reflexes in hexapedal robots, the use of joint compliance controllers with an underlying joint torque control loop, as presented within this thesis, is unique. The strength of this approach is its ability to adapt the behaviour of the robot in a very simple and intuitive way without using computationally expensive inverse dynamics models. Additionally, the controller does not require accurate knowledge of the system parameters, which are hard to determine. The underlying joint torque controller further enables fast reactions (within 1 ms) and strongly limits the interaction forces during collisions. Especially in the case of hard contacts, this approach outperforms force-control methods relying on underlying position control loops as investigated in [Albu-Schäffer and Hirzinger, 2002]. Another advantage of the joint compliance controller is its robustness with respect to disturbances. The implemented controller never got unstable throughout all experiments. Even placing a foot on loose stones does not destabilise the control. Furthermore, the joint compliance controller reduces internal forces that result from uncertainties of the geometric parameters. Therefore, it prevents excessive slippage, which is often experienced in purely position controlled hexapods. In addition, the controller allows the robot to comply with natural terrain and to handle small elevation differences without reflex activity. Accordingly, the joint compliance controller achieves more balanced leg loads while not using an active leg force distribution algorithm. Related to this approach, but involving higher computational efforts and an underlying position control loop, two very recent examples employ virtual model control to achieve balanced leg loads in uneven terrain [Paskarbeit et al., 2015, Bjelonic et al., 2016]. In both cases the stance legs are treated as individual virtual mass-spring-damper systems and their foot positions are adjusted such that the robot body moves towards an equilibrium configuration.

Another important property of the walking layer has been introduced in Chapter 5. Therein, the gait coordination is extended to allow for the adaptation to the loss of single legs. Following the detection of such an event, the walking layer quickly (within 1 ms) reorganises the neighbourhood relations of the functional legs in order to produce proper gaits for the impaired robot. After a short transition phase of approximately 5 to 15 s the adapted gait pattern has fully emerged. A detailed simulation study shows that the range of possible velocity commands reduces to approximately one half of the completely functional robot. However, within these limits the gait coordination preserves the beneficial forward-directed waves of protractions. Similar to this approach, simple adaptation schemes have been investigated by other researchers as well. [Ferrell, 1993] and [El Sayed Auf et al., 2008], for example, apply predefined topological changes to their gait coordination networks. In contrast, [Ren et al., 2015] and [Koos et al., 2013] employ learning-based methods to retune a CPG network or

to completely learn adapted motions, respectively. While the latter two approaches are very powerful, they are slow and computationally expensive. Therefore, they are currently not well suited for use with resources limited hardware. Moreover, only this thesis and the work of [El Sayed Auf, 2010] present solutions to improve the locomotion performance of the impaired robot. For this purpose, El Sayed Auf adds another local rule to each leg. This rule prevents a neighbour of a damaged leg from stepping until its next functional neighbour is at a nearby extreme position (i. e. close to PEP or AEP). As a result, the support polygon provides greater stability, but it is not clear how the gait pattern is affected. In contrast, this thesis introduced several simple measures to improve the locomotion performance, which do not interfere with the gait coordination. An extensive simulation study shows that a combination of COG shift and increased joint stiffness is most effective in reducing the leg-loss-induced rocking motions and the associated yaw drift. However, the simulations also reveal that some of the reflexes disturb the locomotion of the five-legged robot. Therefore, if possible with respect to the terrain they should be turned off. Despite these limitations, the adaptation scheme presented within this thesis reliably and effectively handles leg loss, integrates well with the biologically-inspired gait, and preserves the positive aspects of emerging gait patterns (e. g. temporal adaptation of the gait).

Compared to other approaches, the performance and variability of the presented walking layer are unique. However, a detailed comparison is difficult since the experimental platforms have very different levels of maturity with respect to the hardware used. Thus, it is not always clear to which extent a better performance is the result of algorithmic properties or just better hardware. Nevertheless, the work of [Espenschied et al., 1996] is outstanding as they present the first conclusive use of Cruse's rules in conjunction with reflexes on a real hexapedal robot. In addition, they provide valuable lesion studies investigating the influence of single coordination mechanisms. The approach presented by Manoonpong is very promising since it addresses a rich set of capabilities by using inter-connected CPGs that integrate proprioceptive (joint angles only) and exteroceptive sensory feedback [Manoonpong, 2007]. His research group is also one of the few that investigates adaptation to leg loss [Ren et al., 2015] as discussed above. [El Sayed Auf, 2010] employs the so-called organic computing framework to enable hexapedal walking while only using a set of very simple rules. Based on this approach, the robot Oscar realises different gaits, reflex-based obstacle crossing, and is able to adapt to leg loss. However, the results show less variability and locomotive performance than those presented within this thesis. Nevertheless, they are difficult to compare due to different levels of hardware integration. Anyway, the computational effort involved with Oscar is very low, and thus

provides a great advantage for highly resource limited applications. While those simple algorithms enable hexapods to cross a large variety of terrains, multi-legged robotic locomotion across difficult obstacles currently still requires computationally expensive foot point planning as investigated, for example, in [Belter, 2013].

However, considering the presented walking layer and the related experimental results, this work clearly verifies the first hypothesis of this thesis that simple and distributed algorithms, which only use the measurements provided by the proprioceptive sensors of the legs, are able to generate robust, adaptive, and leg-loss-tolerant walking in torque-controlled hexapedal robots. The performance of the DLR Crawler during at least one hundred hours of operation (mostly during navigation experiments) is very convincing. During these experiments the robot smoothly moved across natural terrain and negotiated obstacles within its walking height autonomously. The approach in general is highly adaptive to uneven ground and to the loss of single legs while being robust with respect to disturbances experienced by the emerging gait (i. e. negotiation of obstacles) or the joint controller. Additionally, most of the presented work is not limited to the hardware concept of the DLR Crawler but also applies to robots with serial elastic actuation and joint springs serving as torque sensors. Furthermore, the presented walking layer easily extends to eight-legged robots as demonstrated in [Görner and Hirzinger, 2010].

7.2 Pose Estimation in Torque-Controlled Hexapods

Regarding pose estimation in torque-controlled hexapods, this work proposed and evaluated a full 6-DOF odometry that only uses the proprioceptive sensors of the legs. Employing an optimisation-based approach, the first stage of the algorithm incrementally computes the robot pose from joint angles while handling the varying ground contact configurations immanent to an emergent gait. The novel fusion of the resulting pitch and roll angle estimates with absolute joint-torque-based values strongly improves the overall result of the pose estimation process. Not requiring a complex motion model of the robot, the error-state Kalman filter represents a simple but powerful algorithmic solution. With respect to the pose computation by its first stage, the odometry algorithm shows similarities to the approaches used for the robots Ambler [Roston and Krotkov, 1992] and Lauron [Gassmann et al., 2005b, Gassmann et al., 2005a]. However, the main difference of the method developed within this thesis is that it does not require an IMU to obtain a complete 6-DOF pose estimate that incorporates absolute pitch and roll information. Moreover, the method can be applied to stiff and compliant walking robots and is able to handle any statically

stable emerging gait. Compared to most of the previous work, the algorithm has been extensively tested and its performance is documented for the DLR Crawler walking on different ground substrates, at different joint stiffness settings, and at different walking velocities, and thus with various emerging gaits. In all of those experimental trials, the Cartesian endpoint error remained within a range of 1 to 5 %. This is a remarkable accuracy for a multi-legged robot that does not use an IMU at this stage. Thus, for walking, the experimental results clearly validate the second hypothesis of this work that leg proprioception alone is sufficient to estimate the pose of a torque-controlled hexapod with respect to its starting point.

Clearly, the presented approach has several limitations as well. First of all, the odometry is only applicable to statically stable walking robots, which was the focus of this work. During dynamic locomotion, pose estimation requires IMU data to cover flight phases as presented in [Reinstein and Hoffmann, 2011, Reinstein and Hoffmann, 2013] and [Lin et al., 2005, Lin et al., 2006]. Additionally, for running, the gravity vector computation is not valid since the ground contact forces are not dominated by gravitational loads anymore. Another limitation of the presented approach is the drift inherent to all dead reckoning methods. This drift results in an estimation error that grows unbounded with the distance travelled, and therefore limits the useful range of the odometry. However, since the leg odometry is intended to serve as an aiding sensor in a multi-modal sensor fusion scheme for localisation, this drift is less important. Within this framework, the leg odometry provides redundancy and strongly increases the robustness of the pose estimate with respect to IMU failures and temporary disturbances of the visual odometry. This advantageous behaviour has been successfully demonstrated in various navigation experiments with the DLR Crawler.

8 Conclusions and Outlook

This thesis investigated the locomotion and pose estimation in compliant, torque-controlled hexapods. The work presented verifies the hypothesis that simple, distributed, and solely leg-proprioception-based algorithms are able to generate robust, adaptive, and leg-loss-tolerant walking in natural terrain. Furthermore, it verifies the hypothesis that leg proprioception alone is sufficient to provide a full pose estimate with respect to the starting point of the robot. The motivation for employing only simple algorithms is to avoid complex, and thus error-prone, model-based approaches. This is especially important for small mobile robots that operate with limited computational resources in a priori unknown environments. A key feature of the work presented is the use of joint torque measurements, enabling simple but highly robust compliance control, a sensitive detection of contacts and collisions, and pose estimation with increased accuracy.

The set of algorithms proposed constitutes a walking layer that realises omnidirectional locomotion. For this purpose, it employs a flexible, biologically-inspired gait coordination, generating stable gaits with forward-directed waves of protractions. Those gaits are beneficial as they increase the static stability margin. Furthermore, the walking layer provides three basic reflexes, which enable the robot to negotiate obstacles within its walking height autonomously. Another important property of the walking layer is that it encapsulates the whole walking process, and thus facilitates interaction with higher-level algorithms using a simple interface. Apart from steering commands and state information, this interface provides a novel binary safety value. Setting this value to zero allows the higher-level algorithm to influence the obstacle-crossing behaviour of the robot such that it takes higher risks. Having lost a single leg, the walking layer is able to adapt the gait coordination within one time step. Thereafter, an adapted gait fully emerges within 15 seconds. Stiffening the joints and shifting the COG with respect to the support polygon strongly improves the locomotion performance of the five-legged robot. To inform a higher-level navigation algorithm about the locomotive capabilities of the robot, a range of feasible velocity commands has been computed for the fully functional as well as the impaired system. With respect to the

leg-proprioception-based pose estimation, this thesis demonstrates the ability of the algorithm to handle emerging gaits with varying ground contact configurations. Moreover, it proves that joint-torque-based attitude information stabilises and improves the overall pose estimate.

Despite of all useful properties, the presented approach also has several limitations. At first, it is only applicable to smaller robots that provide sufficiently strong legs to endure short periods of asymmetric, high loads. Large robots, in contrast, often require a careful distribution of ground reaction forces as their legs can only support smaller portions of their weight. Next, the walking layer requires torque measurements, and thus the integration of additional sensors. However, the hardware effort is justified by its substantial benefits on locomotion and localisation. Furthermore, it is important to note that the walking layer only covers one mode of locomotion, i. e. slow, statically stable walking in uneven terrain. With respect to complex exploration scenarios, future robots should also provide two additional modes: energetically efficient running on uneven ground, and force-controlled climbing in very rugged terrain. Similar to walking, those two should not require computationally complex model-based controllers, but should employ simple, distributed solutions as well. The running mode, for example, could exploit self-stabilising mechanics that are driven by feedforward control. For climbing the problem is much harder as it requires a careful distribution of ground reaction forces. Apart from those different modes of locomotion, a future exploration robot should also provide the capability to manipulate its environment using its legs or its body. As for climbing, manipulation requires the careful control of interaction forces. Being structurally related, a common approach should ideally solve both problems. Using tactile sensors, the legs could also actively explore the environment using the sense of touch. Doing so, they could create sparse 3D maps under visually bad conditions while simultaneously estimating the properties of the local terrain.

To conclude, this thesis demonstrates in simulations and experiments with the DLR Crawler that simple, distributed, and solely leg-proprioception-based algorithms are capable to generate robust, adaptive, and leg-loss-tolerant walking in torque-controlled hexapods. In addition, it shows that leg proprioception is sufficient for these robots to obtain a full pose estimate. Hence by investigating torque-controlled hexapedal walking and by adding new and improved capabilities to these systems, this thesis contributes to the ongoing process of creating autonomous six-legged robots for future exploration missions.

A Parameters of the Simplified Dynamics Model

Table A.1: Parameters of the simplified dynamics model of the DLR Crawler

Parameter	Description	Value
	Body	
m_b [kg]	mass of the robot body	3.66
$J_\mathrm{b,xx}$ [kgm^2]	mass moment of inertia about the x axis of the robot body	0.0153
$J_\mathrm{b,yy}$ [kgm^2]	mass moment of inertia about the y axis of the robot body	0.0221
$J_\mathrm{b,zz}$ [kgm^2]	mass moment of inertia about the z axis of the robot body	0.0313
l_b [m]	length of the robot body in between the front and hind leg	0.25
b_b1 [m]	width of the robot body in between the middle legs	0.22
b_b2 [m]	width of the robot body in between the front/hind legs	0.176
α_leg [°]	downward tilt of the leg base wrt the body xy-plane	15
β_leg [°]	angle of the front/hind legs wrt the body xz-plane	50
	Leg	
l_1 [m]	length of the proximal segment of the leg	0.075
l_2 [m]	length of the medial segment of the leg	0.04
l_3 [m]	length of the distal segment of the leg	0.04
k_1 [Nm/rad]	low(high) joint stiffness value of the protraction/retraction DOF	8(16)
k_2 [Nm/rad]	low(high) joint stiffness value of the elevation/depression DOF	6(12)
k_3 [Nm/rad]	low(high) joint stiffness value of the extension/flexion DOF	6(12)
d_1 [Nms/rad]	joint damping of the protraction/retraction DOF	0.2
d_2 [Nms/rad]	joint damping of the elevation/depression DOF	0.2
d_3 [Nms/rad]	joint damping of the extension/flexion DOF	0.1
	Ground	
μ	coefficient of friction	0.8
$v_\mathrm{slip,nom}$ [m/s]	nominal slipping velocity	1

B | Crossing Steps: Additional Simulation Results

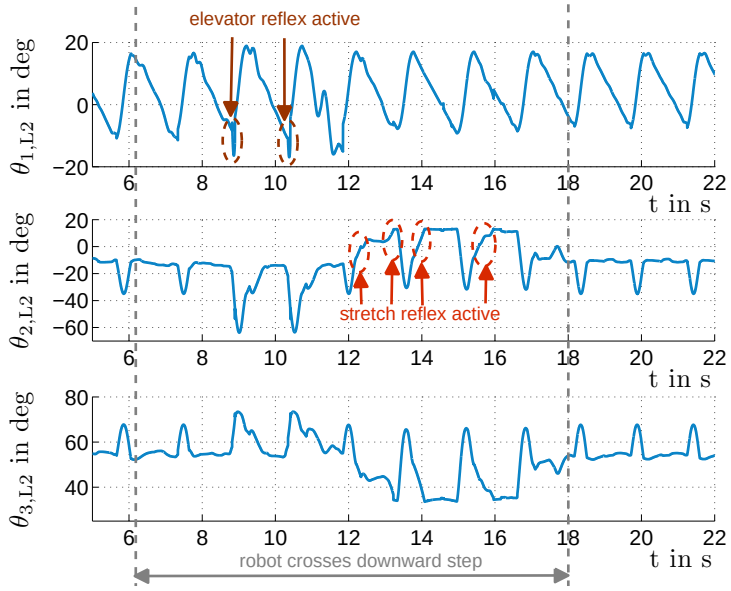


Figure B.1: Joint trajectories of the left middle leg (L2) of the simulated DLR Crawler for crossing an 8-cm-downward step; all reflexes are enabled; red ellipses mark stretch reflex activity; brown ellipses mark elevator reflex activity

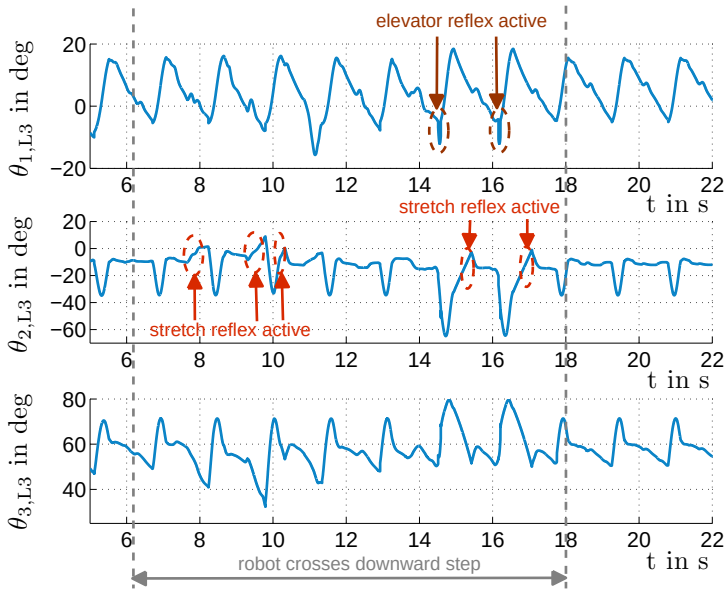


Figure B.2: Joint trajectories of the left hind leg (L3) of the simulated DLR Crawler for crossing an 8-cm-downward step; all reflexes are enabled; red ellipses mark stretch reflex activity; brown ellipses mark elevator reflex activity

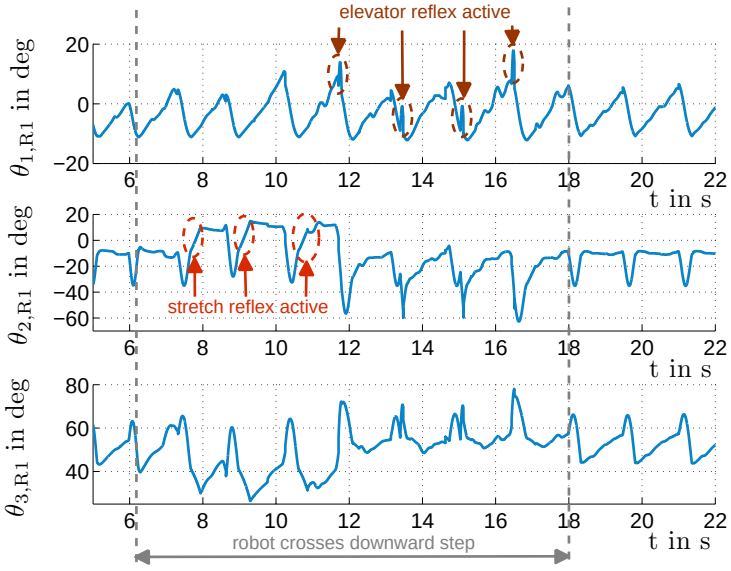


Figure B.3: Joint trajectories of the right front leg (R1) of the simulated DLR Crawler for crossing an 8-cm-downward step; all reflexes are enabled; red ellipses mark stretch reflex activity; brown ellipses mark elevator reflex activity

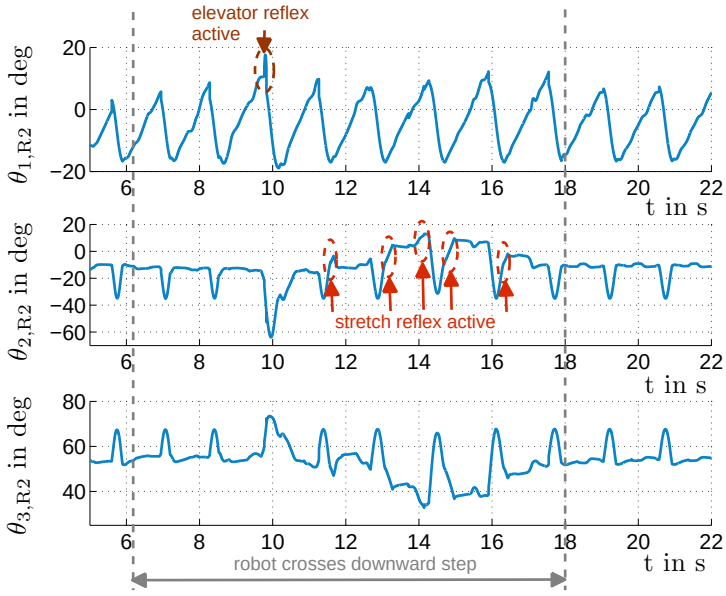


Figure B.4: Joint trajectories of the right middle leg (R2) of the simulated DLR Crawler for crossing an 8-cm-downward step; all reflexes are enabled; red ellipses mark stretch reflex activity; brown ellipses mark elevator reflex activity

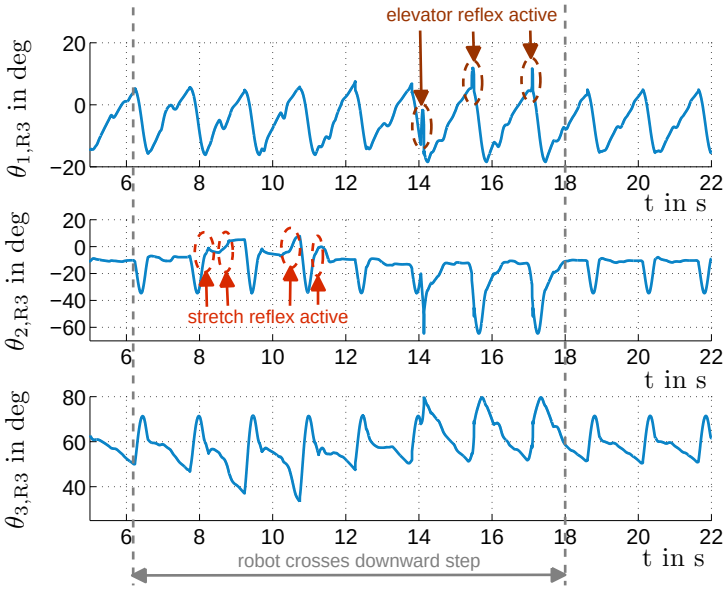
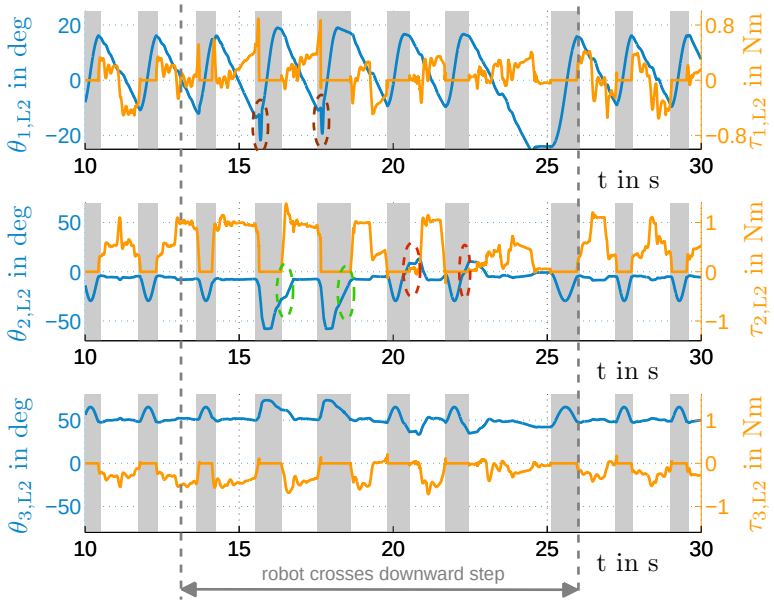
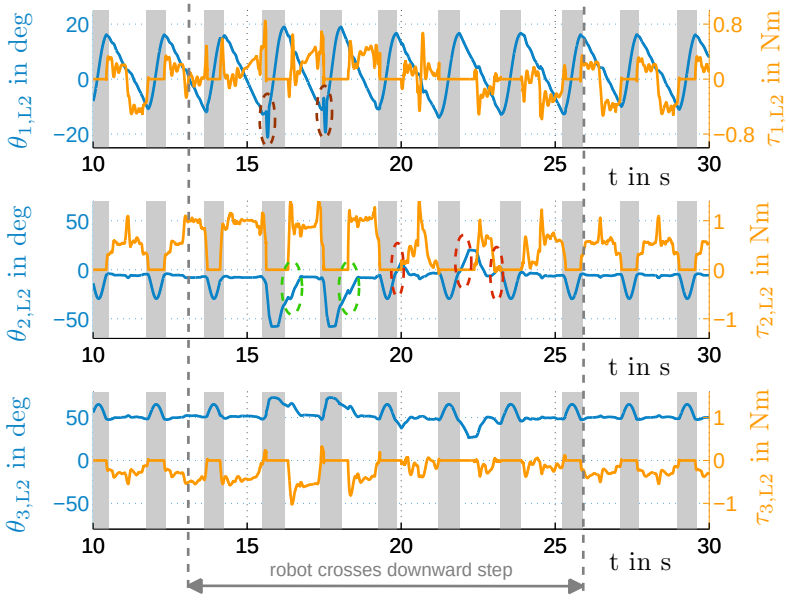


Figure B.5: Joint trajectories of the right hind leg (R3) of the simulated DLR Crawler for crossing an 8-cm-downward step; all reflexes are enabled; red ellipses mark stretch reflex activity; brown ellipses mark elevator reflex activity

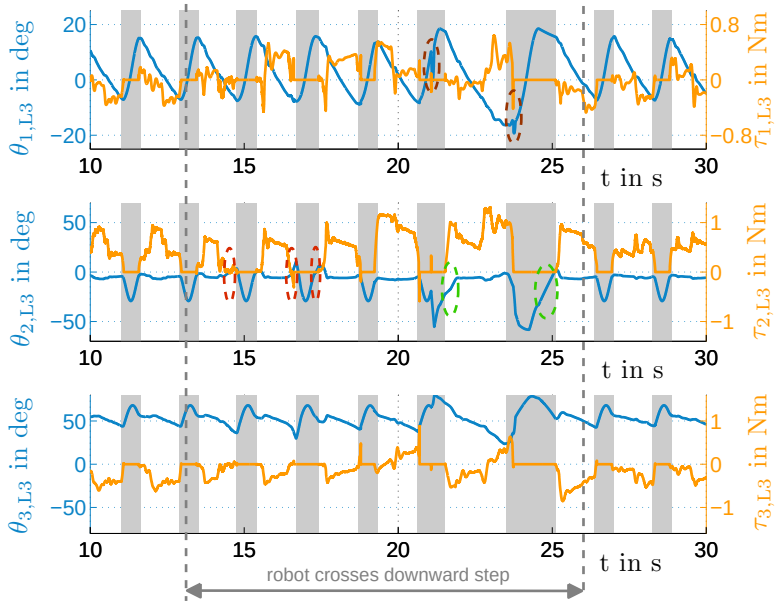


(a) Safety value: 1

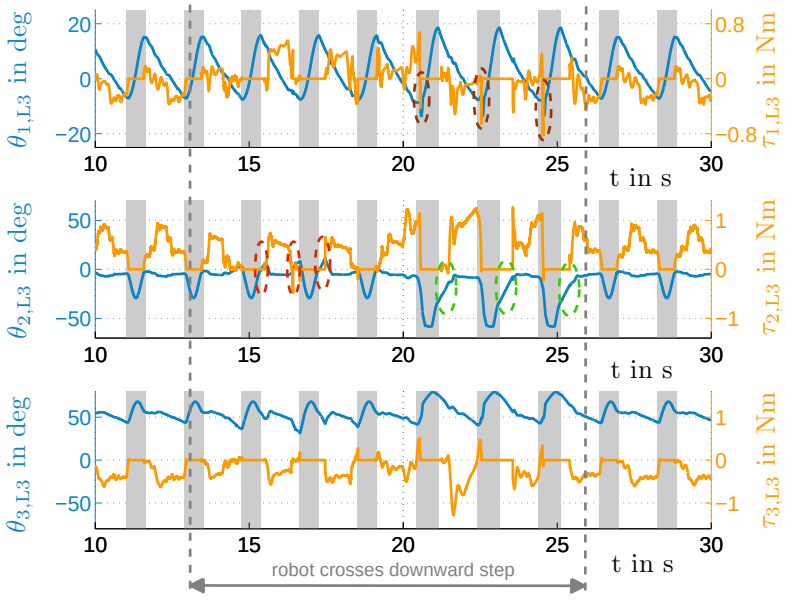


(b) Safety value: 0

Figure B.6: Joint angles (blue) and joint torques (orange) of the left middle leg (L2) of the simulated DLR Crawler for crossing a 5-cm-downward step: reflexes enabled; stretch reflex (red ellipse); elevator reflex (brown ellipse); reflex height reduction (green ellipse)

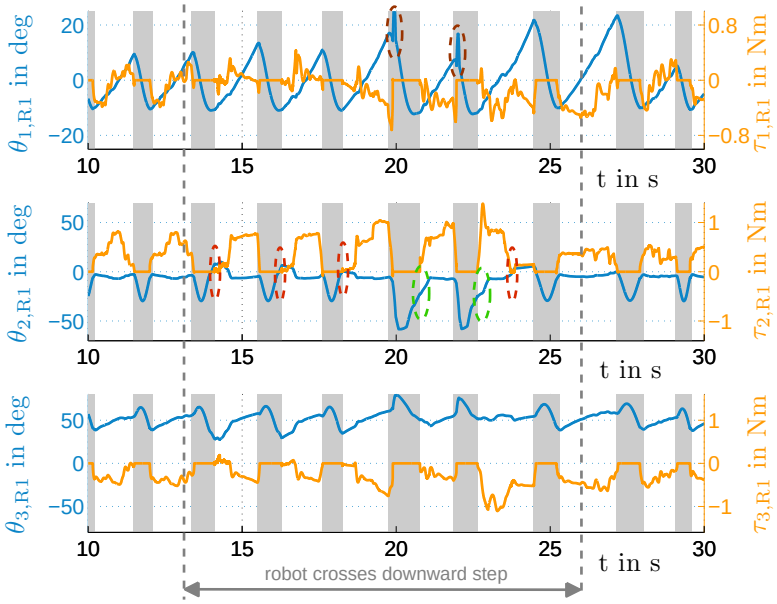


(a) Safety value: 1

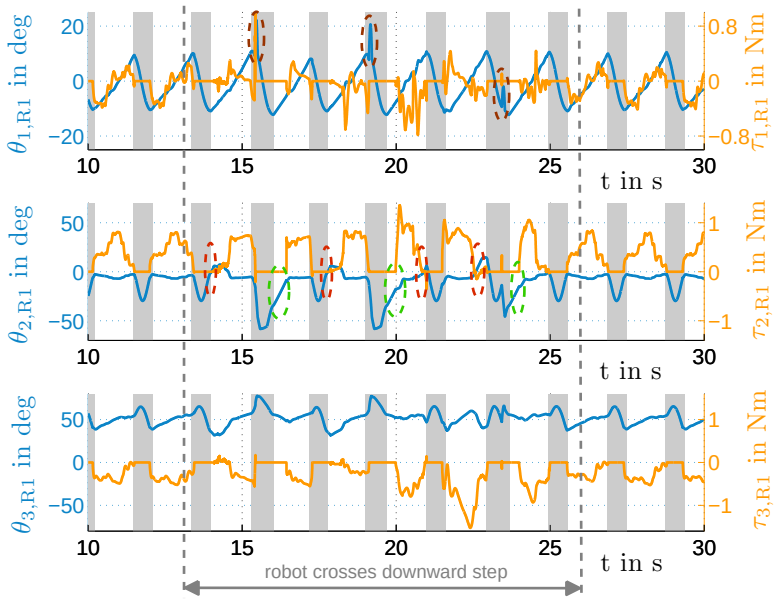


(b) Safety value: 0

Figure B.7: Joint angles (blue) and joint torques (orange) of the left hind leg (L3) of the simulated DLR Crawler for crossing a 5-cm-downward step: reflexes enabled; stretch reflex (red ellipse); elevator reflex (brown ellipse); reflex height reduction (green ellipse)

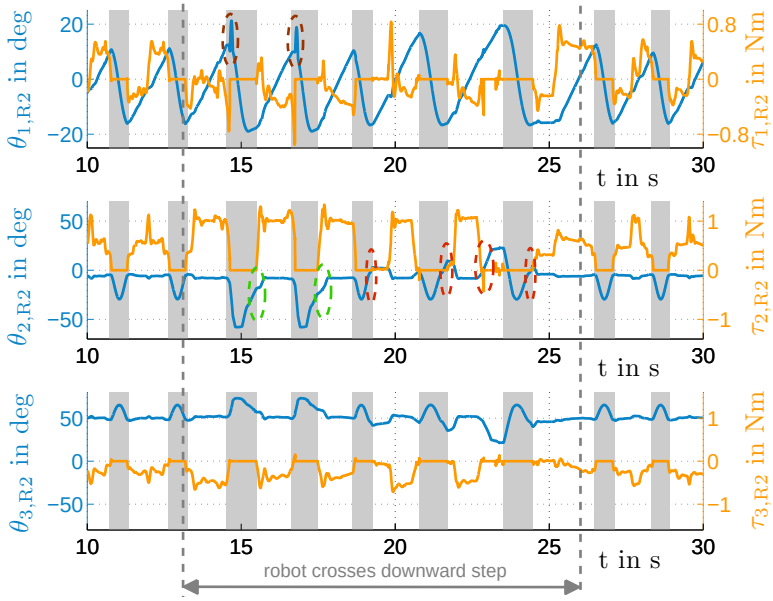


(a) Safety value: 1

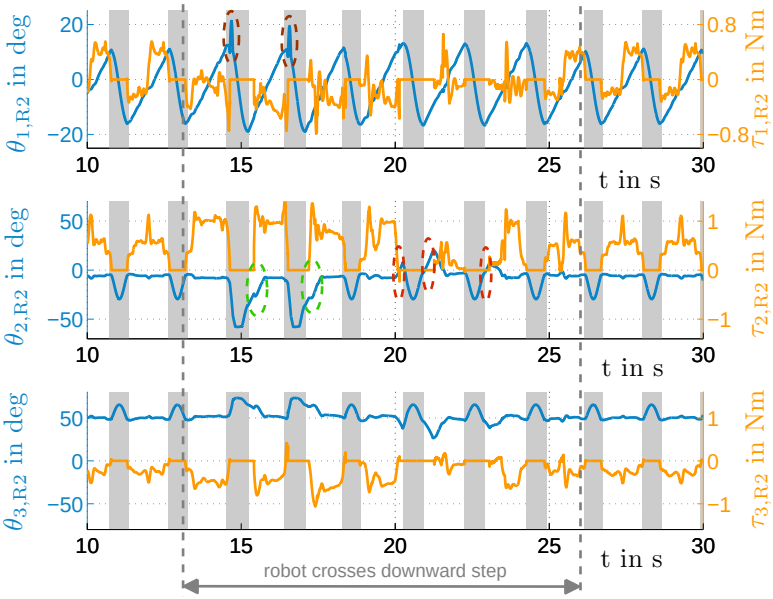


(b) Safety value: 0

Figure B.8: Joint angles (blue) and joint torques (orange) of the right front leg (R1) of the simulated DLR Crawler for crossing a 5-cm-downward step: reflexes enabled; stretch reflex (red ellipse); elevator reflex (brown ellipse); reflex height reduction (green ellipse)

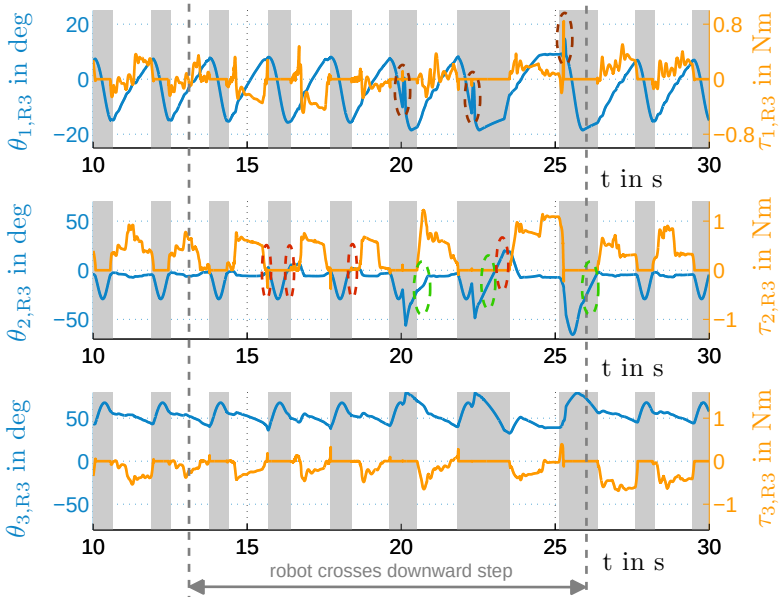


(a) Safety value: 1

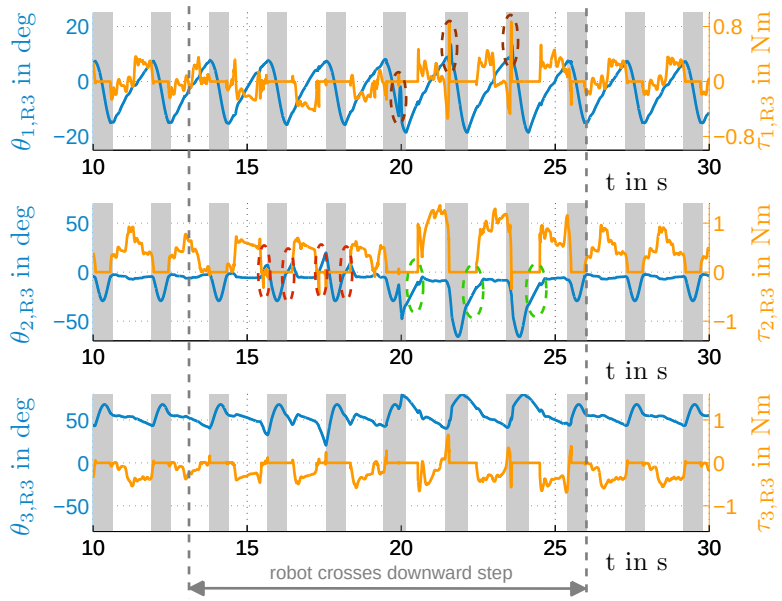


(b) Safety value: 0

Figure B.9: Joint angles (blue) and joint torques (orange) of the right middle leg (R2) of the simulated DLR Crawler for crossing a 5-cm-downward step: reflexes enabled; stretch reflex (red ellipse); elevator reflex (brown ellipse); reflex height reduction (green ellipse)



(a) Safety value: 1



(b) Safety value: 0

Figure B.10: Joint angles (blue) and joint torques (orange) of the right hind leg (R3) of the simulated DLR Crawler for crossing a 5-cm-downward step: reflexes enabled; stretch reflex (red ellipse); elevator reflex (brown ellipse); reflex height reduction (green ellipse)

C | Additional Leg-Loss Simulations

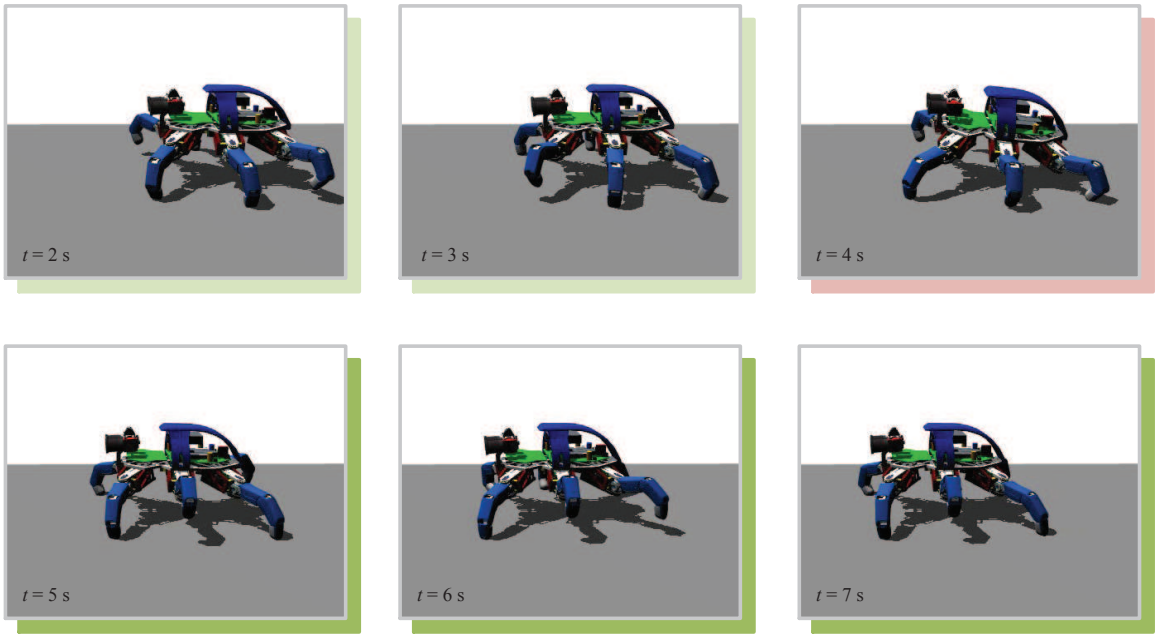


Figure C.1: Simulation of the DLR Crawler experiencing the loss of its left middle leg (L2); light green frames: the robot is fully functional, red frame: the left front leg is lost (i. e. lifted and hold in this position), green frames: the robot instantaneously adapts its gait, shifts its COG away from L1 and increases its joint stiffness settings

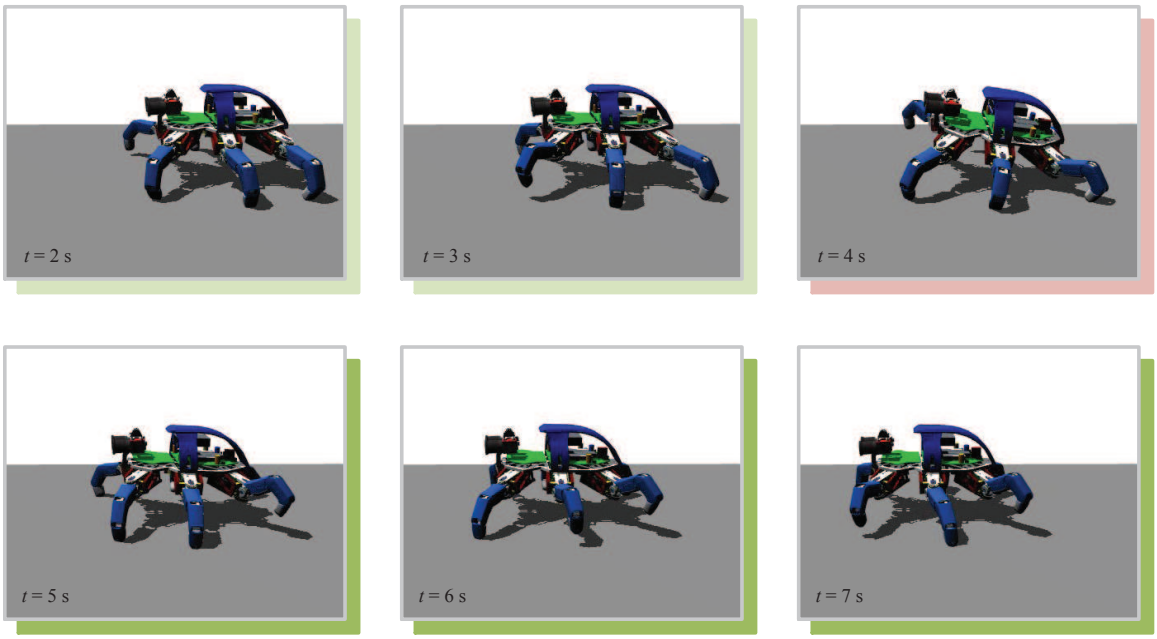


Figure C.2: Simulation of the DLR Crawler experiencing the loss of its left hind leg (L3); light green frames: the robot is fully functional, red frame: the left front leg is lost (i. e. lifted and hold in this position), green frames: the robot instantaneously adapts its gait, shifts its COG away from L1 and increases its joint stiffness settings

D | Thesis-Related Publications

Within the following list, all scientific publications are given that originate in the work summarized and presented by this thesis. The list is split in two parts - one part wherein I am the first author of the published articles and another part wherein I am the co-author. For each of the publications, a short statement is included that relates the presented results to the respective chapter of this thesis.

Publications as first author:

- Journal articles
  - M. Görner, Th. Wimböck and G. Hirzinger (2009), The DLR Crawler: Evaluation of gaits and control of an actively compliant six-legged walking robot, *Industrial Robot: An International Journal*, 36(4):344-351 (This article relates to Chapter 3 and 4 as it gives an overview of the robot, its joint compliance control, the basic gait algorithm and the reflexes used. However, the results presented within this thesis refer to an improved and extended version of the algorithms.)
  - M. Görner and A. Stelzer (2013), A leg proprioception based 6 DOF odometry for statically stable walking robots, *Autonomous Robots*, 34(4):311-326 (This article relates to Chapter 6 and presents the derivation of the algorithm as well as large parts of the respective results given in this thesis.)

- Conference articles
  - M. Görner 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 *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4728 - 4735, Anchorage, AK, USA (This article relates to Chapter 5 and presents the basic adaptation scheme as well as the influence of a leg loss on the feasible walking velocities. However, the results presented within this thesis are based on an improved and extended version of the algorithm.)
  - M. Görner, A. Chilian and H. Hirschmüller (2010), Towards an autonomous walking robot for planetary surfaces, In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, pages 170-177, Sapporo, Japan (This article gives a short overview of the robot, summarizes the gait algorithm as well as the adaptation to leg loss and presents the visual navigation scheme. Some of the navigation experiments presented in this article are included in Chapter 4 of this thesis.)

Publications as co-author:

- Journal articles
  - A. Stelzer, H. Hirschmüller and M. Görner (2012), Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain, *The International Journal of Robotics Research*, 31(4):381-402 (This article presents the complete rough terrain navigation of the DLR Crawler. It is related to Chapter 6 of this thesis, which in one section presents some of the experimental results.)

- Conference articles
  - A. Stelzer, H. Hirschmüller and M. Görner (2011), Multisensor data fusion for robust pose estimation of a six-legged walking robot, In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2497-2504, San Francisco, CA, USA (This article presents the multisensor data fusion of the DLR Crawler including, IMU, visual odometry and leg odometry. It is related to Chapter 6 of this thesis, which includes some of the experimental results under bad visual conditions.)

Bibliography

[Albu-Schäffer and Hirzinger, 2002] Albu-Schäffer, A. and Hirzinger, G. (2002). Cartesian impedance control techniques for torque controlled light-weight robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 657–663, Washington, DC, USA.

[Angle and Brooks, 1990] Angle, C. and Brooks, R. (1990). Small planetary rovers. In *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, pages 383–388, Ibaraki, Japan.

[Ayali et al., 2015] Ayali, A., Couzin-Fuchs, E., David, I., Gal, O., Holmes, P., and Knebel, D. (2015). Sensory feedback in cockroach locomotion: current knowledge and open questions. *Journal of Comparative Physiology A*, 201(9):841–850.

[Bares and Whittaker, 1990] Bares, J. and Whittaker, W. (1990). Walking robot with a circulating gait. In *IEEE International Workshop on Intelligent Robots and Systems (IROS)*, pages 809–816, Ibaraki, Japan.

[Bartsch et al., 2012] Bartsch, S., Birnschein, T., Römmermann, M., Hilljegerdes, J., Kühn, D., and Kirchner, F. (2012). Development of the six-legged walking and climbing robot spaceclimber. *Journal of Field Robotics*, 29(3):506–532.

[Belter, 2013] Belter, D. (2013). Optimization-based approach for motion planning of a robot walking on rough terrain. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 7:34–41.

[Berns et al., 1994] Berns, K., Cordes, S., and Ilg, W. (1994). Adaptive, neural control architecture for the walking machine lauron. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1172 – 1177, Munich, Germany.

[Birkmeyer et al., 2009] Birkmeyer, P., Peterson, K., and Fearing, R. (2009). DASH: A dynamic 15g hexapedal robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 418 – 419, St Louis, MO, USA.

[Bjelonic et al., 2016] Bjelonic, M., Kottege, N., and Beckerle, P. (2016). Proprioceptive control of an over-actuated hexapod robot in unstructured terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Deajeon, Korea.

[Blickhan, 1989] Blickhan, R. (1989). The spring-mass model for running and hopping. *Journal of Biomechanics*, 22:1217–1227.

[Blickhan and Full, 1993] Blickhan, R. and Full, R. (1993). Similarity in multilegged locomotion: Bouncing like a monopode. *Journal of Comparative Physiology A*, 173:509–517.

[Boaventura et al., 2012] Boaventura, T., Semini, C., Buchli, J., Frigerio, M., Focchi, M., and Caldwell, D. (2012). Dynamic torque control of a hydraulic quadruped robot. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1889–1894, Saint Paul, MN, USA.

[Brooks, 1985] Brooks, R. (1985). A robust layered control system for a mobile robot. Technical report, Massachusetts Institute of Technology.

[Butterfass et al., 2001] Butterfass, J., Grebenstein, M., Liu, H., and Hirzinger, G. (2001). DLR-Hand II: Next generation of a dextrous robot hand. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 109–114, Seoul, Korea.

[Calvitti, 2004] Calvitti, A. (2004). *Phase locking in coupled oscillators as hybrid automata*. PhD thesis, Case Western Reserve University, Cleveland, OH, USA.

[Calvitti and Beer, 2000] Calvitti, A. and Beer, R. (2000). Analysis of a distributed model of leg coordination. *Biological Cybernetics*, 82:197–206.

[Cham et al., 2002] Cham, J., Bailey, S., Clark, J., Full, R., and Cutkosky, M. (2002). Fast and robust: Hexapedal robots via shape deposition manufacturing. *The International Journal of Robotics Research*, 21(10-11):869–882.

[Chilian et al., 2011] Chilian, A., Hirschmüller, H., and Görner, M. (2011). Multisensor data fusion for robust pose estimation of a six-legged walking robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2497–2504, San Francisco, CA, USA.

[Clark and Cutkosky, 2005] Clark, J. and Cutkosky, M. (2005). The effect of leg specialization in a biomimetic hexapedal running robot. *Journal of Dynamic Systems, Measurement and Control*, 128:26–35.

[Cordes et al., 1997] Cordes, S., Berns, K., and Leppanen, I. (1997). Sensor components of the six-legged walking machine lauron ii. In *International Conference on Advanced Robotics (ICAR)*, pages 71–76, Monterey, CA, USA.

[Craig, 2005] Craig, J. (2005). *Introduction to Robotics, Mechanics and Control*. Pearson.

[Cruse, 1990] Cruse, H. (1990). What mechanisms coordinate leg movements in walking arthropods? *Trends in Neurosciences*, 13:15–21.

[Cruse et al., 1998] Cruse, H., Kindermann, T., Schumm, M., Dean, J., and Schmitz, J. (1998). Walknet - a biologically inspired network to control six-legged walking. *Neural Networks*, 11:1435–1447.

[Cully et al., 2015] Cully, A., Clune, J., Tarapore, D., and Mouret, J. (2015). Robots that can adapt like animals. *Nature*, 521:503–507.

[Delcomyn et al., 1996] Delcomyn, F., Nelson, M., and Cocatre-Zilgien, J. (1996). Sense organs of insect legs and the selection of sensors for agile walking robots. *The International Journal of Robotics Research*, 15(2):113–127.

[Dürr et al., 2004] Dürr, V., Schmitz, J., and Cruse, H. (2004). Behaviour-based modelling of hexapod locomotion: linking biology and technical application. *Arthropod Structure and Development*, 33:237–250.

[El Sayed Auf, 2010] El Sayed Auf, A. (2010). *Eine Organic Computing basierte Steuerung für einen hexapoden Laufroboter unter dem Aspekt reaktiver Zuverlässigkeit und Robustheit*. PhD thesis, Universität zu Lübeck, Lübeck, Germany.

[El Sayed Auf et al., 2008] El Sayed Auf, A., Litza, M., and Maehle, E. (2008). Distributed fault-tolerant robot control architecture based on organic computing principles. In Hinchey, M., Pagnoni, A., Rammig, F., and Schmeck, H., editors, *Biologically-inspired collaborative computing (IFIP 20th World Computer Congress, Second IFIP TC 10 International Conference on Biologically-Inspired Collaborative Computing)*. Springer, US.

[Espenschied, 1994] Espenschied, K. (1994). *Biologically-inspired control of an insect-like hexapod robot on rough terrain*. PhD thesis, Case Western Reserve University, Cleveland, OH, USA.

[Espenschied et al., 1996] Espenschied, K., Quinn, R., Beer, R., and Chiel, H. (1996). Biologically based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot. *Robotics and Autonomous Systems*, 18:59–64.

[Ferrell, 1993] Ferrell, C. (1993). Robust agent control of an autonomous robot with many sensors and actuators. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

[Ferrell, 1995] Ferrell, C. (1995). A comparison of three insect-inspired locomotion controllers. *Robotics and Autonomous Systems*, 16:135–159.

[Frik et al., 1999] Frik, M., Guddat, M., Karatas, M., and Losch, D. (1999). A novel approach to autonomous control of walking machines. In *International Conference of Climbing and Walking Robots (CLAWAR)*, pages 333–342, Portsmouth, UK.

[Full et al., 1991] Full, R., Blickhan, R., and Ting, L. (1991). Leg design in hexapedal runners. *Journal of Experimental Biology*, 158:369–390.

[Full and Koditschek, 1999] Full, R. and Koditschek, D. (1999). Templates and anchors: Neuromechanical hypotheses of legged locomotion on land. *Journal of Experimental Biology*, 202:3325–3332.

[Full and Tu, 1990] Full, R. and Tu, M. (1990). The mechanics of six-legged runners. *Journal of Experimental Biology*, 148:129–146.

[Gassmann et al., 2005a] Gassmann, B., Zacharias, F., Zöllner, J., and Dillmann, R. (2005a). Localization of walking robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1483–1488, Barcelona, Spain.

[Gassmann et al., 2005b] Gassmann, B., Zöllner, J., and Dillmann, R. (2005b). Navigation of walking robots: Localisation by odometry. In Armada, M. and de Santos, P. G., editors, *Climbing and Walking Robots (Proceedings of the Seventh International Conference CLAWAR 2004)*. Springer, Berlin.

[Goldschmidt et al., 2014] Goldschmidt, D., Wörgötter, F., and Manoonpong, P. (2014). Biologically-inspired adaptive obstacle negotiation behavior of hexapod robots. *Frontiers in Neurorobotics*, 8(3).

[Gorinevsky and Shneider, 1990] Gorinevsky, D. and Shneider, A. (1990). Force control in locomotion of legged vehicles over rigid and soft surfaces. *The International Journal of Robotics Research*, 9(2):4–23.

[Görner, 2007] Görner, M. (2007). Entwurf eines mechatronischen Konzepts für einen Krabbelroboter basierend auf der DLR-Hand Fingertechnologie. Master's thesis, Otto-von-Guericke Universität Magdeburg, Magdeburg, Germany.

[Görner and Hirzinger, 2010] Görner, M. and Hirzinger, G. (2010). Analysis and evaluation of the stability of a biologically inspired, leg loss tolerant gait for six-and eight-legged walking robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4728–4735, Anchorage, AK, USA.

[Görner et al., 2008] Görner, M., Wimböck, T., Baumann, A., Fuchs, M., Bahls, T., Grebenstein, M., Borst, C., Butterfass, J., and Hirzinger, G. (2008). The dlr-crawler: A testbed for actively compliant hexapod walking based on the fingers of dlr-hand ii. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1525 – 1531, Nice, France.

[Graham, 1972] Graham, D. (1972). A behavioral analysis of the temporal organisation of walking movements in the 1st instar and adult stick insect (Carausius morosus). *Journal of Comparative Physiology*, 81:23–52.

[Guddat, 2002] Guddat, M. (2002). *Autonome, addaptive Bewegungskoordination von Gehmaschinen in komplexer Umgebung*. PhD thesis, GerhardMercatorUniversität Duisburg, Duisburg, Germany.

[Haralick et al., 1989] Haralick, R., Joo, H., Lee, C.-N., Zhuang, X., Vaidya, V., and Kim, M. (1989). Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1446.

[Holmes et al., 2006] Holmes, P., Full, R., Koditschek, D., and Guckenheimer, J. (2006). The dynamics of legged locomotion: models, analyses, and challenges. *SIAM Review*, 48(2):207–304.

[Hoover et al., 2010] Hoover, A., Burden, S., Fu, X.-Y., Sastry, S., and Fearing, R. (2010). Bio-inspired design and dynamic maneuverability of a minimally actuated six-legged robot. In *IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 869–876, Tokyo, Japan.

[Ihme, 2002] Ihme, T. (2002). *Steuerung von sechsbeinigen Laufrobotern unter dem Aspekt technischer Anwendungen*. PhD thesis, Otto-von-Guericke Universität Magdeburg, Magdeburg, Germany.

[Ijspeert, 2008] Ijspeert, A. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 8:642–653.

[Johnson et al., 2011] Johnson, A., Hale, M., Haynes, G., and Koditschek, D. (2011). Autonomous legged hill and stairwell ascent. In *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 134–142, Kyoto, Japan.

[Kaliyamoorthy et al., 2005] Kaliyamoorthy, S., Quinn, R., and Zill, S. (2005). Force sensors in hexapod locomotion. *The International Journal of Robotics Research*, 24(7):563–574.

[Kennedy et al., 2006] Kennedy, B., Okon, A., Aghazarian, H., Garrett, M., Huntsberger, T., Magone, L., Robinson, M., and Townsend, J. (2006). The Lemur II-class robots for inspection and maintenance of orbital structures: A system description. In Tokhi, M., Virk, G., and Hossain, M., editors, *Climbing and Walking Robots (Proceedings of the Eighth International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines)*. Springer, Berlin.

[Kerscher et al., 2008] Kerscher, T., Roennau, A., Ziegenmeyer, M., Gassmann, B., Zoeller, J., and Dillmann, R. (2008). Behavior-based control of the six-legged walking machine LAURON IVc. In Marques, L., de Almeida, A., Tokhi, M., and Virk, G., editors, *Advances in Mobile Robotics (Proceedings of the Eleventh International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines)*. World Scientific, Singapore.

[Klein and Briggs, 1980] Klein, C. and Briggs, R. (1980). Use of active compliance in the control of legged vehicles. *IEEE Transactions on Systems, Man and Cybernetics*, 10(7):393–400.

[Klein et al., 1983] Klein, C., Olson, K., and Pugh, D. (1983). Use of force and attitude sensors for locomotion of a legged vehicle over irregular terrain. *The International Journal of Robotics Research*, 2(2):3–17.

[Koos et al., 2013] Koos, S., Cully, A., and Mouret, J.-B. (2013). Fast damage recovery in robotics with the t-resilience algorithm. *The International Journal of Robotics Research*, 32(14):1700–1723.

[Krotkov et al., 1991] Krotkov, E., Bares, J., Kanade, T., Mitchell, T., Simmons, R., and Whittaker, W. (1991). Ambler: A six-legged planetary rover. In *International Conference on Advanced Robotics (ICAR)*, pages 717–722, Pisa, Italy.

[Krotkov et al., 1995] Krotkov, E., Simmons, R., and Whittaker, W. (1995). Ambler: Performance of a six-legged planetary rover. *Acta Astronautica*, 35(1):75–81.

[Lamon and Siegwart, 2004] Lamon, P. and Siegwart, R. (2004). Inertial and 3D-odometry fusion in rough terrain - Towards real 3D navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1716–1721, Sendai, Japan.

[Lewinger, 2005] Lewinger, W. (2005). Insect-inspired actively compliant robotic hexapod. Master's thesis, Case Western Reserve University, Cleveland, OH, USA.

[Lin et al., 2005] Lin, P., Komsuoglu, H., and Koditscheck, D. (2005). A leg configuration measurement system for full-body pose estimates in a hexapod robot. *IEEE Transactions on Robotics*, 21(3):411–422.

[Lin et al., 2006] Lin, P., Komsuoglu, H., and Koditscheck, D. (2006). Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits. *IEEE Transactions on Robotics*, 22(5):932–943.

[Ma et al., 2012] Ma, J., Susca, S., Bajracharya, M., Matthies, L., Malchano, M., and Wooden, D. (2012). Robust multi-sensor, day/night 6-DOF pose estimation for a dynamic legged vehicle in GPS-denied environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 619–626, Saint Paul, MN, USA.

[Manoonpong, 2007] Manoonpong, P. (2007). *Neural preprocessing and control of reactive walking machines*. Springer.

[Neville et al., 2006] Neville, N., Buehler, M., and Sharf, I. (2006). A bipedal running robot with one actuator per leg. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 848–853, Orlando, FL, USA.

[Ott et al., 2011] Ott, C., Roa, M., and Hirzinger, G. (2011). Posture and balance control for biped robots based on contact force optimization. In *IEEE-RAS International Conference on humanoid robots (Humanoids)*, pages 26–33, Bled, Slovenia.

[Paskarbeit et al., 2015] Paskarbeit, J., Schilling, M., Schmitz, J., and Schneider, A. (2015). Obstacle crossing of a real, compliant robot based on local evasion movements and averaging of stance heights using singular value decomposition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3140–3145, Seattle, WA, USA.

[Pfeiffer, 2007] Pfeiffer, F. (2007). The tum walking machines. *Philosophical Transactions of the Royal Society A*, 365(1850):109–131.

[Pfeiffer et al., 1995] Pfeiffer, F., Eltze, J., and Weidemann, H.-J. (1995). The tum-walking machine. *Intelligent Automation and Soft Computing*, 1(3):307–323.

[Pringle, 1938] Pringle, J. (1938). Proprioception in insects: Ii. the action of the campaniform sensilla on the legs. *Journal of Experimental Biology*, 15:114–131.

[Pugh et al., 1990] Pugh, D., Ribble, E., Vohnout, V., Bihari, T., Walliser, T., Patterson, M., and Waldron, K. (1990). Technical description of the adaptive suspension vehicle. *The International Journal of Robotics Research*, 9(2):24–42.

[Reinstein and Hoffmann, 2011] Reinstein, M. and Hoffmann, M. (2011). Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 617–624, Shanghai, China.

[Reinstein and Hoffmann, 2013] Reinstein, M. and Hoffmann, M. (2013). Dead reckoning in a dynamic quadruped robot based on multimodal proprioceptive sensory information. *IEEE Transactions on robotics*, 29(2):563–571.

[Ren et al., 2015] Ren, G., Chen, W., Dasgupta, S., kolodziejski, C., Wörgtter, F., and Manoonpong, P. (2015). Multiple chaotic central pattern generators with learning for legged locomotion and malfunction compensation. *Information Sciences*, 294:666–682.

[Rönnau et al., 2014] Rönnau, A., Heppner, G., Nowicki, M., and Dillmann, R. (2014). LAURON V: A versatile six-legged walking robot with advanced maneuverability. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 82–87, Besancon, France.

[Roston and Krotkov, 1992] Roston, G. and Krotkov, E. (1992). Dead reckoning navigation for walking robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 607–612, Raleigh, NC, USA.

[Saranli, 2000] Saranli, U. (2000). SimSect - hybrid dynamical simulation environment - user's manual. Technical report, University of Michigan at Ann Arbor, MI, USA.

[Saranli et al., 2001] Saranli, U., Buehler, M., and Koditschek, D. (2001). Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631.

[Schilling et al., 2013a] Schilling, M., Hoinville, T., Schmitz, J., and Cruse, H. (2013a). Walknet, a bio-inspired controller for hexapod walking. *Biological Cybernetics*, 107:397–419.

[Schilling et al., 2013b] Schilling, M., Paskarbeit, J., Hüffmeier, T. H. A., Schneider, A., Schmitz, J., and Cruse, H. (2013b). A hexapod walker using a heterarchical architecture for action selection. *Frontiers in Computational Neuroscience*, 7.

[Schmucker et al., 1996] Schmucker, U., Schneider, A., and Ihme, T. (1996). Hexagonal walking vehicle with force sensing capability. In *International Symposium on Measurement and Control in Robotics*, pages 354–359, Brussels, Belgium.

[Schneider et al., 2014] Schneider, A., Paskarbeit, J., Schilling, M., and Schmitz, J. (2014). HECTOR, a bio-inspired and compliant hexapod robot. In Duff, A., Lepora, N., Mura, A., Prescott, T., and Verschure, P., editors, *Biomimetic and Biohybrid Systems (Third International Conference, Living Machines)*. Springer International Publishing.

[Stelzer et al., 2012] Stelzer, A., Hirschmüller, H., and Görner, M. (2012). Stereo vision based navigation of a six-legged walking robot in unknown rough terrain. *The International Journal of Robotics Research*, 31(4):381–402.

[Waldron and McGhee, 1986] Waldron, K. and McGhee, R. (1986). The adaptive suspension vehicle. *IEEE Control Systems Magazine*, 6(6):7–12.

[Welch and Bishop, 2006] Welch, G. and Bishop, G. (2006). An introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill, NC, USA.

[Zhao et al., 2012] Zhao, J., Zhang, H., Liu, Y., Yan, J., Zang, X., and Zhou, Z. (2012). Development of the hexapod robot hitcr-ii for walking on unstructured terrain. In *International Conference on Mechatronics and Automation (ICMA)*, pages 64–69, Chengdu, China.

[Zill et al., 2004] Zill, S., Schmitz, J., and Büschges, A. (2004). Load sensing and control of posture and locomotion. *Arthropod Structure and Development*, 33(3):273–286.