Control of Bio-Inspired Sprawling Posture Quadruped Robots with an Actuated Spine

THÈSE N^{O} 8788 (2018)

PRÉSENTÉE LE 10 AOÛT 2018
À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE BIOROBOTIQUE
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Tomislav HORVAT

acceptée sur proposition du jury:

Dr R. Boulic, président du jury
Prof. A. Ijspeert, directeur de thèse
Prof. M. Cutkosky, rapporteur
Prof. P. Wensing, rapporteur
Prof. D. Floreano, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

James T. Kirk: *Mr. Scott. Have you always multiplied your repair estimates by a factor of four?*
Montgomery Scott: *Certainly, sir. How else can I keep my reputation as a miracle worker?*

To my parents, sister and Olguţa.

Acknowledgements

Before delving into this thesis, I would like to acknowledge and express my gratitude to extraordinary people who had a great influence on my PhD. Without their guidance, support and friendship, this adventure would be far less enjoyable, or even not possible.

First of all, I would like to thank my advisor Prof. Auke Jan Ijspeert for accepting me to be a part of amazing Biorob. I appreciate his support, understanding and the right balance of guidance and given freedom in exploring different ideas. He also gave me a chance to participate in many memorable and unique events like presenting our robots at the Bay Area Science Festival (San Francisco), controlling one of them during his TED talk and to be a part of a documentary series filmed in Uganda.

It is important to thank and acknowledge the Swiss National Center of Competence in Research (NCCR), whose funding made this research possible. Furthermore, NCCR gave me the opportunity to be a part of a larger project with a noble goal of building robots for search and rescue scenarios.

I would also like to thank the members of my thesis committee: Prof. Mark Cutkosky, Prof. Dario Floreano, Prof. Patrick Wensing as well as the president Dr. Ronan Boulic for their time and willingness to participate in my defense.

A special gratitude goes to Kamilo (Melo), with whom I closely collaborated for the last four years. His resourceful thinking, advice, discussions and the robots he provided me with were invaluable for my work. It impressed me how he was always ready to get his hands dirty, whether we had to build setups for hardware experiments or the robots required maintenance. More importantly, after going together through so many adventures (filming in Uganda!), conferences and Chez Xu dinners, he became a very good friend of mine.

A big Dankeschön goes to Robin (Thandiackal) for introducing me to Biorob and to the first robot I worked with. He became a great colleague, collaborator and an awesome friend, always ready to correct my articles.

Further, I would like to thank Prof. John Nyakatura for giving me the unique opportunity to merge robotics with paleontology. The mix of his professionalism and enthusiasm about our work, made him a great person to collaborate with.

A big role in my overall experience was played by the other researchers in the lab. A friendly atmosphere, group lunches, internal jokes and social activities made my time in Biorob truly enjoyable. I would like to specially thank:

Simon (Hauser), for being a great officemate and traveling companion, for numerous dinners, board game and movie nights and for not seeking a revenge after a good prank. Mehmet (Mutlu), for his explorative spirit and being a great source of funny stories. Shravan (Tata Ramalingasetty), for being friendly and giving us sweet insights into the Indian culture. Behzad (Bayat), for a huge help with moving to a new apartment and great car-related advice. Hamed (Razavi), for being an unlimited source of existential questions and freely sharing ideas from his future book. Jonathan (Arreguit), for helping with French and keeping me company during weekends spent in the lab. Salman (Faraji), for great control related discussions. Peter (Eckert), for always being ready to help. Amy (Wu), for trying to clarify our cryptic questions about English. Mos (Ajallooeian), for sharing his great engineering experience and bringing Lego to the lab. Massimo (Vespignani), for always elevating the lab's atmosphere, pushing for SAT, starting "E Lunch?" group and shenanigans with nerf guns. Alexandre (Tuleu), for trying to explain the Linux kernel and how to compile it. Florin (Dzeladini), for all of his unconventional ideas. Laura (Paez), for connecting people. Luca (Colasanto), for making it hard to be serious around him. Jessica (Lanini), for teaching me humility in ping-pong. Jesse (van den Kieboom), for time spent fixing my bugs, even while being busy with his own thesis. Daniel (Renjewski), for his professionalism and great discussions. Tadej (Petrič), for always questioning everything. Nicolas (Van der Noot), for landing jokes when you least expect them. Alessandro (Crespi), for his huge technical support, sharing his knowledge and making everything in the lab work. Sylvie (Fiaux), for her administrative support and advices.

Furthermore, I would like to thank all the interns and students who contributed both professionally and personally to my time in Biorob: Cole (Simpson), Steve (W. Heim), Gary (Lee), Laura (Fleury), Lukas (Hann), Stas (Tsitkov), Nina (Mohebbi), Astrid (Petitjean), and many others.

All of this would not be possible without unconditional support and help of my parents Ruža and Željko and my little sister Marina. Even though it was not easy for them to see me leaving the country, they encouraged me to follow my dream career and always pursued my happiness. Also, thank you for patient listening to all of my technobabble.

Finally, I would like to express my huge gratitude to Olguţa for her love, support, understanding, help and patience for work related discussions through my PhD. Her presence made even the most difficult times more bearable. And most importantly, thank you for making Switzerland truly feel like a home!

Lausanne, 21 May 2018 T. H.

Abstract

Sprawling posture robots are characterized by upper limb segments protruding horizontally from the body, resulting in lower body height and wider support on the ground. Combined with an actuated segmented spine and tail, such morphology resembles that of salamanders or crocodiles. Although bio-inspired salamander-like robots with simple rotational limbs have been created, not much research has been done on kinematically redundant bio-mimetic robots that can closely replicate kinematics of sprawling animal gaits.

Being bio-mimetic could allow a robot to have some of the locomotion skills observed in those animals, expanding its potential applications in challenging scenarios. At the same time, the robot could be used to answer questions about the animal's locomotion.

This thesis is focused on developing locomotion controllers for such robots. Due to their high number of degrees of freedom (DoF), the control is based on solving the limb and spine inverse kinematics to properly coordinate different body parts. It is demonstrated how active use of a spine improves the robot's walking and turning performance. Further performance improvement across a variety of gaits is achieved by using model predictive control (MPC) methods to dictate the motion of the robot's center of mass (CoM).

The locomotion controller is reused on an another robot (OroBOT) with similar morphology, designed to mimic the kinematics of a fossil belonging to *Orobates*, an extinct early tetrapod. Being capable of generating different gaits and quantitatively measuring their characteristics, OroBOT was used to find the most probable way the animal moved. This is useful because understanding locomotion of extinct vertebrates helps to conceptualize major transitions in their evolution.

To tackle field applications, e.g. in disaster response missions, a new generation of field-oriented sprawling posture robots was built. The robustness of their initial crocodile-inspired design was tested in the animal's natural habitat (Uganda, Africa) and subsequently enhanced with additional sensors, cameras and computer. The improvements to the software framework involved a smartphone user interface visualizing the robot's state and camera feed to improve the ease of use for the operator.

Using force sensors, the locomotion controller is expanded with a set of reflex control modules. It is demonstrated how these modules improve the robot's performance on rough and unstructured terrain.

The robot's design and its low profile allow it to traverse low passages. To also tackle narrow passages like pipes, an unconventional crawling gait is explored. While using it, the robot lies on

the ground and pushes against the pipe walls to move the body. To achieve such a task, several new control and estimation modules were developed.
By exploring these problems, this thesis illustrates fruitful interactions that can take place between robotics, biology and paleontology.

Key words: Bio-inspired robots, Locomotion control, Sprawled posture, Robotic spine, Disaster robotics, Software and control architecture

Zusammenfassung

Roboter mit ausladender Haltung zeichnen horizontal vom Körper hervorstehende obere Gliedmassenteile aus, welche zu einer geringeren Körperhöhe und einem breiteren Stand auf dem Boden führen. Kombiniert mit einem angetriebenen segmentierten Rückgrat und Schwanz, ähnelt eine solche Morphologie jenen von Salamandern oder Krokodilen. Bioinspirierte salamanderähnliche Roboter mit einfachen rotierenden Gliedmassen wurden schon in der Vergangenheit konstruiert, jedoch gibt es nicht viele Forschungsstudien, welche sich mit überaktuierten biomimetischen Robotern befassen, die die Kinematik von Tieren mit ausladender Haltung replizieren können. Die biomimetischen Charakteristiken könnten einem Roboter die Fortbewegungsfertigkeiten solcher Tiere ermöglichen und daher die potentiellen Einsatzgebiete in anspruchsvollen Szenarien erweitern. Gleichzeitig könnte ein solcher Roboter benutzt werden, um Fragen betreffend Fortbewegungsmechanismen in Tieren zu beantworten.

Diese Dissertation legt ihren Fokus auf die Entwicklung von Regelung für die Fortbewegung solcher Roboter. Aufgrund ihrer grossen Anzahl Freiheitsgrade, basiert die Regelung auf dem Lösen der inversen Kinematik von Gliedmassen und dem Rückgrat, um die verschiedenen Körperteile richtig zu koordinieren. Es wird dargelegt, wie die aktive Nutzung des Rückgrats das Laufen und das Drehen verbessert. Weitere Leistungssteigerung einer Vielzahl an Gangarten wird durch Model Predictive Control erzielt, womit die Bewegung des Schwerpunkts des Roboters kontrolliert wird.

Der Fortbewegungsregler wird für einen anderen Roboter (OroBOT) mit ähnlicher Morphologie wiederverwendet, welcher entwickelt wurde, um die Kinematik eines Fossils von Orobates, einem ausgestorbenen frühen Tetrapoden, zu imitieren. In der Lage verschiedene Gangarten auszuführen und quantitative Charakteristiken zu messen, wurde OroBOT verwendet, herauszufinden wie sich das Tier mit höchster Wahrscheinlichkeit bewegt hat. Dies ist förderlich, da ein Verständnis der Fortbewegung von ausgestorbenen Wirbeltieren helfen kann, um grundlegende Evolutionsübergänge abzuleiten.

Um Feldeinsätze zu meistern, z.B. Katastropheneinsatzmissionen, wurde eine neue Generation von feldtauglichen Robotern mit ausladender Haltung konzipiert. Die Robustheit ihres ursprünglichen krokodil-inspirierten Designs wurde im natürlichen Habitat (Uganda, Afrika) des Tieres getestet und anschliessend mit zusätzlichen Sensoren, Kameras und einem Computer erweitert. Die Verbesserungen betreffend des Software-Frameworks beinhalteten ein Smartphone User Interface, welches den Zustand des Roboters und einen Kamerafeed visualisiert, um die Bedienung für den Benutzer zu vereinfachen. Mit Hilfe von Kraftsensoren wurde der Fortbewegungsregler

mit einem Set von Reflexkontrollmodulen erweitert. Es wird aufgezeigt wie diese Module die Performance des Roboters auf unwegsamem und unstrukturiertem Gelände verbessern. Das Design des Roboters und sein kleingehaltenes Höhenprofil erlauben es ihm niedrige Passagen zu begehen. Um auch schmale Passagen wie Rohre zu durchqueren, wurde eine unkonventionelle Gangart untersucht. Dabei liegt der Roboter auf dem Boden und drückt sich an die Rohrwände um den Körper zu bewegen. Um diese Aufgabe zu erfüllen, wurden einige neue Regelungs- und Schätzungsmodule entwickelt.
Durch die Untersuchung dieser Probleme zeigt diese Dissertation ein ertragreiches Zusammenwirken von Robotik, Biologie und Paläontologie auf.

Stichwörter: Bioinspirierte Roboter, Lokomotionskontrolle, Ausladende Haltung, robotisches Rückgrat, Katastrophenrobotik, Software- und Regelungsarchitektur

Résumé

Les robots à posture tentaculaire sont caractérisés par des segments de membres supérieurs qui sortent horizontalement du corps, ce qui entraîne une diminution de la hauteur du corps et un soutien plus écarté sur le terrain. Combiné avec une colonne vertébrale segmentée et actionnée ainsi qu'une queue, une telle morphologie ressemble à celle des salamandres ou des crocodiles.

Bien que des robots bio-inspirés de type salamandre avec des membres rotatifs simples aient déjà été créés, peu de recherches ont été faites sur des robots bio-mimétiques sur-actionnés qui peuvent reproduire fidèlement la cinématique des démarches d'animaux de posture tentaculaire.

Le fait d'être bio-mimétique pourrait accorder à un robot des certaines compétences de locomotion observées chez ces animaux, permettant ainsi d'élargir ses applications potentielles dans des scénarios difficiles. Par ailleurs, le robot pourrait aussi être utilisé pour répondre à des questions sur la locomotion de l'animal.

Cette thèse est axée sur le développement de contrôleurs de locomotion pour de tels robots. En raison de leur nombre élevé de degrés de liberté (DOF), le contrôle est basé sur la résolution de la cinématique inverse des membres et de la colonne vertébrale afin de coordonner correctement les différentes parties du corps. Il est démontré que l'utilisation active de la colonne améliore les performances de marche et de rotation du robot. Une amélioration supplémentaire des performances à travers une variété de démarches est obtenue en utilisant des méthodes de commande prédictive (MPC) pour dicter le mouvement du centre de masse du robot (CoM).

Le contrôleur de locomotion est réutilisé sur un autre robot (OroBOT) de morphologie ressemblante, conçu pour imiter la cinématique d'un fossile appartenant à *Orobates*, un tétrapode précoce disparu. Étant capable de générer différentes allures et de mesurer quantitativement leurs caractéristiques, OroBOT a été utilisé pour trouver la manière de déplacement la plus probable de l'animal. Ceci est utile car comprendre la locomotion des vertébrés disparus aide à conceptualiser les transitions majeures dans leur évolution.

Pour aborder les applications de terrain, par exemple dans les missions de réponse aux catastrophes, une nouvelle génération de robots de posture tentaculaire destinés pour le terrain a été conçue. La robustesse de leur conception, initialement inspirée du crocodile, a été testée dans l'habitat naturel de l'animal (Ouganda, Afrique), puis renforcée avec des capteurs, des caméras et un ordinateur supplémentaires. Les améliorations de la structure logicielle impliquaient une interface utilisateur sur smartphone permettant de visualiser l'état du robot et l'alimentation de la caméra afin d'améliorer la facilité d'utilisation pour l'opérateur.

À travers l'utilisation de capteurs de force, le contrôleur de locomotion est accru avec un

ensemble de modules de contrôle de réflexe. Il est démontré comment ces modules améliorent les performances du robot sur un terrain accidenté et non-structuré.
La conception du robot et son profil bas lui permettent de traverser des passages bas. Pour aborder également des passages étroits comme des tuyaux, une démarche rampante non conventionnelle est explorée. En l'adoptant, le robot repose sur le sol et pousse contre les parois du tuyau pour déplacer son corps. Pour réaliser une telle tâche, plusieurs nouveaux modules de contrôle et d'estimation ont été développés.
Par l'exploration de ces problèmes, cette thèse illustre des interactions fructueuses qui peuvent avoir lieu entre la robotique, la biologie et la paléontologie.

Mots-clés : robots bio-inspirés, contrôle de la locomotion, posture tentaculaire, colonne vertébrale robotisée, robotique en cas de catastrophe, architecture de logiciel et de contrôle

Contents

Acknowledgements **v**

Abstract (English/Deutsch/Français) **vii**

List of figures **xvi**

List of tables **xviii**

1 Introduction **1**
- 1.1 Motivation 1
- 1.2 Challenges 3
- 1.3 Approach and thesis outline 3
- 1.4 Contributions 5
- 1.5 Background 5
  - 1.5.1 Sprawled posture in robots 5
  - 1.5.2 Robots with an actuated spine 6
  - 1.5.3 Using robots to study animal locomotion 7
  - 1.5.4 Disaster robots 8

I Control **11**

2 Pleurobot - a Salamander-like Robot **13**
- 2.1 Motivation 14
- 2.2 Robot hardware 15
- 2.3 Robot software 16
  - 2.3.1 Locomotion controller 17
- 2.4 Pleurobot’s potential for search and rescue applications 19

3 Leg and Spine Controllers for a Sprawling Posture Robot **21**
- 3.1 Introduction 22
- 3.2 Our approach 23
  - 3.2.1 Reference frames 24
- 3.3 Leg controller 25

3.3.1 Feet trajectories and coordination 25
3.3.2 Leg inverse kinematics 26
3.4 Leg-spine synchronization 27
3.4.1 Girdle control 27
3.5 Spine inverse kinematics 30
3.5.1 Spline approximation 31
3.5.2 Optimization 31
3.6 Walking and turning 35
3.7 Experiments 35
3.7.1 Simulation 36
3.7.2 Hardware experiments 37
3.8 Discussion and conclusion 39

4 Model Predictive Control Based Framework for CoM Control of a Quadruped Robot 41
4.1 Introduction 42
4.2 Center of mass control framework 42
4.2.1 CoM dynamics 44
4.2.2 MPC problem formulation 45
4.2.3 CoM follower 46
4.3 The platform 47
4.4 Experiments 48
4.4.1 Simulation 50
4.4.2 Hardware experiments 55
4.5 Discussion 57
4.5.1 Future steps 57

II Science 59

5 Reverse Engineering the Locomotion of an Extinct Stem Amniote 61
5.1 Abstract 62
5.2 Introduction 64
5.3 Results 64
5.4 Discussion 69
5.5 Methods 72
5.5.1 Comparative motion analysis of extant species 72
5.5.2 Digital marionette design and kinematic simulation of Orobates 74
5.5.3 Validation of kinematic simulation workflow 75
5.5.4 Robot design 77
5.5.5 Mass distribution 79
5.5.6 Simulated robot 79
5.5.7 Walking frequency 79

5.5.8 Locomotion control of OroBOT . . . 80
5.5.9 Gait exploration and evaluation . . . 81

III Application 83

6 Testing Bio-Robots in African Wilderness Prepares them for Disaster Response Missions 85
6.1 Introduction . . . 86
6.1.1 Challenges . . . 87
6.1.2 Approach . . . 88
6.2 Results . . . 90
6.2.1 Krock - a new class of sprawling posture robots . . . 90
6.2.2 Born in a lab, to live in the field . . . 92
6.2.3 Testing in natural scenarios . . . 94
6.2.4 Addressing the disaster response challenges . . . 96
6.3 Discussion . . . 98
6.4 Materials and methods . . . 100
6.4.1 Bio-informed robot design . . . 100
6.4.2 Rapid fabrication . . . 101
6.4.3 Control and integration . . . 103
6.4.4 Long range communication system . . . 106
6.4.5 Rapid development of the user interface . . . 107

7 Reflex Based Controller for Walking on Uneven Terrain 111
7.1 Introduction . . . 112
7.2 The setup . . . 113
7.3 Reflexes and attitude controller . . . 114
7.3.1 Leg extension reflex . . . 114
7.3.2 Stumbling correction reflex . . . 114
7.3.3 Attitude controller . . . 115
7.4 Experiments . . . 117
7.4.1 Simulation . . . 117
7.4.2 Hardware experiments . . . 118
7.5 Results . . . 118
7.6 Porting the reflex module to Krock-2 robot . . . 123
7.7 Discussion and conclusion . . . 125
7.7.1 Future work . . . 125

8 Crawling in Cluttered Environments and Pipes 127
8.1 Introduction . . . 128
8.2 Problem statement . . . 128
8.3 Pipe crawling algorithm . . . 130

8.3.1 Crawling state machine 131
8.3.2 Contact force distribution 132
8.3.3 Contact force estimation 133
8.3.4 Contact force control 135
8.3.5 Pipe navigation 136
8.4 Experiments 138
8.4.1 Simulation 138
8.4.2 Hardware experiments 145
8.5 Discussion 147
8.5.1 Future work 148

9 Conclusion 151
9.1 Future steps 154
9.2 Final words 155

A Supplementary Information: Orobates and OroBOT 157
A.1 OroBOT leg control 157
A.2 OroBOT spine control 157
A.3 OroBOT foot trajectory 158
A.4 OroBOT spine trajectory 158
A.5 Temporal gait parameters of OroBOT 158
A.6 Main gait characteristics of OroBOT 159
A.7 Constraints to the OroBOT gaits imposed by the trackways 160
A.8 Defining open gait parameters 161
A.9 Trajectory shift and feet parameters 162
A.10 Metrics definition 164
A.10.1 Efficiency 164
A.10.2 Balance 165
A.10.3 Precision 165
A.11 Gait exploration and evaluation 166

B Supplementary Tables: Orobates and OroBOT 169

Bibliography 191

Curriculum Vitae

List of Figures

1.1 Illustration of sprawling and upright posture 2

2.1 Pleurobot with a total of 29 degrees of freedom 15
2.2 High level software scheme of Pleurobot's controller 18

3.1 The inverse kinematics based locomotion of Pleurobot 23
3.2 The right hand reference frames of Pleurobot 24
3.3 Parametrization of Pleurobot's feet trajectories 26
3.4 Angles between the front girdle, feet and girdle frame 28
3.5 A reachable space of the front left foot in the function of leg-spine synchronization 29
3.6 The calculation of the reference for the girdle oscillation 30
3.7 Steps in solving the inverse kinematics of the spine 33
3.8 Effect of the optimization step in the spine motion 34
3.9 The setup of the simulation experiment 36
3.10 The setup of the hardware experiment 38
3.11 An overlay of girdle trajectories while walking in a circle 39

4.1 An overview of the control architecture 43
4.2 The Pleurobot's reference frame 48
4.3 Snapshots of the simulation experiment 51
4.4 The results of the simulation experiments 53
4.5 The results of the simulation experiments 54
4.6 The results of the hardware experiments 56

5.1 Flow chart of the basic steps of Orobates locomotion analysis 62
5.2 Fossil, robot, and trackway detailed description 66
5.3 Extant animal data 68
5.4 Procedure of identifying plausible OroBOT gaits 70
5.5 Kinematic simulation of Orobates 75
5.6 Kinematic simulation workflow validation using caiman 77
5.7 OroBOT's reference frame and kinematic gait parameters 81

6.1 Krock - a new class of sprawling posture robots 88
6.2 Bio-informed design of Krock 91

6.3 Different terrains encountered in Uganda 95
6.4 Krock-2 and its main features 97
6.5 Snapshots of Krock-2 passing through a low passage 98
6.6 An example of producing an aluminum part of Krock robot 103
6.7 System integration scheme of Krock robots 105
6.8 The long range communication system 107
6.9 The user interface application made in Unity 109

7.1 The force sensor setup 113
7.2 Reflex and attitude modules added to the basic locomotion controller 116
7.3 The terrains used in the simulation experiments 117
7.4 Simulation experiments with tailed version of Pleurobot 120
7.5 Simulation experiments with tailless version of Pleurobot 121
7.6 The stumbling reflex experiment with the tailless configuration of Pleurobot . . . 122
7.7 The leg extension reflex experiment with the tailed configuration of Pleurobot . . 122
7.8 Simulation experiments of Krock-2 walking over obstacles 123
7.9 Hardware experiment of Krock-2 walking over obstacles 124

8.1 Comparison of possible locomotion strategies 130
8.2 State machine of crawling gait 131
8.3 Odometry based local curvature estimation 137
8.4 Snapshots of a crawling through a turn 140
8.5 Contact force during the simulation experiment 141
8.6 Overlaid snapshots of crawling through a pipe with multiple turns 142
8.7 Overlaid snapshots of crawling through a pipe with increasing inclination 143
8.8 Overlaid snapshots of crawling through a helix pipe 143
8.9 Overlaid snapshots of crawling through a corridor with the variable width 144
8.10 Evaluation of the force estimation 145
8.11 Crawling experiment with hardware - straight flat segment 146
8.12 Crawling experiment with hardware - turning 147

A.1 Exploration of the optimal foot stiffness and trajectory offset values 163
A.2 Computation of the precision metrics 166

List of Tables

4.1 Gait parameters used in the experiments. 49
4.2 Controller parameters used in the experiments. 50

7.1 Reflex and attitude control module parameters 118

8.1 Crawling parameters 138

B.1 Analyzed experimental subjects 169
B.2 Fossil / robot dimensions 170
B.3 Fossil / robot mass distribution 171
B.4 OroBOT mass breakdown 172
B.5 Gait parameters and calculated Froude Numbers of different sprawling posture species 173

1 Introduction

1.1 Motivation

We can argue that nature has successfully solved the problem of locomotion. Locomotion has been constantly explored and refined for millions of years as an important factor in the fitness function driving evolution. Whether you are a hunter or a prey, locomotion proficiency gives you an advantage, either to outmaneuver the opponent or access areas that are out of reach for others. The results are evident - animals, in all shapes and sizes, managed to occupy a wide range of different environments on Earth. We (humans) have been improving and finding new ways of locomotion (transportation) since the invention of the wheel. However, cars, trains, boats, planes, submarines, all have precisely structured and well known environments they interact with. It was not long ago that we started to build (semi)autonomous machines with less strict boundaries on their locomotion domain. Instead of starting from scratch, scientists and engineers seek inspiration in nature to build and control such machines. We call them bio-inspired robots.

The adjective "bio-inspired" can be applied to a variety of robot types. Flying robots borrow principles from birds or insects that influence body design, wing structure and function as well as new applications [41, 119, 170]. Aquatic robots mimic fish morphology to propel themselves using fins or they utilize the entire body like lampreys and eels. Legged robots are the most common terrestrial bio-inspired robots. Their morphology is often influenced, to some degree, by tetrapods (mammals, reptiles and amphibians). Besides influencing robot's structure and design, nature also provides inspiration for control algorithms which drive the robots. In both cases, the ideas can be taken at different levels of abstraction [57].

One of the basic features robots can borrow from animals is the way the body is supported by the legs, which characterizes the posture. The posture can be sprawled or upright, as illustrated in Fig. 1.1. The sprawling posture, characterized by the stance where upper limb segments are being held horizontally, while the lower segments are vertical to the ground, is considered to be the oldest [28]. The living animals that still use it are the amphibians and most of the reptiles. The sprawling gait often requires the trunk to bend sideways while the animal walks or runs [28]. The trunk bending, together with a flexible tail, gives a possibility of utilizing the body undulations to produce anguilliform swimming in water.

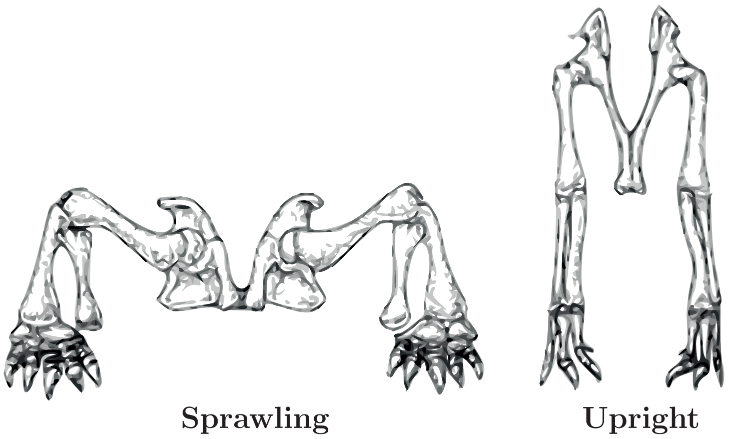


Figure 1.1 – Illustration of sprawling and upright (erect) posture in animals. Picture taken and modified from [42].

Most well known larger legged quadruped robots use an upright posture, similar to cats or dogs, with a rigid body (spine). They have shown great performance on both flat ground and more challenging irregular terrains [156]. On the other hand, legged reptile and amphibian inspired robots are predominantly used to test specific scientific hypotheses about animal locomotion. Such robots with morphology that serves a precisely defined purpose have limited locomotion capabilities in other domains. It is interesting to explore what can be achieved with a sprawling posture robot whose kinematics and morphology closely match its animal counterpart and how far its locomotion skills can be pushed.

Sprawling posture in robots have the potential to provide certain advantages. The body is closer to the ground and the feet form a larger support polygon making falls due to a balancing failure less likely and also less impacting for the robot's structural integrity. Besides, the robot can easily lie on the ground between instances of motion thus conserving energy. An elongated, segmented and actuated spine provides kinematic advantage for improved maneuverability. Combined with an actuated tail, it can be used to swim thus expanding the robot's locomotion capabilities and therefore application to multiple domains.

A specific application of sprawling posture robots, worth considering, is in the field of disaster robotics. Natural (earthquakes, tsunamis, hurricanes etc.) and man-inflicted disasters often cause severe damage of urban infrastructure and have a high social impact [127]. The resulting rubble, flooding, and collapsed buildings form extremely unstructured environments, that represent a big challenge for legged robots. Utilizing the sprawling posture, robots can tackle complex terrain even with simple actuation, sensing and control. Furthermore, the low profile of the sprawled posture makes it possible to enter small cracks and openings. If the robot is waterproofed it can also access and swim in flooded areas.

1.2 Challenges

In order to replicate a sprawled posture and actuated spine in a robot, we have to tackle several mechanical and control challenges. An inherent drawback of sprawled posture, compared to the upright posture, is a higher torque demand on the shoulder joint in order to support the body. Combined with increased body weight due to the actuated spine, the choice of actuators and a balance of size, weight and output torque has to be thoughtfully considered. The sheer number of possible actuated degrees of freedom requires a careful integration of all the components. Even mounting options of different parts are limited due to the segmentation of the body.

The control challenges arise from the high number of DoFs and from how to properly coordinate them. For example, sprawling animals provide us with an insight in spine and limb motion, however it is not trivial to translate such behavior to robots and use the spine to improve their mobility and maneuverability during periodic and non-periodic gaits. The previous salamander-like robot from our lab, Salamandra Robotica [33], with simple 1 DoF rotational limbs, used a fully bio-inspired joint space controller. Such an approach might not be suitable for a robot with more complex limbs as controlling the limb motion in Cartesian space becomes more important.

Use of robots in disaster response missions often requires them to act over rough terrain, unstructured environments and possibly flooded areas. Locomotion in such places is a challenging task due to environmental variety, unpredictability and difficulties of modeling it. Therefore, robots require additional sensors and feedback driven control that adapts and reacts to the environment. While bigger quadruped robots can carry a variety of high performance sensors, the available options, fit for a smaller sprawled posture robot, are limited. As the robot is intended to explore hardly accessible cluttered areas, a direct line of sight between the robot and an operator is not always possible. Thus the controller should be able to autonomously handle the local environment while the operator only provides high level commands. To tackle flooded areas, the robot has to be able to swim. The greatest challenge in doing it is making the robot waterproof. The waterproofing technique must not interfere with robot's terrestrial capabilities by reducing the limb motion or significantly increasing the weight. The tail which is important for anguilliform swimming [33], does not bring clear benefits during ground locomotion while increasing the robot's complexity and weight.

1.3 Approach and thesis outline

This thesis is focused on the development of locomotion controllers for the sprawling posture robots to address some of the aforementioned challenges. As we focus on replicating animal skills in robots, rather than explaining underlying neural mechanisms, the used control techniques are not always bio-inspired. Nevertheless, we show how such an approach can still be used to answer scientific questions about animal locomotion. By augmenting the locomotion controllers with additional control modules, we address the possible applications in disaster response robotics.

More precisely, we aim to answer the following three questions:

1. How to design and test a locomotion controller for a kinematically redundant quadruped robot that fully utilizes its segmented spine and properly coordinates it with the limbs, resulting in increased maneuverability?
2. How to use a bio-informed robotic platform to study locomotion capabilities of an extinct animal?
3. Which design modifications and control add-ons can give these robots unique capabilities in a field of disaster robotics?

These questions are addressed in the three main parts of the thesis that cover control, science and application aspects of our sprawling posture robots.

In **Part I: Control** we start with a quick overview of our salamander-like robot Pleurobot, with a focus on the control architecture (**Chapter 2**). We use it as a main sprawling posture platform to develop an inverse kinematics based locomotion controller that can produce animal-like locomotion (**Chapter 3**). To fully exploit robot's segmented spine, a spine controller is developed that improves the robot's maneuverability. Although losing the balance is not of a big concern for sprawling posture robots, we take a model predictive control approach to command the center of mass position in order to improve Pleurobot's attitude across multiple gaits (**Chapter 4**).

In **Part II: Science** we use the developed controller to study the locomotion of early tetrapods. For that purpose, we present a new sprawling posture robot (OroBOT) that is designed to mimic the kinematics of an extinct animal (*Orobates pabsti*) based on the fossilized remains. By using a kinematic simulation of *Orobates* and dynamic simulation of OroBOT we explore a variety of different gaits and find most probable candidates for the way the animal moved (**Chapter 5**).

In **Part III: Application** we focus on expanding and specializing capabilities of sprawling robots for disaster response applications. A new class of robots (Krock-1 and Krock-2) is introduced that is field oriented and equipped with additional sensors (**Chapter 6**). By using the contact feedback provided by force sensors, we implemented a set of reflexes to improve the robot's performance on a rough terrain (**Chapter 7**). Finally we designed a specialized pipe crawling controller for Krock-2 aimed to tackle tight openings in rubble and unstructured environments (**Chapter 8**).

1.4 Contributions

Control:

- An inverse kinematics based leg and spine controller for sprawled posture robots.
- Active control of center of mass position, based on model predictive control framework, for a small quadruped robot.
- A pipe crawling controller that makes use of an actuated spine.

Science:

- A methodology for reconstruction of locomotion capabilities in early tetrapods by using a bio-informed robotic platform.

Application:

- A software and control framework for a novel sprawling posture robot oriented towards disaster response applications.

1.5 Background

The robots we work with in this thesis are of unique morphology and not many similar robots are reported in the literature. At the same time the area of bio-inspired robotics, where our work fits, is very broad. Therefore, in this section we try to give a context to the three parts of this thesis: utilizing sprawled posture and actuated spine, studying animal locomotion with robots and disaster robotics.

1.5.1 Sprawled posture in robots

Although sprawled posture is mostly associated with amphibians and reptiles, numerous robots using it are hexapods inspired by arthropods (e.g. insects). Such small scale robots like Sprawlita [27], iSprawl [104], DASH [12], VelociRoACH [70], HAMR3 [8] and variations of them are capable of a highly dynamic locomotion, achieving velocities of multiple body lengths per second. At the same time they are severely underactuated, leveraging on passive structures, compliant leg design [27] and wide base of support on the ground, due to their sprawled posture, to maintain static and dynamic stability. A well known larger hexapod is RHex [172] with individually controllable rotational legs (whegs). Different variations like Rugged RHex (built by Boston Dynamics, Inc.) or X-RHex [59] packed on-board sensors, robust design and even multimodal locomotion (platform AQUA [153] based on Rugged RHex). As it proved to be a versatile, field-ready platform with remarkable rough terrain locomotion performance (including sand, mud, rocks, vegetation etc.), RHex sets a benchmark for many other smaller legged platforms aiming towards field application. At the same time, these robots rely on a simple, mostly open loop joint space control framework that proved efficient due to a low number of actuated DoFs.

Domain specific climbing robots often utilize sprawled posture to keep their bodies close to the wall surface and reduce turn-over torque [67]. Notable examples are hexapod RiSE [179], quadruped RiSE V3 [74] and gecko-inspired Stickybot [105, 73], relying on feet covered with small hairs, called microwedges [6], acting like a directional dry adhesive to stick on vertical surfaces.

Examples of sprawled robots with more articulated limbs achieving a larger workspace, similar to our robots, are TITAN-XIII [107, 108] and Snake Monster [95]. With three and four DoF limbs respectively, both robots rely on solving limb inverse kinematics to follow commanded trajectories. A control framework enables users to easily provide high level commands, like walking direction and steering, via a gamepad. Similar spider-like limb design and sprawled posture is present in some commercially available robotic kits (e.g. PhantomX AX-12 from Trossen Robotics, Inc.) and many quadrupeds designed by hobbyists. Such interest into sprawled posture robots can be attributed to their stability (difficult to tumble over), without a need of advanced control algorithms or sensors. The benefits of sprawling locomotion were also recognized by a Darpa Robotics Challenge participant RoboSimian [99]. Leveraging its seven DoF limbs, RoboSimian can switch between different locomotion modes, including a low sprawled walk that is useful for rough terrain locomotion [75].

1.5.2 Robots with an actuated spine

The majority of robots come in a form of a rigid trunk with appendages attached to it. The trunk's primary purpose is to provide a base structure for attaching actuators, sensors, batteries, computational unit and other components. However, there are robots which completely rely on their body to achieve locomotion. A great example are the snake-inspired robots which come in a form of a kinematic chain — a series of rigid bodies linked together with rotational joints. Planar snake robots, with all joint axes of rotation being parallel, allowing only for bending in coronal (horizontal) plane, locomote by utilizing anisotropic friction with the ground (e.g. by using passive wheels) [79, 80] or push against obstacles [114]. If the axes of the neighboring joints are perpendicular to each other [198, 126, 77], a snake robot becomes capable of 3D bending which brings possibilities of more versatile gaits, like rolling or sidewinding [72, 123]. A snake-like body is also present in amphibious robots that utilize undulatory swimming [102, 31, 77].

A use of an active spine in legged robots is less common and mostly allows bending in sagittal (vertical) plane. In order to explore energetic benefits of spine motion, MIT cheetah [177] had a flexible spine, whose bending was coupled to the motion of hind legs. However, such design was abandoned with its successor MIT Cheetah 2 [146] in favor of a rigid trunk. In smaller quadrupedal robots, the use of active spine proved to be beneficial for the robot's speed and stability of a bounding gait [103]. In [45] authors argue that a flexible but passive bio-inspired spine can result in more natural bounding gait compared to a single joint active spine. When configured to bend in the coronal plane, a spine can be used to improve a robot's turning capabilities [193]. Salamandra robotica, a robot that combines a snake like body and rotational sprawled posture legs, can utilize its spine for swimming, turning and to increase speed of terrestrial locomotion [33].

1.5.3 Using robots to study animal locomotion

While we often seek inspiration from biology in our continuous attempts to build and program robots with better locomotion performance, the robots can also be used to provide us with deeper insights in animal locomotion. What makes them a great tool for studying animals are these characteristics: (i) repeatability of experiments, (ii) possibility to measure and/or estimate values which would be difficult to access in animals (e.g. joint torque), (iii) morphology can be modified, (iv) full command over its behavior (through the robot's controller design), (v) possibility to execute movements that are unnatural or even dangerous for an animal [89]. Leveraging on them, we can perform experiments and studies that would be difficult or not possible with animals.

For a while researchers have been relying on numerical simulations to model animal locomotion [83]. Using robots we can complement or substitute numerical simulations in situations when modeling physical interactions with an environment (water, mud, sand, gravel) is too challenging. For example in [109] a robotic model of a pectoral fin found in fishes was used to study its role in aquatic propulsion. Similarly, the role of an undulatory fin found in knifefish was explored in [34] by using a biomimetic robot. In [112] a hexapod robot helped to develop and test models of interaction with granular media and how it depends on leg's shape.

When dealing with solid ground, numerical simulations (e.g. dedicated robot simulators) are mature enough to accurately simulate rigid body dynamics and interactions with ground. Thus they can be used to complement both animal and robot experiments. For example, in [157] a combination of animal, robot and simulated experiments was used to test a hypothesis about fruit fly locomotion strategies. Easy modifications of a simulated environment allowed rapid exploration of possible walking gaits for different contact parameters (leg adhesion) and ground orientations (vertical and horizontal). The robot is subsequently used to test a subset of findings thus confirming the simulation.

Alongside investigating the role of specific morphology in animals, robotic platforms are ideal to test different bio-inspired control techniques. Although we are still not able to fully replicate animals' neural circuits, robots can pack enough computational resources to emulate simplified neural models. In [91] a salamander like robot, driven by a spinal cord model, replicated swimming, walking and transition between the two locomotion modes. The spinal cord model represented by a central pattern generators (CPGs) — a neural network made of coupled oscillators

— showed how a specific oscillator coupling combined with a high level modulation signal can replicate basic locomotion skills observed in salamanders. A different approach presented in [144], showed how a CPG, based on decoupled oscillators, with an addition of a simple contact force feedback rule can produce walking patterns seen in real animals. The control approach was demonstrated on a quadruped robot that was able to mimic walking patterns of monkeys, horses and camels by changing a weight distribution of the robot's body.

Although proven useful, involvement of robots in animal locomotion research is still held back by technical challenges [89]. The actuation systems are still not able to fully emulate complexity and viscoelastic properties of animal musculoskeletal system. The same applies to artificial sensory systems, specially touch sensors which would emulate skin. Even though some progress is made in manufacturing soft touch sensors [147], covering an entire robotic system with them is difficult. Furthermore, the required amount of time and skill to develop, maintain and modify robots can be high, making a use of numerical simulations a more viable option.

1.5.4 Disaster robots

Also known as search and rescue or just rescue robots, disaster robots are built to sense and act in a disaster environment. Used as a tool in rescuers' arsenal, disaster robots have as a goal inspecting areas that are hardly accessible for humans or dogs, are unsafe, or do not support life [127]. Furthermore, as stated in [127], their design is guided by the following constraints: (i) functioning in extreme environments and operating conditions, (ii) ability to perform in GPS- and wireless-denied environments, (iii) able to provide appropriate human-robot interfaces for both operators and possible victims.

One of the oldest documented uses of disaster robots are the operations following the Chernobyl nuclear accident in 1986. Remotely operated robot-based systems were used in decontamination and reconstruction operations in the zones of a high level radiation [152]. The first documented rescue mission involving robots took place after the attack on World Trade Center (WTC) in 2001 [22]. Multiple robots equipped with a variety of cameras (including thermal) were deployed to inspect small openings in rubble. After the Fukushima Daiichi nuclear disaster in 2011, robots were used to assess the structural damage and flooding levels in the reactor buildings [130]. The robots in these scenarios were based on tracked drive due to its great performance on unstructured terrain of medium complexity. The ones used in Fukushima had additional sub-tracks, with adjustable angle with respect to the body, to improve locomotion over stairs [131].

As new technologies are emerging, many researchers explore their use in search and rescue operations. The robots being considered or even used in such applications are becoming more diverse. Aerial drones are aimed to map and give an overview of disaster areas by providing detailed 3D reconstruction of an environment that can be used by ground robots and rescuers [173, 47]. By utilizing deep learning techniques, they can rapidly classify a terrain to identify traversable areas [39]. Snake robots, due to the slim body, can enter and inspect areas that were previously inaccessible [129]. Their slow terrestrial locomotion can be overcome by using dogs for rapid robot delivery [50].

For efficient locomotion in environments designed for humans (e.g. stairways), legs might be necessary. A highly mobile and dynamic quadruped ANYmal [87] is being designed with aim to operate in harsh environments [88]. Packing a variety of sensors and advanced locomotion and navigation algorithms, it proved to be able of autonomous inspection of multistory oil and gas sites. A similar or even higher locomotion performance is observed in quadrupeds developed by Boston Dynamics, Inc (BigDog, Spot, SpotMini).

A lot of emphasis in disaster robotics is put on human-robot interaction [128], as the robots only serve to complement rescuers, not replace them [127]. To allow for rapid deployment, robots should be easily packable, transportable and easy to operate. As seen in the examples of WTC and Fukushima disasters, this is not always the case since some of the deployed robots required multiple trained operators [22, 130]. Thus, some researchers have been studying ways to reduce the number of required operators per robot, as well as their required skill level to control the robot. For example, in [165] a soft upper body exoskeleton is developed for immersive and natural drone control, which could subsequently be adapted to other types of robots. A step towards flipping the ratio of operators and robots is presented in [68], where a wearable multi-modal interface allows user to issue voice and gesture commands to multiple robots. Ultimately, a single operator could control and coordinate a swarm of different robots acting in a disaster area.

Control Part I

2 Pleurobot - a Salamander-like Robot

This chapter is the introduction to the class of sprawling posture robots with segmented spine on which this entire thesis is based. We introduce the robot - Pleurobot which was the basis for most of the work presented in this thesis. The robot's design methodology and details about mechanical construction are already featured in another work from our laboratory [185], therefore they are only briefly described. More focus is put on the software and control framework, which I fully designed and which are my main contributions.

Parts of the material presented in this chapter are adapted from:

[98] Karakasiliotis, K., Thandiackal, R., Melo, K., Horvat, T., Mahabadi, N., Tsitkov, S., Cabelguen, J., and Ijspeert, A. From cineradiography to biorobots: an approach for designing robots to emulate and study animal locomotion. *Journal of The Royal Society Interface 13*, 119 (2016), 20151089.

My original contribution: Programmed the robot and helped with robot experiments.

2.1 Motivation

The research objective of our lab is using robots and numerical simulations to study animal and human locomotion. One of the first animals we studied was a lamprey - a fish that uses body undulations to swim. Scientists are compelled to decode and create mathematical models that describe its simple nervous system, specially the neural mechanisms underlying its locomotion. To test our mathematical models, a lamprey-like robot AmphiBot [29] was used. Being capable of generating undulations with its elongated body, AmphiBot was used to test spinal cord models that can reproduce swimming and even crawling locomotion [30, 31, 90].

After the lamprey, the next step from an evolutionary perspective was to study salamanders that resemble the first terrestrial tetrapods [11]. Like lampreys, salamanders can also use undulatory swimming for aquatic locomotion, however transitioning to land resulted in a more complex nervous system. To test hypotheses about how salamander locomotion is generated, Salamandra Robotica II was built [32, 33]. Although successful in explaining basic principles behind salamander swimming and walking as well the switching mechanism between locomotion modes [91], its terrestrial locomotion capabilities were limited. With only simple 1 DoF rotational limbs, its locomotion is restricted to a plane.

To study rich motor skills in salamanders, we made a shift from a bio-inspired (Amphibot, Salamandra Robotica II) towards a bio-mimetic approach in designing robots. The result is Pleurobot: a salamander-like robot that closely replicates its biological counterpart *Pleurodeles waltl*. The original concept and mechanical development of the robot was done by Kostas Karakasiliotis and it is described in detail in another work from our lab [96], that lists the following objectives guiding the design of Pleurobot:

1. Similar morphology between animal and robot.
2. Ability to replay three dimensional kinematics as recorded from real salamanders. In that, the robot should have the appropriate DoF and range of motion in order to reproduce the animal's movements.
3. Reasonable size based on the available actuation technology.
4. Easy mechanical construction.
5. Fast and easy repair.
6. Mechanical durability.
7. Low cost.

The first two points were necessary for robot's primary use as a tool for science. The remaining points made Pleurobot interesting to consider as a platform for search and rescue applications, which are discussed in section 2.4.

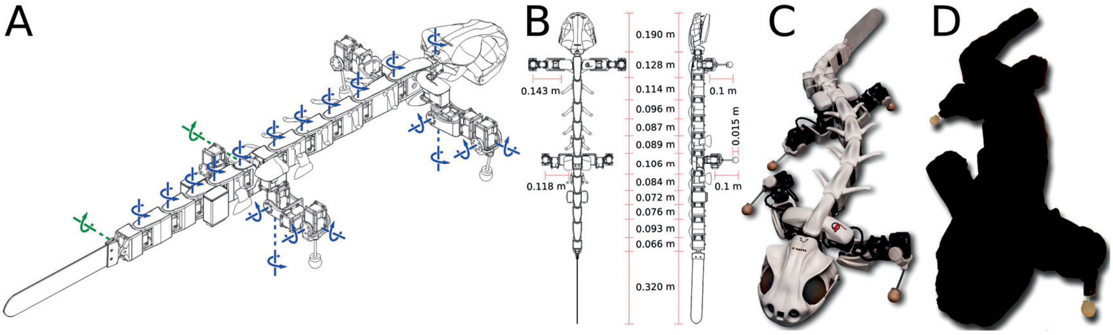


Figure 2.1 – Pleurobot with a total of 29 DoFs (degrees of freedom). (**A**) Blue axes indicate the 27 actuated DoFs, whereas the green axes indicate 2 passive DoF. (**B**) The link lengths of the segmented spine and limbs. The joint distribution is optimized [98] to allow replicating various animal gaits, such as walking, swimming and aquatic stepping. (**C**) The robot design comprises 3D printed parts (white), 27 off-the-shelf servomotors (black), silicon feet and a flexible tail. (**D**) A waterproof swimming suit out of a Lycra® Nylon fabric laminated with a 1mm layer of polyurethane completes the design that allows locomotion in terrestrial as well as aquatic environments.

2.2 Robot hardware

Pleurobot has 27 actuated DoFs, 4 per each limb and 11 along the spine. Most of the rich behaviors of the salamander depend on its segmented limbs (turning, paddling and more), so particular focus was given in replicating the DoFs of the limbs, as opposed to Salamandra Robotica II. The number of spinal DoFs was determined as a trade-off between complexity and accuracy of reproducing lateral spine curvatures observed in the animal. The joint distribution (Fig. 2.1), under given size constraints, was found through optimization that maximized the aforementioned accuracy of reproducing animal motion.

The servo motors we chose to actuate the robot are Dynamixel MX64R from ROBOTIS, Inc. as they offer a good trade-off featuring a fairly high torque-mass ratio (7.3 Nm of stall torque at 126 g), max no-load speed of 78 min^{-1} , and positional accuracy (0.088 ° resolution) at a reasonable price. To actuate the robot's neck we chose lighter and less powerful Dynamixel MX28R. All the servo motors are connected to a single RS-485 bus which is used for communication with the robot's computer located inside its head. The first computer, based on a single-core Intel Atom @ 1.6 GHz CPU, was a part of DARWIN Electronics Kit DW-EK01 from ROBOTIS which made it easy to interface with Dynamixel servo motors. Subsequently it got replaced by a more powerful Odroid XU4 from Hardkernel Co., Ltd., based on an octa-core Samsung Exynos 5422 CPU.

The entire mechanical structure of Pleurobot is created using three-dimensional (3D) printing (Laser sintering) of polyamide 12 (plastic). The ball-shaped feet are made of silicone rubber molded around a spherical structure attached to steel rods that act as the lower leg of each limb. In Chapter 7, the steel rods will be replaced with a specially made enclosures, housing force sensors.

Since the electronics and servo motors are not waterproofed, a specially tailored waterproof suit has been developed allowing the robot to swim. The suit is made of a Lycra® Nylon fabric laminated with a 1 mm layer of polyurethane that ensured waterproofness. The robot can be powered either by a tether or a pack of lithium polymer batteries (LiPo).

2.3 Robot software

Robots are often complex machines, that can vary immensely in their size, function, morphology and cost. Due to such variety there are no clearly defined rules or standards about development of robot software. This is specially true for research oriented robots that never exit their prototype stage. Software robustness and reliability are of lower priority than rapid development, adding novel features and experimentation. Development of Pleurobot's software was no exception. Our decisions were driven by providing solutions for current problems [186], while keeping in mind future expandability and reusability.

Pleurobot is not designed for a highly dynamic locomotion, which would require a fast communication between the software and hardware with known deterministic delays provided by a real time operating system (RTOS). Instead we chose to use a Linux distribution Ubuntu as an operating system (OS) for the robot's computer. The locomotion controller is developed in the C++ programming language. It interfaces directly with the hardware, without a specialized middleware like ROS (Robotic Operating System). We opted for C++ because of its flexibility, popularity and performance. Choosing a widely spread OS like Ubuntu in combination with C++ programming language brings several benefits regarding the software development:

- Possibility to build C++ projects directly on the robot without a need to setup a cross-compiler.
- Easy remote connection to the robot over Secure Shell (SSH) and robot's filesystem mounting (e.g. over SSHFS). Together with the previous point, this makes modifying the locomotion controller fast and easy. Due to the robot's wireless connectivity (WiFi embedded into the main computer board), it can be accessed from any computer connected to the same WiFi network.
- A huge variety of useful, freely available software libraries for linear algebra, numerical optimization, communication, machine learning etc. that simplify development of the locomotion controller.
- Already existing drivers and libraries for servo motors and sensors which simplifies integration of hardware and software.
- Similar OS and environment between robot's computer and personal computers (PC) used for development speed up code deployment. It is possible to develop and test locomotion controller on a more powerful PC running robot dynamics simulator and, once ready for deployment, transfer it to the robot with minimal modifications.

For low level communication with servo motors we use a library provided by ROBOTIS Inc., made for DARWIN Electronics Kit. The library contains functions for reading and writing commands to Dynamixel servo motors as well as accessing additional electronics provided by the Darwin Kit, like a camera, speakers or LED lights which we used as status indicators. The communication with servo motors is done in a broadcast manner. It means that the commands are being sent to all of them at once instead sequentially, resulting in increased speed.

2.3.1 Locomotion controller

As mention earlier, while developing the robot's controller we kept in mind future expansions of features and portability to robots with similar hardware components. The resulting software realization of the locomotion controller is based on two C++ classes - `Controller` class and `Robot` class which are meant to separate control logic from hardware specific functions, as illustrated in Fig. 2.2. For example the `Controller` class contains control related methods like forward and inverse kinematics calculation, feet trajectory generation, locomotion state machine etc. Most relevant variables describing the robot's state are declared as members of the class, making them available to any method which might use them.

The `Robot` class handles communication with the hardware like sending position reference to servo motors or adjusting their settings, reading position from encoders for which it uses functions from the Darwin library. Communication with any additional sensors the robot might use is also meant to be handled by this class.

To issue commands to the robot in real time, a Sony PlayStation 3 gamepad is used. The gamepad communicates with the robot over a Bluetooth connection and allows user to send high level commands, like steering direction or walking speed. A C++ program reading the gamepad runs independently of the main locomotion controller. The program starts on system boot and waits for the pairing signal from the gamepad. Once the gamepad is connected, it can be used to start and stop the locomotion controller program, thus allowing the robot to be operated without a need of another computer (e.g. laptop) and WiFi network. Data between the gamepad program and the locomotion controller is transfered by using a shared memory object.

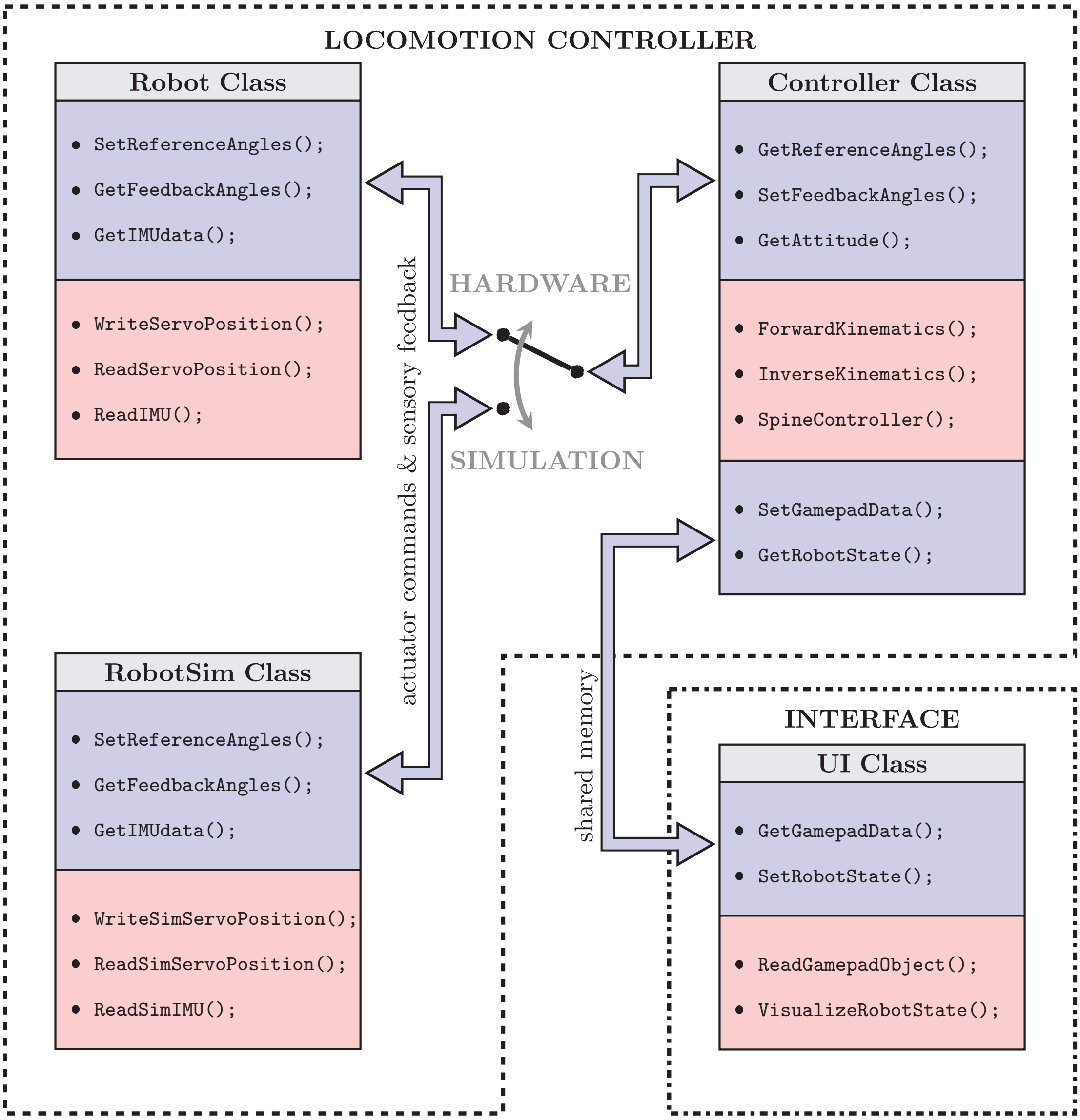


Figure 2.2 – High level software scheme of Pleurobot's controller. Locomotion controller program is built around two C++ classes - `Controller` class and `Robot(Sim)` class. The `Controller` class encapsulates a set of methods (public in blue, private in red) that provide control signals to the robot (e.g. joint angles) in order to achieve desired functionality, while simultaneously taking into account robot's feedback (e.g. IMU data) and user input. Robot class provides a high level interface for the hardware. It acts as a middleware between the hardware and Controller class. Creating the `RobotSim` class, with the same interface (public members) as the `Robot` class, it is possible to use the same `Controller` class for both simulation and the real robot. The user controls and receives feedback from the robot through `UI` class, which is part of a separate Interface program. The communication between the two programs is implemented by using a shared memory object. The figure lists only a small subset of methods for illustrative purpose.

2.4 Pleurobot's potential for search and rescue applications

In the introduction we discussed what the broad and vague term "search and rescue" (S&R) means when combined with robotics. Our state of the art hardware and software are still not mature enough to pursue a goal of making a robot that can replace or work side by side with human rescuers in any conditions, not to mention that the rescuers themselves are still apprehensive about such technology [127]. Nevertheless we aim to close such gaps by providing a platform, specialized for well defined tasks and scenarios, that can be a potential asset in the rescuers' arsenal. As the niche suitable for our bio-inspired or bio-mimetic robots like Pleurobot, we chose following scenarios:

1. Inspection of small openings, narrow passages or tunnels within a rubble.
2. Exploration of shallow water.

The first scenario requires a robot with low and narrow profile that can crawl through openings that are too tight for human rescuers, larger tracked, wheeled or quadruped robots. Furthermore, sprawling posture keeps the robot's body and its center of mass close to the ground reducing risk of falling over due to a loss of balance while walking over a challenging terrain. The challenging, or more specifically rough terrain is automatically assumed due to the amount of rubble expected after a disaster. Pleurobot's mechanical structure satisfies these conditions. The second scenario is constrained to shallow water due to Pleurobot's aquatic capabilities being limited to surface swimming. In both scenarios, we will focus on solving the problem of locomotion, while the perception needed for "search" part of S&R will be briefly discussed in Chapter 6.

Use of Pleurobot for this specific niche comes from its unique morphology suitable for multimodal locomotion, making it capable of seamless transition between the two scenarios. In parallel, Pleurobot has several attributes that are desirable for S&R robots. They are already listed as objectives (points 3 to 7) that guided Pleurobot's design in section 2.1. Reasonable size based on the available actuation technology accounts for robot's low and narrow profile while still utilizing relatively powerful servo motors. Moreover, the robot can be carried and handled by a single person. Fast and easy repair is crucial for field maintenance. For example, any broken servo motor can be replaced under 15 minutes with nothing other than a screwdriver[1]. Low cost, easy mechanical construction and off the shelf components make it possible to quickly (re)build the entire robot from scratch. This ultimately leads to multiple robots active in the field at once, being treated as expendable goods rather than a unique and fragile piece of equipment.

---

[1] A laptop is required to adjust internal servo motor parameters like ID and communication speed.

3 Leg and Spine Controllers for a Sprawling Posture Robot

The previous chapter gave a high level description of Pleurobot's software and control framework, whose main part is the robot's locomotion controller. Here, we introduce the basic components of the locomotion controller: inverse kinematics based leg and spine controllers, and a coordination mechanism between them. They make the robot easy to operate in real time and achieve great maneuverability on flat ground, which is demonstrated in simulation and on the real robot. As they provide flexibility in designing robot's behavior (e.g. gait) and are easy to expand with additional control modules, these controllers will be partially or fully reused in the upcoming chapters.

The material presented in this chapter is adapted from:

[84] Horvat, T., Karakasiliotis, K., Melo, K., Fleury, L., Thandiackal, R., and Ijspeert, A. J. Inverse kinematics and reflex based controller for body-limb coordination of a salamander-like robot walking on uneven terrain. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (2015), IEEE, pp. 195–201.

[86] Horvat, T., Melo, K., and Ijspeert, A. J. Spine controller for a sprawling posture robot. *IEEE Robotics and Automation Letters 2*, 2 (2017), 1195–1202.

My contributions: Fully designed the controller and programmed the robot, designed and carried out the experiments, writing.

3.1 Introduction

Controlling a robot with a segmented spine is challenging due to an increased number of degrees of freedom. The spine often comes in a form of a kinematically redundant kinematic chain, which directly creates contacts with an environment (snake robots) [81, 90, 187, 65], or connects two (or more) girdles (attachment points for the legs) [91]. Its role and control depend on the spine configuration and the robot type.

In the control of snake robots which allow 3D motion, the usual approach is to directly drive joints with carefully designed parametrized wave functions [184] or to define desired body shape in Cartesian space and fit the robot kinematics to it [199, 72]. Navigation-oriented control of planar (2D) snake robots often uses a predecessor-follower control scheme [115], where only the orientation of the first spine segment is directly controlled, while the others follow it with a certain time delay.

Segmented spines are still fairly underutilized in legged robots. Most of the research focuses on improving robot performance by a low degree of freedom spine (active or passive) with a bending in the sagittal (vertical) plane [55, 103]. In [204, 205] it is demonstrated how an actuated spine can be sufficient to produce locomotion in a quadruped robot with passive legs.

In this chapter we will focus on a salamander robot - a quadruped with a segmented spine that allows bending in the transverse (horizontal) plane [98]. In this case, the role of the spine is to allow for multimodal locomotion. It improves robot's terrestrial locomotion [97] and it allows it to swim by using an anguilliform swimming gait. Salamander robots in [90, 91] for both swimming and walking are controlled in joint space by coupled phase oscillators. The behavior of the spine and synchronization between spine and legs are solely determined by coupling weights and intrinsic frequencies of phase oscillators. The turning was achieved by simply bending the spine (adding an offset to the spine angles) without modifying the leg trajectories. Although interesting for modeling a central nervous system (spinal cord) of lamprey and salamander animals and answering scientific questions about their locomotion, such an approach does not take into account the posture of a robot in Cartesian space. The latter is necessary for the precise control of (i) the overall robot posture, (ii) the coordination between girdle movements, and (iii) more precise leg trajectories in relation to the robot's body. As we will demonstrate in this chapter, improving the control of the spine allows for more accurate steering behavior (of the robot).

Subsequently we justify the importance of proper leg-spine coordination and present a controller for achieving it. Afterwards we show how to use the spine for efficient turning with the robot. In the end we show the performance of the controller in both simulation and the real robot.

3.2 Our approach

The robotic platform on which we study the use of spine for locomotion is Pleurobot (Fig. 3.2). Pleurobot is a sprawling posture quadruped robot with segmented spine. Its spine (without the tail) has five actuated degrees of freedom, allowing bending in the horizontal plane. Each leg has four degrees of freedom with accessible motion shown in Fig. 3.5. A detailed description of the robot design methodology and its use as a scientific tool can be found in [98]. The robot is actuated by servo motors that are capable of following joint position references. Therefore, to control the robot's locomotion, we take an inverse kinematics based approach that maps the desired behavior in Cartesian space to a high dimensional joint space of the robot.

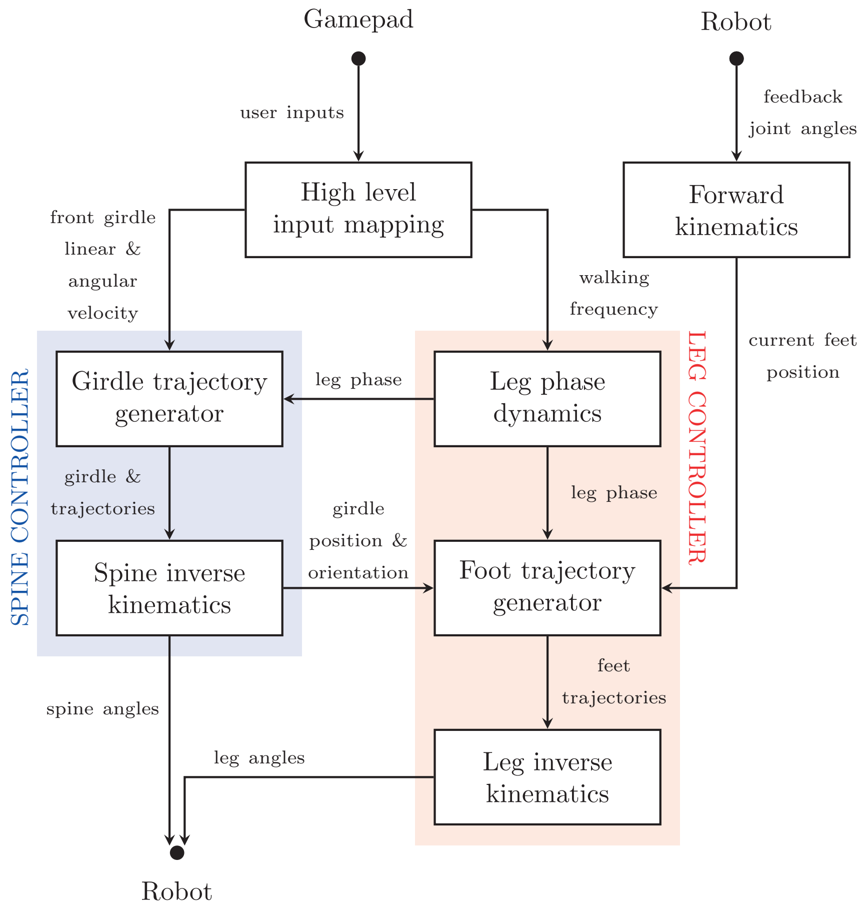


Figure 3.1 – The inverse kinematics based locomotion controller of Pleurobot. The arrows indicate only the most important information flow between the modules. The main controller inputs are user commands issued by a gamepad and a position feedback given by the joint encoders. The outputs are joint position references for the servo motors.

A schematic overview of the locomotion controller that we describe in this chapter is shown in Fig. 3.1. Note the separation between the leg (red) and spine (blue) controllers. Although separated, the two controllers are not independent as they share certain variables assuring a proper spatial and temporal coordination of the body. The leg controller will be described in section 3.3, while the the spine controller, including the mechanism for leg-spine coordination, is described in sections 3.4 and 3.5.

3.2.1 Reference frames

Having the segmented spine makes it difficult to attach a local coordinate frame to the robot. This is a common problem in locomotion of full body undulatory robots [166].

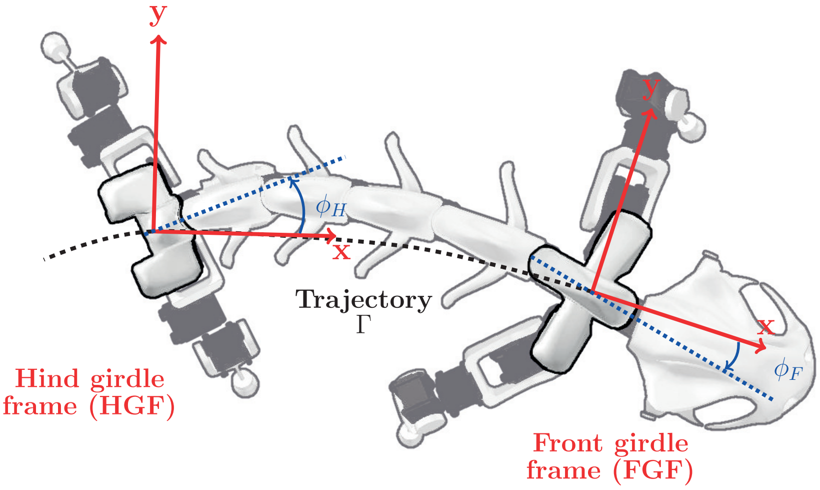


Figure 3.2 – The right hand reference frames of Pleurobot (the z-axis points towards the reader). The front girdle frame is positioned in center of the line connecting shoulders and its orientation is determined by the instantaneous direction of locomotion. The trajectory Γ is an idealized path the front girdle follows in the world frame. The hind girdle frame is positioned in center of the line connecting hips and its orientation is determined by the tangent to the Γ.

Since the girdles are able to move relative to each other as locomotion progresses [98], it is more intuitive to assign separate coordinate frames to each of them. We will refer to them as front girdle frame (FGF) and hind girdle frame (HGF). Their origins correspond to girdle locations (center point between shoulder/hips) while the x-axis of those frames are aligned with girdle's instantaneous direction of locomotion, which is shown in Fig. 3.2. It is worth to notice that the orientations of these frames are not fixed to the corresponding girdle body segment, only their origins are. The instantaneous direction of locomotion of the front girdle is defined by linear and angular velocity which are high-level control signals (e.g. provided by user). The hind girdle follows the trajectory Γ taken by the front girdle. Such design choice minimizes the robot's footprint while turning, enabling it to move through narrow corridors (see Section 3.7). The trajectory Γ does not have to correspond to the actual trajectory followed by the whole robot. It is just a way of relating front/hind girdle frames to each other. More about girdle trajectories is explained in Section 3.6.

3.3 Leg controller

We start the leg controller design by assuming the reference frame locations are known. The leg controller can be divided into several parts: (i) determining the coordination between legs, (ii) generating feet trajectories in Cartesian space and (iii) solving the inverse kinematics.

3.3.1 Feet trajectories and coordination

The foot trajectory of i-th leg is defined as a continuous curve in the frame of the girdle it is attached to:

$$\boldsymbol{\xi}_i(\theta_i) = \begin{bmatrix} x_i(\theta_i) & y_i(\theta_i) & z_i(\theta_i) \end{bmatrix}^\top, \tag{3.1}$$

where $0 \le \theta_i < 1$ is the phase of i-th leg. It consists of two phases, determined by a duty ratio D: a stance phase, when the leg is touching the ground ($\theta_i \le D$), and a swing phase, when the leg is in the air ($\theta_i > D$). The phase θ_i is calculated as follows:

$$\theta_i = \mod(f \cdot t + \theta_{\text{off},i}, 1), \tag{3.2}$$

where f is the walking frequency and $\theta_{\text{off},i}$ is the phase offset between the legs and the main parameter to determine the gait coordination.

The main parameters that characterize the foot trajectory are initial and final points of the stance phase (they define the stride length and width) and the height of the swing, as shown in Fig. 3.3. Touch-down and take-off angles can also be set in both lateral and longitudinal directions. The width of the swing and stance phase can be set separately. The stance trajectory is a line connecting points T_1 and T_2. The swing trajectory is generated by interpolating paths between points T_2, T_3 and T_3, T_1 with two cubic Bezier curves (Fig. 3.3). The advantage of using Bezier curves instead of a sine-shaped swing phase trajectory is the flexibility to define arbitrary touch-down and take-off angles. The duration of the stance and swing phases is determined by the duty factor D and the walking frequency f.

The legs are coordinated by defining a phase offset between them. The phase offset determines the relative relation of starting the stance phase between the legs. For example, a phase offset of 0.5 between left and right front leg means the right leg starts the stance phase half a cycle later than the left leg. Other than the relative relationship between the legs, a phase offset determines the relation between the legs and the spine. It should be set in a way it increases the maximum reach of the legs (e.g. the left hind leg starts the stance phase when the spine is bent to the left).

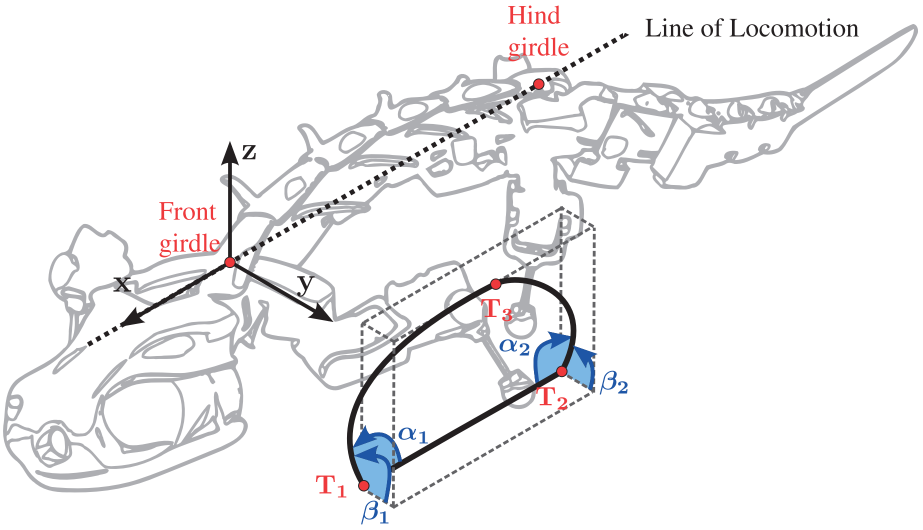


Figure 3.3 – Parametrization of Pleurobot's feet trajectories. The user defines points T_1, T_2 and T_3, as well as angles α_1, α_2, β_1 and β_2.

3.3.2 Leg inverse kinematics

To follow the foot trajectories, we need to solve the inverse kinematics problem for each of the legs. Let $\boldsymbol{p}_r \in \boldsymbol{\xi}_i$ (i.e. desired position is a point that belongs to the generated foot trajectory) and $\boldsymbol{p}_0$ be the desired and the current foot position in the Cartesian space, with respect to the girdle where the i-th leg is attached.

The distance between the current and desired foot position is denoted by $\Delta\boldsymbol{p}_r = \boldsymbol{p}_r - \boldsymbol{p}_0$. As the number of leg DoFs (4) is higher than the number of coordinates describing foot position (3), we have one redundant DoF and the solution of the inverse kinematics is not unique. The extra DoF can be used to adjust leg posture by driving current joint angles $\boldsymbol{q}_0$ towards the desired configuration $\boldsymbol{q}_r$. The difference between the current and desired leg posture is denoted by $\Delta\boldsymbol{q}_r = \boldsymbol{q}_r - \boldsymbol{q}_0$. Finally, we can state the inverse kinematics problem as an optimization problem which finds a change in joint angles $\Delta\boldsymbol{q}$ such the new joint configuration $\boldsymbol{q} = \boldsymbol{q}_0 + \Delta\boldsymbol{q}$ results in minimal values of $\Delta\boldsymbol{p}_r$ and $\Delta\boldsymbol{q}_r$:

$$\begin{aligned} \underset{\Delta\boldsymbol{q}}{\text{minimize}} \quad & \left\|\Delta\boldsymbol{p}_r - \boldsymbol{J}\Delta\boldsymbol{q}\right\|^2 + \lambda\left\|\Delta\boldsymbol{q}\right\|^2 + \left\|\Delta\boldsymbol{q}_r - \boldsymbol{J}_{\text{NULL}}\Delta\boldsymbol{q}\right\|^2_{\boldsymbol{M}}, \\ \text{subject to} \quad & \Delta\boldsymbol{q}_{\min} \le \Delta\boldsymbol{q} \le \Delta\boldsymbol{q}_{\max}. \end{aligned} \tag{3.3}$$

where $\|\cdot\|$ is the 2-norm and $\|\cdot\|_{\boldsymbol{M}}$ is a weighted 2-norm defined as $\sqrt{(\cdot)^\top \boldsymbol{M}(\cdot)}$, $\boldsymbol{M} \in \mathbb{R}^{4x4}$, λ is the damping coefficient, $\boldsymbol{J} \in \mathbb{R}^{3x4}$ is the leg Jacobian matrix and $\boldsymbol{J}_{\text{NULL}} \in \mathbb{R}^{4x4}$ is the null space projection of the Jacobian matrix. The Jacobian matrix is defined as $\boldsymbol{J} = \dfrac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}}$, while its null space matrix $\boldsymbol{J}_{\text{NULL}}$ is calculated from a singular value decomposition (SVD) of $\boldsymbol{J}$. The first term in the optimization cost function moves the foot towards the desired position $\boldsymbol{p}_r$. The second term penalizes big changes of joint angles $\Delta\boldsymbol{q}$ to assure smooth motion. The third term drives the joint angles towards the desired values $\boldsymbol{q}_r$ without affecting the foot position. If the constraints would not be considered, the minimum of the cost function could be found analytically. However to incorporate the joint limits, we decided to formulate it as a constrained optimization problem. Finally, we can represent it as a quadratic program (QP):

$$\begin{aligned} \underset{\Delta\boldsymbol{q}}{\text{minimize}} \quad & \frac{1}{2}\Delta\boldsymbol{q}^\top \left(\boldsymbol{J}^\top\boldsymbol{J} + \boldsymbol{J}_{\text{NULL}}^\top\boldsymbol{M}\boldsymbol{J}_{\text{NULL}} + \lambda\boldsymbol{I}_{4x4}\right)\Delta\boldsymbol{q} + \Delta\boldsymbol{q}^\top\left(-\boldsymbol{J}\Delta\boldsymbol{p}_r - \boldsymbol{J}_{\text{NULL}}^\top\boldsymbol{M}\Delta\boldsymbol{q}_r\right), \\ \text{subject to} \quad & \begin{bmatrix} -\Delta\boldsymbol{q} \\ \Delta\boldsymbol{q} \end{bmatrix} \leq \begin{bmatrix} -\Delta\boldsymbol{q}_{\text{min}} \\ \Delta\boldsymbol{q}_{\text{max}} \end{bmatrix}. \end{aligned} \tag{3.4}$$

To solve it, we use a QP solver qpOASES [49].

3.4 Leg-spine synchronization

3.4.1 Girdle control

The most important segments in Pleurobot's spine, from the ground locomotion point of view, are the girdles, since legs are attached to them. Therefore, the highest priority of the spine controller is to control the girdles' motion. What we want to control, besides relative position between girdles, are their orientations with respect to the girdle frames. We define the girdle rotation as an angle between the y-axis of the corresponding girdle frame and the line passing through both shoulders/hips (the blue line in Fig. 3.4 for the front girdle). Let's denote the front girdle angle as ϕ_F and the hind girdle angle as ϕ_H. The leg trajectories are then defined with respect to the corresponding girdle frame. While walking straight, a foot is following a half-sine shaped trajectory relative to the girdle (flat stance phase, positive half cycle of a sine wave for swing phase)[84]. Let ψ_{FL}/ψ_{FR} be an angle between the front left/right foot location and the y-axis of the FGF measured about z-axis (Fig. 3.4). In the case of the hind feet, those angles are denoted by ψ_{HL} and ψ_{HR} respectively. For the rest of the section, we will focus on only one girdle, since the control scheme is identical for both of them. For convenience, if the subscript $_F$ or $_H$ is dropped from the variable's name, it means the same relations apply to both front and hind girdle.

Because a shoulder/hip has an offset from the center of the girdle in lateral direction, girdle rotation influences the maximum reach of the leg. Fig. 3.5 shows how the relation between angles ψ_L and ϕ affects the reachable space of the front left leg. When ψ_L and ϕ are in phase (blue), which means the girdle rotates clockwise when the left leg is reaching forward and counterclockwise when the leg is pushing backwards, the reachable space is increased compared to the case when ϕ is fixed to zero (green). If they are in anti-phase, the reachable space decreases (red). These results indicate that it is beneficial for the girdle to oscillate by the angle ϕ which is proportional to the angle ψ_L (shoulder/hip always pointing towards foot location). In real sprawling posture animal's ground locomotion, this is the very essence of body-limb coordination

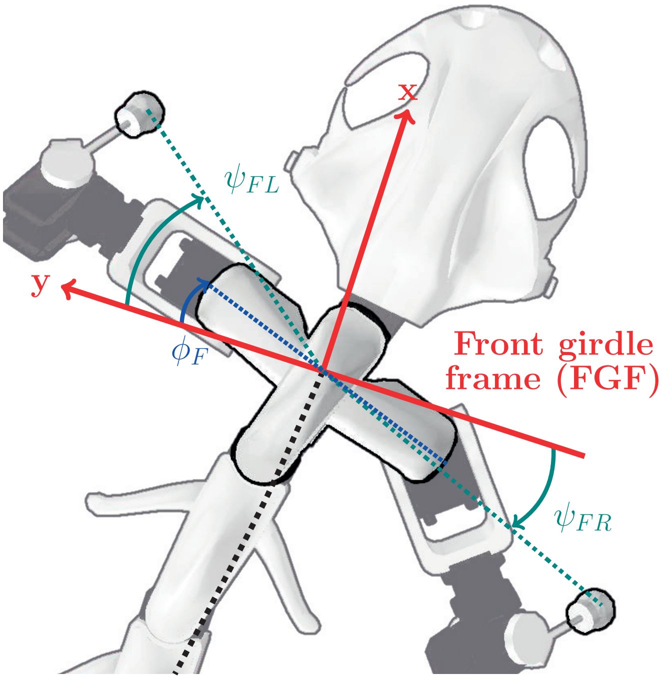


Figure 3.4 – The measure of angles between the front girdle (blue), feet (green) and the front girdle frame (red). The same principle applies to the hind girdle.

and the spine undulatory motion as a standing wave [98], [91].

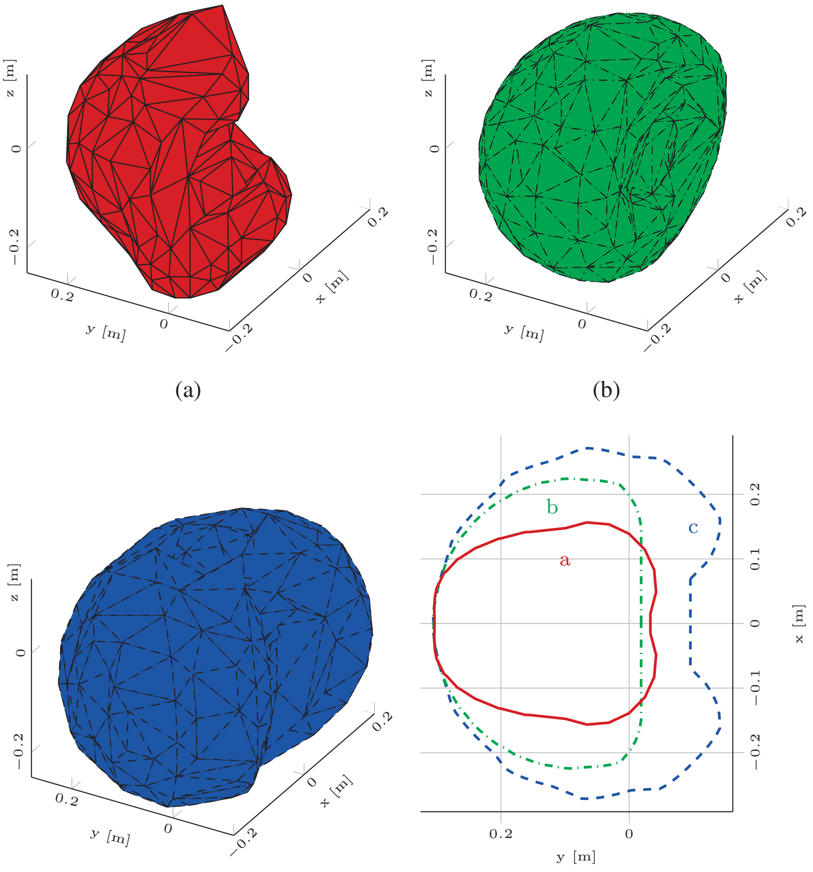
During normal walking, the right leg has a shift in phase compared to the left (typical shift value is set to π rad). To equally benefit both legs and maximize reach during walking, the girdle should be oscillating by the mean of angles ψ_L and ψ_R:

$$\psi_{\text{mean}} = \frac{\psi_L + \psi_R}{2}. \tag{3.5}$$

In case of half-sine foot trajectories, the signal ψ_{mean} is not necessarily smooth (differentiable) which, if used as a reference for the girdle oscillations, would result with jerky spine movements. Filtering the signal with a low pass filter would introduce a delay and distort the signal. Here, we take a different approach, inspired by use of phase oscillators as filters in [1]. We choose the girdle to follow the output ϕ_{ref} of the following phase oscillator, which is a sine signal:

$$\begin{aligned} \dot{r} &= a \cdot (R - r), \\ \dot{\theta} &= 2\pi f_{\text{walk}} + k \cdot (\theta_{\text{ref}} - \theta), \\ \phi_{\text{ref}} &= r \cdot cos(\theta), \end{aligned} \tag{3.6}$$

where a and k are respectively the amplitude and phase convergence parameters, R is the desired oscillation amplitude and θ_{ref} the desired phase. The value of R is chosen to match the amplitude of ψ_{mean} in Equation (3.5). The term $k \cdot (\theta_{\text{ref}} - \theta)$ drives the oscillator phase to follow the phase reference θ_{ref} [43]. The signal θ_{ref} is the phase of ψ_{mean} at the dominant frequency



(a) (b)

(c) (d)

Figure 3.5 – A reachable space of the front left foot in the function of leg-spine synchronization: girdle movement counteract the leg movement (a), no girdle movement (b), girdle movement helps leg movement (c). The top view (d) shows how the proper girdle movement can extend the reachable space of the leg.

which corresponds to the walking frequency f_{walk}. To determine it, we use Discrete Fourier Transformation (DFT). Since we know the current f_{walk} and gait parameters (duty factor, stride length, etc.) we can easily predict the future feet trajectories and from them the signal ψ_{mean}. Let N be the number of time steps we look ahead. Then the DFT of ψ_{mean} is given by:

$$X_{f_{\text{walk}}} = \sum_{k=0}^{N-1} \psi_{\text{mean}} \cdot \left(\cos\left(2\pi f_{\text{walk}} \cdot k \cdot dt\right) - \sin\left(2\pi f_{\text{walk}} \cdot k \cdot dt\right) \cdot i \right), \tag{3.7}$$

where dt is prediction time step which can be larger than a basic time step at which the control loop runs in order to avoid having large prediction horizon N. We have used the following values: $dt = 50$ ms and $N = 60$. Such choice of parameters allows for prediction of one walking cycle

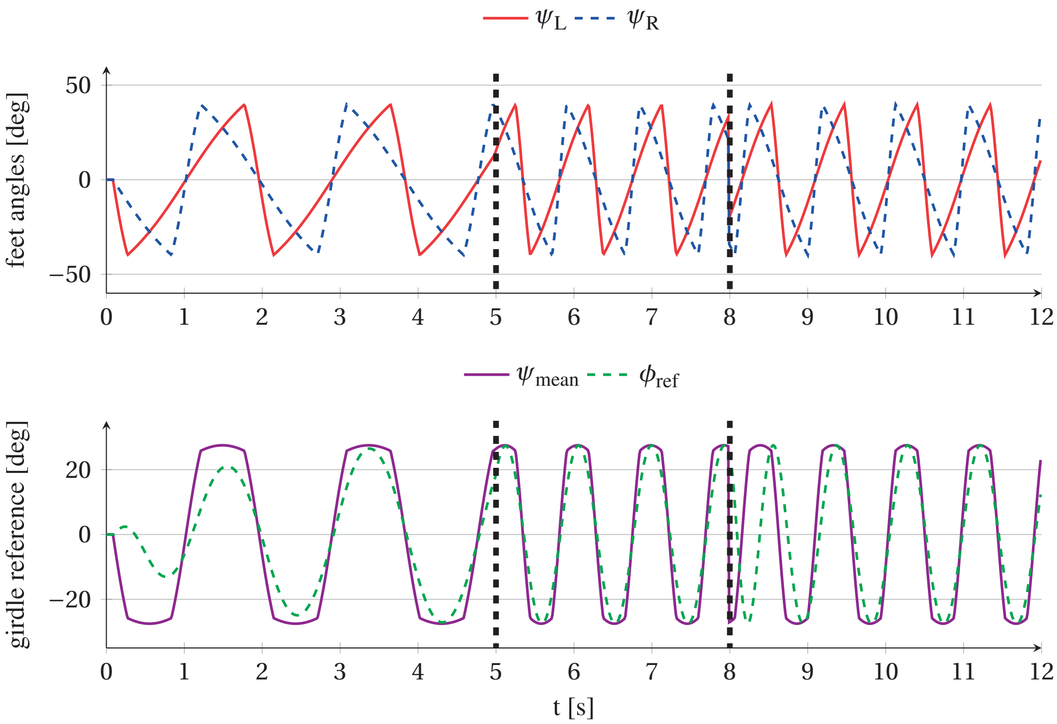


Figure 3.6 – The calculation of the reference for the girdle oscillation. At t = 5 s the frequency is changed from 0.53 Hz to 1.07 Hz (as the result of increasing walking speed from 0.2 $\mathrm{m\,s^{-1}}$ to 0.4 $\mathrm{m\,s^{-1}}$). At t = 8 s, the phase of the feet angles (top) is shifted for $\pi/2$ rad which results in abrupt change in the ψ_{mean} (bottom). In both cases, a phase of the oscillator is corrected (a phase of ϕ_{ref} converges to the phase of ψ_{mean}).

even for slow gaits at 0.33 Hz. From $X_{f_{walk}}$ we get θ_{ref} as follows:

$$\theta_{ref} = \operatorname{atan}\left(\frac{\operatorname{Im}(X_{f_{walk}})}{\operatorname{Re}(X_{f_{walk}})}\right). \tag{3.8}$$

The phase θ_{ref} obtained from the DFT modulates the phase of girdle phase oscillator according to equation (3.6). An example, with a sudden change of a walking frequency and a phase, can be seen in Fig. 3.6. The girdle oscillations successfully remain in phase with the leg movements. The described approach is done for both girdles separately (each girdle has assigned its own phase oscillator).

3.5 Spine inverse kinematics

Given reference angles $\phi_{F,ref}$ and $\phi_{H,ref}$ for the respective girdle motion, we need to ensure that positions P_F and P_H (i.e. the positions of the first and last segment of the spine) remain on the trajectory Γ as the robot progresses.

To do so, we need to solve the inverse kinematics of the spine. We choose the front girdle to be the base of the spine so it can already be brought into desired position and orientation. Solving the inverse kinematics will provide the joint angles which will ensure the desired position and orientation of the hind girdle.

A usual approach would be to find an explicit solution or use the Jacobian inverse method [19]. The former is not possible for redundant systems like our sprawling posture robot, and the weakness of the latter option is finding solutions near singularities of the kinematic chain when its Jacobian becomes ill conditioned. During walking, the spine oscillates laterally as a standing wave. Thus, all the joints are passing through their singularities (straight spine) every cycle, which makes it difficult to solve the inverse kinematics by using the Jacobian inverse method. Therefore we take a different approach which involves spline fitting as an initial guess for optimization.

3.5.1 Spline approximation

Since we have to match positions and angles of the first and last segment of the spine, a good initial guess for the spine joint angles is fitting a spline between girdles. One of the options is to use a cubic Hermite spline given by the equation:

$$p(s) = \quad (2s^3 - 3s^2 + 1)p_1 + (s^3 - 2s^2 + s)m_1 + (-2s^3 + 3s^2)p_2 + (s^3 - s^2)m_2, \tag{3.9}$$

where $s \in [0,1]$, m_1 and m_2 are tangent to the spline at the initial and final points p_1 and p_2. The point p_1 is placed at the position of front girdle P_F. Because we still don't know the spine joint angles, for the point p_1 we use the position of hind girdle from the previous time step P_H^-. This is good enough since the controller's refresh rate is relatively high (100 Hz) compared to the robot walking frequency (maximum 1 Hz). The tangents m_1 and m_2 are calculated from the girdle angles ϕ_F and ϕ_H.

We divide the spline into segments for the values of spline parameter s proportional to the length of the spine segments. The angles between spline segments $\boldsymbol{q_0} \in \mathbb{R}^5$ serve as an initial guess for the spine joint angles.

3.5.2 Optimization

Our assumption is that given the approximate angles $\boldsymbol{q_0}$, the desired hind girdle position and orientation can be achieved by an affine transformation of $\boldsymbol{q_0}$:

$$\boldsymbol{q} = \boldsymbol{q_0} \cdot x_1 + \mathbf{1} \cdot x_2, \tag{3.10}$$

where $x_1, x_2 \in \mathbb{R}$ are respectively scaling and offset parameters, $\mathbf{1} \in \mathbb{R}^5$ is a vector of ones and $\boldsymbol{q} \in \mathbb{R}^5$ is the solution of the inverse kinematics. To determine x_1 and x_2 we formulate a nonlinear optimization problem:

$$\begin{aligned}
\underset{x_1,x_2}{\text{minimize}} \quad & k_1||P_H-\Gamma||+ \\
& k_2(\phi_{H,\text{ref}}-\phi_H)^2+ \\
& k_3\left((x_1-1)^2+x_2^2\right)+ \\
& k_4((x_1-x_1^-)^2+(x_2-x_2^-)^2) \\
\text{subject to} \quad & x_{1,\min}\le x_1\le x_{1,\max} \\
& x_{2,\min}\le x_2\le x_{2,\max}
\end{aligned} \tag{3.11}$$

The cost function is composed of four weighted terms. The first term is the distance of the hind girdle position P_H from the trajectory Γ. To get the P_H, the forward kinematics of the spine has to be solved during every evaluation of the cost function. The second takes care of the orientation of the hind girdle. The third term penalizes the influence of the optimization variables to prevent high deviations from the initial angles $\boldsymbol{q_0}$. The fourth and last term penalizes big changes of optimization variables in consecutive time steps to get a smooth motion of the spine (x_1^- and x_2^- are the solution from previous time step).

The weights $k_1...k_4$ were tuned manually. The values that worked satisfactorily in our case were $k_1 = 100$, $k_2 = k_4 = 20$ and $k_3 = 1$. An example of using the proposed method for solving the spine inverse kinematics is shown in Fig. 3.7. Fig. 3.8 shows how the proposed method works for different number of spine joints. The optimization step is reducing the positioning error of the hind girdle for at least an order of magnitude which allows for precise control of the hind girdle. Although the secondary objective (an average angle error) increases in some cases, we still find it acceptable and it could be further decreased by tuning the optimization weights for each of the spines individually (we used the same set of weights for all the spines). To solve the optimization problem above, we used Dlib C++ Library [106]. On the robot, which is powered by an ARM based computer ODROID-XU4, solving the problem takes less than 1 ms.

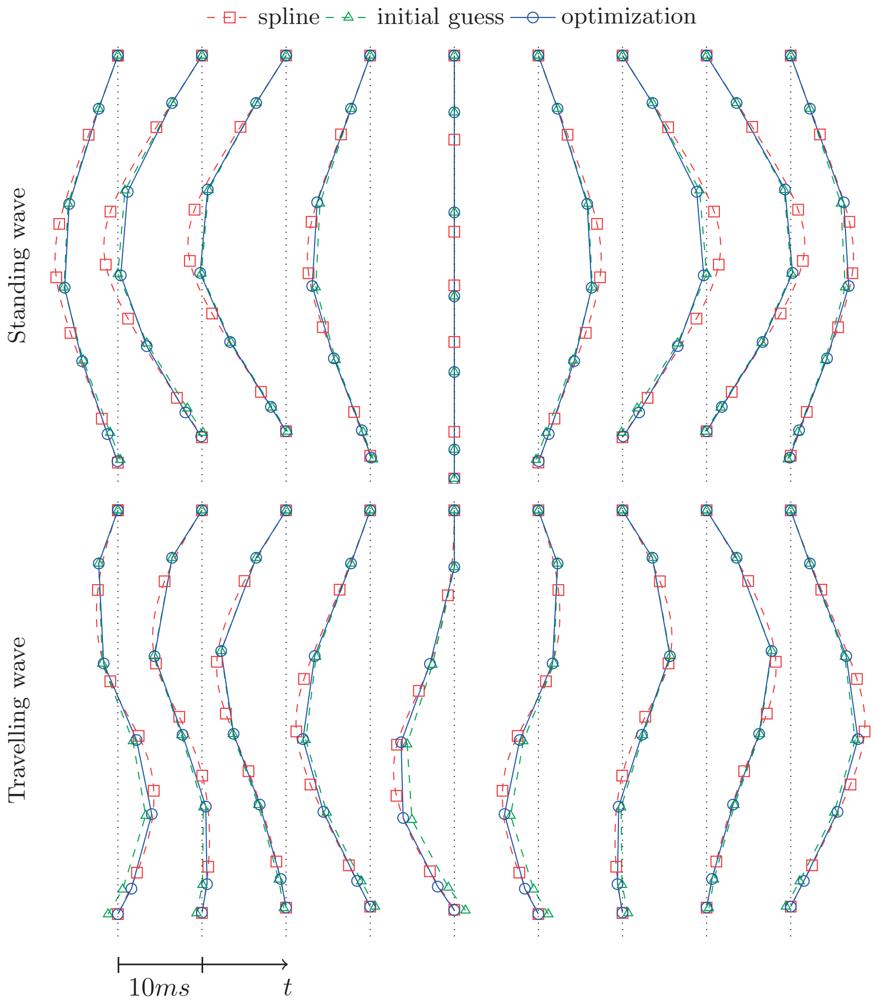


Figure 3.7 – Steps in solving the inverse kinematics of the spine. A spline between the front and hind girdle is shown in red. By applying the angles between spline segments to the spine, we get an initial guess for the spine angles, which is shown in green. The optimization step does minor adjustments to the spine angles (shown in blue) in order to achieve the desired position (the dashed line) and orientation of the hind girdle. Depending on the offset between girdles' oscillations, the spine can take different shapes. For example, the offset of π rad results with a standing wave along the spine (top) and the offset of $\pi/2$ rad results with a traveling wave (bottom). In both cases the proposed algorithm finds the solution.

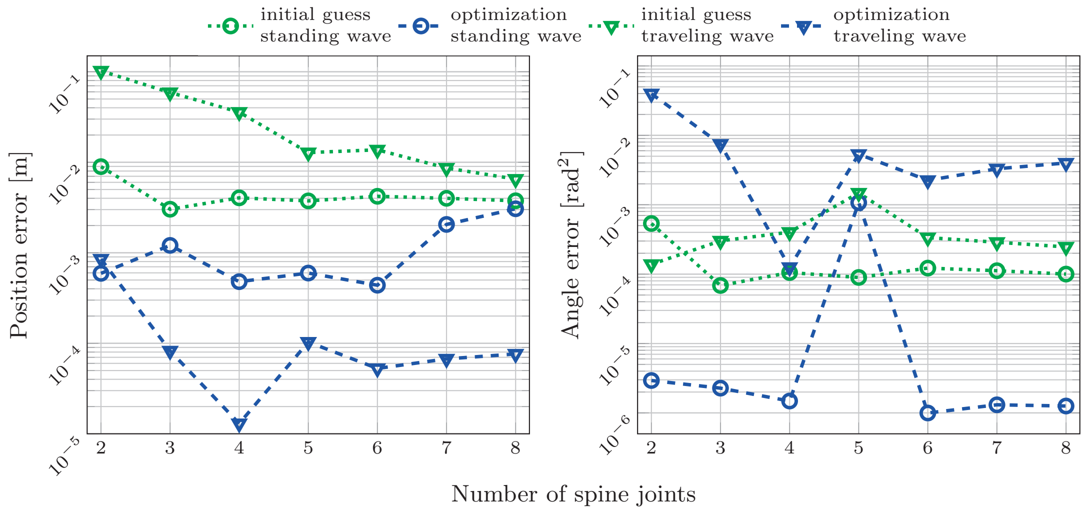


Figure 3.8 – Comparison of the hind girdle position and orientation error prior to and after the optimization step. The position error is the maximum distance of the hind girdle from the trajectory Γ over a walking cycle. The angle error is an average squared error of the hind girdle orientation. The proposed method is tested on spines with different number of joints. All the spines have the same total length as Pleurobot's five joint spine. The extra spines have an uniform distribution of their joints (unlike Pleurobot's [98]).

3.6 Walking and turning

The movement of the robot is controlled by providing the linear (v_F) and angular (ω_F) velocities for the FGF (defined in Fig. 3.2). Those references are here provided by a robot operator, but are also suitable to be set by path planning/following algorithms [145]. The actual movement is achieved by moving front feet in their respective stance phase with velocities $-v_F$ and $-\omega_F$ relative to the FGF. The trajectory Γ is obtained by integrating FGF movements over time with respect to a world frame. It is not necessary to remember the entire history of the Γ. Since we only use it to determine the placement of the hind girdle, it is only needed to store the length of it which corresponds to the inter-girdle distance (IGD) of the robot (IGD = 0.48 m, in the case of Pleurobot [98]).

The position of HGF is obtained by solving the forward kinematics of the spine (as it depends on the spine movements). The orientation of the HGF (defined in Fig. 3.2) is tangent to Γ in the closest point to the position of HGF. This choice of moving the HGF does not contradict the spine movements and keeps the same orientation of the HGF as the FGF had in the same point on Γ. The linear (v_H) and angular (ω_H) velocities of the HGF are obtained by differentiating its position and orientation between two consecutive time steps.

While turning with the robot, it is necessary to limit the turning radius to avoid self collisions. The turning radius of a body, with velocities v_F and ω_F, is equal to the ratio v_F/ω_F. By limiting the maximum value of the ω_F in the function of v_F, we can limit the minimum radius the robot will use to turn. The experiments showed that the turning radius of 0.5 m (which is close to one IGD) is still acceptable[1] so we use the rule $\omega_{F,\max} = 2 \cdot v_F$ to limit the maximum commanded angular velocity.

To avoid additional spine bending and possible self collisions while turning with the minimum radius, the girdle oscillations ϕ are gradually decreased when the turning radius decreases by changing the amplitude of the phase oscillator (Equation (3.6)). The amplitude R is linearly decreased: $R = (1 - \omega_F/\omega_{F,\max}) \cdot R_{\max}$, where $R_{\max}$ is the oscillation amplitude when the robot walks straight.

3.7 Experiments

A series of experiments was carried out to test the walking and turning capabilities of the robot while using the proposed spine controller. First, we simulate the robot by using a model of the robot in a virtual physical simulation environment to test the feasibility of the experiment. Then, by using the robot hardware, experiments were performed to test the output of the controller under real conditions (e.g. with real contact and motion dynamics like feet slipping, limited servo speed and limited torque). In this work, we focus on the girdle trajectories in order to measure

---

[1] We did not encounter self collisions or leg reachability problems since, at the moment, the controller is not finding the optimal foot placements as a function of a turning radius.

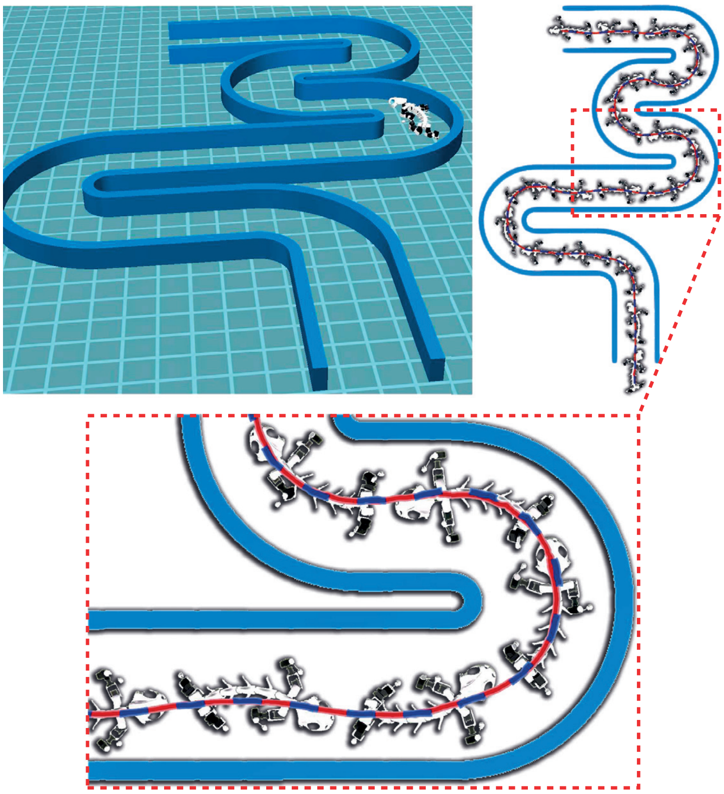


Figure 3.9 – The setup of the simulation experiment. The track consists of 5 turns and straight segments in between. The curvature radius of the first two turns is 0.7 m and 0.6 m respectively, while the last three have it set to 0.5 m. The width of the track is 0.7 m, while the used step width of the robot is 0.32 m. The linear velocity of the robot was kept constant at 0.3 $\mathrm{m\,s^{-1}}$.The red and dashed blue lines are showing the positions of the front and hind girdles respectively during the run. The snapshots of the robot are spaced 3 s apart.

how close the hind girdle follows the front girdle. For each trial, the robot was commanded to walk over a track with sharp turns and fixed dimensions.The commands v_F and ω_F were given by a human operator using a gamepad in real time for both the simulated and the real robot.

3.7.1 Simulation

To simulate the robot and its environment, we use the robot simulator Webots$^{\mathrm{TM}}$. We created a track with turns and straight segments to test the maneuverability of the simulated robot. Fig. 3.9 shows that the hind girdle closely follows the trajectory done by the front girdle. The red line indicates the human-operated trajectory traced by the front girdle, while the dashed blue line shows the trajectory of the hind girdle generated by the controller. The simulation results (Fig. 3.9) validate the theoretical predictions of the controller. The tracking error between the two trajectories was fairly small in this simulation controlled environment, averaging at 2.35 cm. The simulations show that the controller allows the robot to precisely go around corners without touching the walls.

Moreover, as the speed was set, the operator was limited to steer the robot trajectory left and right. The reduction in turning radius along the path (i.e. increasing the turn difficulty), allows the operator to get familiar with the steering controller. The simulation experiments carried out then serve as a training stage for the hardware experiments.

3.7.2 Hardware experiments

We carried out two hardware experiments to measure the effectiveness of the controller. In the first experiment we created a track similar to that of the simulation experiments to measure the error of the hind girdle tracking along the path. Then we performed walking in a circle as a way to quantify the cumulative error deviation after several turns. For both experiments we used Naturalpoint Inc.-Optitrack® motion capture system of 12 cameras. Tracking data was obtained using Motive™ software. Markers were sent on each of the girdles. Additionally, video[2] was taken with an overhead camera at 4K and a second camera at 1080p for perspective views.

Girdle tracking along the path

For the first set of hardware experiments, we prepared a track with 3 turns, each of them with a curvature of 0.5 m. The track width is also 0.7 m. The maximum linear velocity of the robot was set to 0.2 $\mathrm{m\,s^{-1}}$ in order to avoid servo motor limitations. The run was conducted multiple times with similar results despite the robot being operated by a human. Fig. 3.10 shows that the hind girdle still manages to stay close to the trajectory done by the front girdle. For the section of the track covered by motion capture system, we could precisely calculate the Euclidean distance (in the ground plane) between the paths done by the front and hind girdles. The mean distance and deviation over 10 runs in the function of robot's position in the tracked area is shown in Fig. 3.10c. The mean distance between girdle paths is around 2.5 cm which is 5.8 % of the IGD.

Walking in a circle

Fig. 3.11 shows an overlay of front and hind girdles trajectories, tracked by motion capture system while the robot was walking in circles with a constant turning radius of 0.5 m. The duration was 100 s, which was enough for around 6 circles. Qualitatively we can see the both girdles are staying on the circle with a radius equal to the specified turning radius. Quantitatively, the error between trajectories of front girdle and hind girdle also presents similar results or better than that of the track experiment above. Over time, a deviation of about 7 cm of the overall trajectory of the whole robot from the original circle is noticed. This is due to some slippage present between the robot feet and the locomotion surface. These experiments were performed and reported both clockwise and anti-clockwise with similar results.

---

[2] The video of the experiments can be fount at: https://biorob2.epfl.ch/video/311

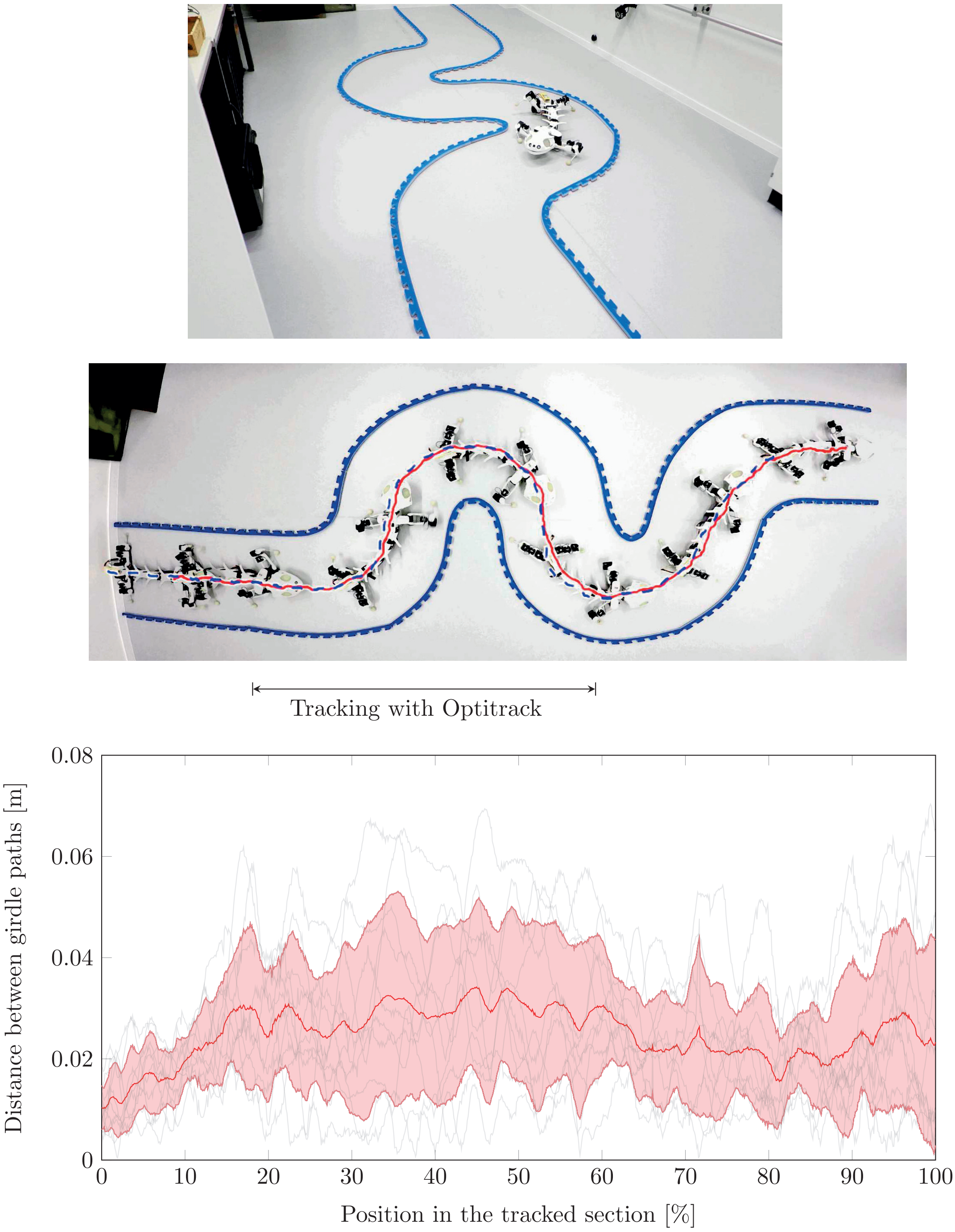


Figure 3.10 – The setup of the hardware experiment - perspective view (top) and overhead view (middle). The red and blue line in the overhead view (middle) are showing the positions of the front and hind girdles respectively during the run, tracked by Matlab from the overhead camera video. The snapshots of the robot are spaced 6 s apart. In the bottom we can see the mean distance and deviation between paths done by front and hind girdles in the section of the track covered by the motion capture system.

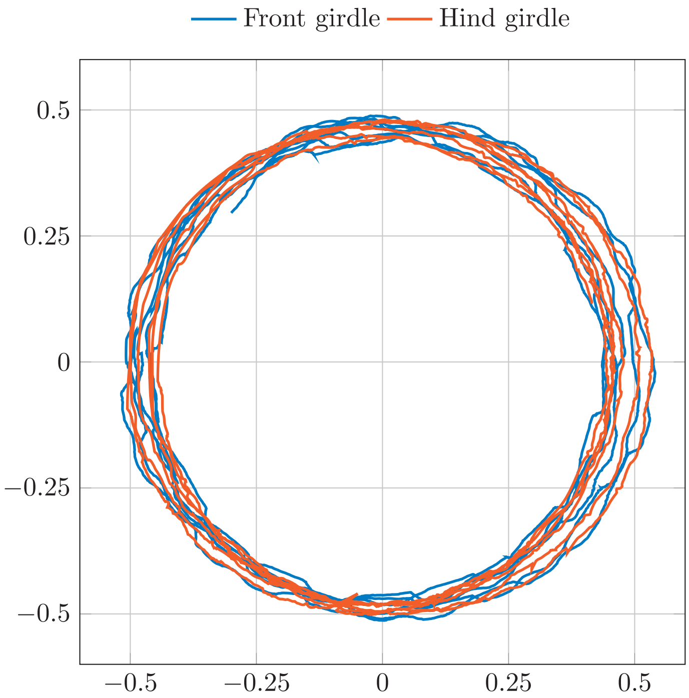


Figure 3.11 – An overlay of girdle trajectories captured while walking with a constant turning to the left for 100 s. The robot did not try to stay on a circle with help of the motion capture system. It was solely commanded to turn with a constant turning radius.

3.8 Discussion and conclusion

We presented a spine controller for a sprawling posture quadruped robot with segmented spine. We assigned two main roles to the spine. First, it extends the reach of the legs by rotating the girdles and second, the spine serves as main mechanism for performing turning motions.

Due to the girdle controller we presented here, the synchronization between the legs and spine does not need to be manually tuned, improving the existing controllers [91, 84]. Previously the robot used a standing wave along the spine (i.e. phase between the girdles of π rad), featuring a girdle oscillation in counter phase, which was inspired by the salamander animal [98]. In order to improve the balance of the robot in the current hardware experiments, the offset between front and hind legs was manually set to 0.7π rad. Such phase offset results with a traveling wave along the spine. During turning, the controller suppresses the oscillation in order to avoid too much bending that can lead to self collisions.

The experiments show that the paradigm of the hind girdle following the front made it easy for a human operator to control the robot walking over a track with sharp turns. Interestingly, the results of the quantification of the tracking error between front and hind girdle trajectories along the proposed experimental track show that the error remains small and consistent along a path. In the case of our experiments, a path of 6.7 m (about 14 times larger than the robot IGD) was used. The controller proved good tracking performance of the the front girdle trajectory with the hind one. An error of less than 6 % of the robot's IGD length (i.e. error 84 times smaller than the track length) was obtained. This result was computed as the minimum distance between girdle paths (i.e. how close the hind girdle passes over the same points that the front girdle passed), validating the proposed controller and its usefulness for further developments in control of sprawling posture robots.

The main challenge of this controller was to enable smooth turning by synchronizing the spine and leg motion automatically. Additional experiments of circle turning with the robot using the proposed controller, showed that the robot manages to accomplish commanded turning radius. Even after walking for about 18 m (completing 6 circles), the robot did not deviate significantly from the initial path.

In the future we want to use similar robots (sprawling posture, segmented spine) for search and rescue scenarios. More precisely, we aim to tackle shallow water scenarios (Robots like Pleurobot are equipped with a tail, allowing them to swim [98]) and hardly accessible cluttered places like pipes with sediments. By carefully tuning the robot's walking (or crawling) gait it is possible to reduce its step width even further (the shoulder width of Pleurobot is 0.19 m). The track width could be further reduced from the one we used in the presented experiments. The turning radius could also be reduced by adapting the feet landing positions in function of the robot's angular velocity to increase the possible range of motion (e.g. while turning the outer leg has a longer stride than the inner leg). Another feature worth exploring and still useful in the aforementioned scenario is the ability to produce reverse motions by adjusting the same controller backwards.

Furthermore, as the control of steering and speed with these robots is now enabled with our proposed controller, path planning and navigation algorithms can easily be implemented in order to provide autonomous behaviors. For example, a future goal is closing the loop with perception systems and running SLAM algorithms in order to increase robot's search and rescue capabilities. Our next step is to improve the traversability capabilities of our robot by addressing issues like the center of mass balance with high postures, vertical bending of the spine and force feedback to tackle complex 3D environments with possible amphibious water to land transitions.

4 Model Predictive Control Based Framework for CoM Control of a Quadruped Robot

The locomotion controller described in the previous chapter requires the user to specify most of the gait parameters (feet trajectories, timings, duty factor etc.). The parameters have to be hand tuned to achieve a well balanced walk, meaning the body roll and pitch angles are kept small. Even though flipping over is unlikely, badly balanced gait results in a stumbling behavior where the swing leg often touches the ground. In this chapter we look into how to alleviate the need for hand tuning the gait by using model predictive control techniques. The newly developed control module serves as an add-on to the locomotion controller.

The material presented in this chapter is adapted from:

[85] Horvat, T., Melo, K., and Ijspeert, A. J. Model predictive control based framework for CoM control of a quadruped robot. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on* (2017), IEEE, pp. 3372–3378.

My contributions: Fully designed the controller and programmed the robot, designed and carried out the experiments, writing.

4.1 Introduction

Legged robots bring many advantages over wheeled and tracked platforms, especially in traversing rough and complex terrain where the advantage of having legs becomes obvious. At the same time, they are more complex, in both mechanics and control. Being often inherently unstable, the performance and capabilities of legged robots are greatly defined by their control algorithms.

A lot of research is done on improving the stability of robots while executing different gaits. A robot's stability can be characterized by its center of mass location (CoM) during static walking and zero moment point (ZMP) during dynamic walking. In this chapter we focus on the stability of a position controller, static walking robot, but also dynamic gaits will be tested. A broad overview of that topic can be found in [71] and many other teams are doing related research. In [20, 69] the CoM and ZMP of a quadruped robot were controlled by using a preview control method which was first implemented on a biped robot [94]. In [150, 18] researchers present a work on generating CoM trajectories based on legs movement. The latter approach relies on the knowledge of a robot's walking sequence.

We aspire to make a CoM control framework which will be as independent as possible of the utilized walking gait characteristics: phase offset between legs, duty factor, stride length and width, frequency and finally the walking direction relative to the robot's body. As long as we can approximately predict a near future leg motion, a suitable approach which can satisfy our requirement for generality is Model Predictive Control (MPC) [60]. Such control strategy has already been widely used in bipedal robotics to adapt positions of footsteps ensuring the robot's stability [48, 76]. A similar approach can be applied to a quadruped robot [13, 135]. Through constraints and user specified cost function, MPC can generate a stable CoM trajectories which will dynamically adapt to the changes in the walking gait. Such feature makes it also convenient for non-periodic walking.

In the chapter, we start with describing the MPC based framework for the center of mass reference generation and control. The proposed solution is furthermore validated on a sprawling posture quadruped robot, in both simulation and the real robot. Although the tests are done on a single robot, the framework is applicable to any quadruped robot that can control its feet position in 3D space (e.g. with at least 3 degrees of freedom per leg which allow for leg abduction/adduction, protraction/retraction and extension/flexion). In the end we discuss possible applications, improvements and extensions of the presented framework.

4.2 Center of mass control framework

The main goal of the control framework presented here is to improve the balance of a static walking quadruped robot. A robot's posture is stable if its zero moment point (ZMP) lies within a support polygon formed by the contact points of the legs in stance phase [189]. If the robot's body is close to the ground and its accelerations are small, we can approximate the ZMP location with the projection of the CoM on the ground. Another assumption we make is that the robot walks on a horizontal ground plane and its body attitude is kept horizontal. In that way, we can describe the position of CoM (and its projection) with only two variables (x, y). In further text when we mention CoM, we refer to its ground projection.

Although these assumptions are restrictive, we will show how the controller behaves on the simulated and on the real robot, where those assumptions do not always apply. In the final section we discuss how some of the assumptions can be alleviated.

If we look at the control hierarchy of a legged robot system, the low level control would for instance include leg inverse-kinematics and dynamics, individual servo control, high speed

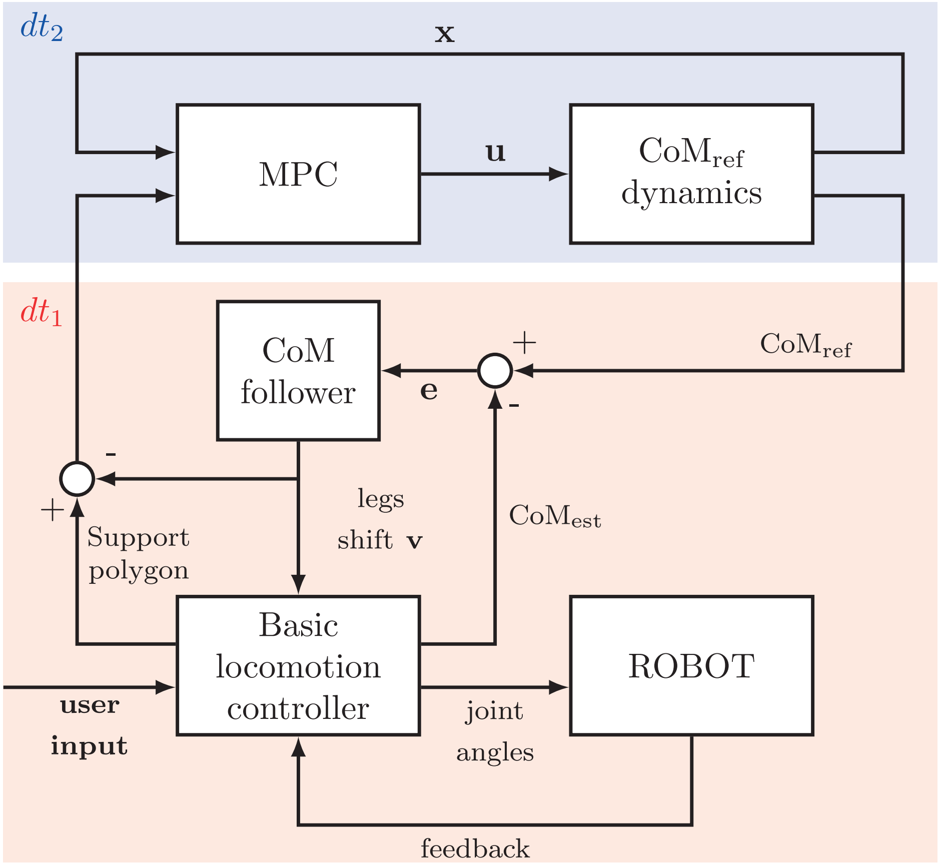

dt_2
x
MPC
u
CoM_{ref} dynamics
dt_1
CoM follower
e
+
-
CoM_{ref}
-
+
Support polygon
legs shift v
CoM_{est}
Basic locomotion controller
ROBOT
user input
joint angles
feedback

Figure 4.1 – An overview of the control architecture. The basic locomotion controller (BLC) coordinates the robot's body, solves inverse kinematics and translates high-level (user) inputs to the locomotion. The MPC based CoM controller is an additional module that can be put next to the BLC, providing a body posture adjustment. The colored rectangles indicate the high speed control loop of the BLC (red) with the sampling time dt_1 and the slower MPC control loop (blue) with the sampling time dt_2.

reflexes [84] etc. Similarly, the high level control would include a motion planner, trajectory follower or direct walking velocity and direction commands. In such a robot control architecture, our control framework would be the middle level module which provides posture adjustments to the low level control (Fig. 4.1).

Our approach of controlling the position of CoM is based on the Model Predictive Control (MPC) which acts on the dynamical system representing the desired CoM dynamics. The main feature of MPC is the receding prediction horizon over which the controller (i) predicts future dynamics of the system, (ii) satisfies system constraints, (iii) optimizes user specified criteria and (iv) obtains an optimal control sequence[1]. In every time step, the computation repeats and only the first value of the optimal control sequence is applied to the system. Output of the system is then used to provide a reference CoM$_{\text{ref}}$ for the movement of the actual CoM of the robot. A feedback the MPC is getting from the low level controller are current and predicted support polygons formed by the robot's legs.

4.2.1 CoM dynamics

We will consider a desired motion of CoM$_{\text{ref}}$ in x (longitudinal) and y (transversal) direction to be mutually independent. Motion in each direction is modeled as a 3rd order dynamical system, similarly to the one used in [76]. Consequently, as control inputs we choose to use the jerks $\dddot{x}_{cr}$ and $\dddot{y}_{cr}$ to guarantee smooth motion of the CoM$_{\text{ref}}$, with piecewise constant jerks over an interval of the sampling time dt. The state space representation of such a 6th order discrete dynamical system, where CoM$_{\text{ref}}$ position is described with variables x_{cr} and y_{cr} is:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{4.1}$$

where:

$$\mathbf{x} = \begin{bmatrix} x_{cr} & \dot{x}_{cr} & \ddot{x}_{cr} & y_{cr} & \dot{y}_{cr} & \ddot{y}_{cr} \end{bmatrix}^T, \quad \mathbf{u} = \begin{bmatrix} \dddot{x}_{cr} & \dddot{y}_{cr} \end{bmatrix}^T,$$

$$\mathbf{A} = \begin{bmatrix} 1 & dt & dt^2/2 & 0 & 0 & 0 \\ 0 & 1 & dt & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt & dt^2/2 \\ 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} dt^3/6 & 0 \\ dt^2/2 & 0 \\ dt & 0 \\ 0 & dt^3/6 \\ 0 & dt^2/2 \\ 0 & dt \end{bmatrix}.$$

To decouple the CoM$_{\text{ref}}$ from the robot's frame, we need to compensate for posture adjustments caused by following the CoM$_{\text{ref}}$. Since the only feedback the MPC is getting from the robot are the support polygons, it is sufficient to substract the posture adjustment from them.

---

[1] Optimal from the standpoint of the user specified criteria.

4.2.2 MPC problem formulation

The user is given flexibility in formulating a MPC problem. In this case, the MPC should drive $\mathrm{CoM_{ref}}$ to the desired state while trying to keep it within a support polygon and satisfy the constraints on its position, velocity and acceleration. With those guidelines in mind we can formulate the following optimization problem:

$$
\begin{aligned}
\min_{\mathbf{x},\mathbf{u},\boldsymbol{\epsilon}} \quad & \sum_{k=0}^{N-1} (\mathbf{x}_k - \mathbf{r}_k)^T \mathbf{Q} (\mathbf{x}_k - \mathbf{r}_k) + && \text{(4.2a)}\\
& (\mathbf{x}_N - \mathbf{r}_N)^T \mathbf{Q}_N (\mathbf{x}_N - \mathbf{r}_N) + \\
& \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k + \sum_{k=1}^{N} \boldsymbol{\epsilon}_k^T \mathbf{P} \boldsymbol{\epsilon}_k
\end{aligned}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k && k = 0, \dots N-1 && \text{(4.2b)}\\
& \mathbf{F}\mathbf{x}_k \leq \mathbf{f} && k = 1, \dots N && \text{(4.2c)}\\
& \mathbf{M}\mathbf{u}_k \leq \mathbf{m} && k = 0, \dots N-1 && \text{(4.2d)}\\
& \mathbf{S}_k\mathbf{x}_k \leq \mathbf{s}_k + \boldsymbol{\epsilon}_k && k = 1, \dots N && \text{(4.2e)}\\
& \boldsymbol{\epsilon}_k \geq \mathbf{0} && k = 1, \dots N && \text{(4.2f)}
\end{aligned}
$$

The weight matrices $\mathbf{Q} \in \mathbb{R}^{6\times6}$, $\mathbf{Q}_N \in \mathbb{R}^{6\times6}$ and $\mathbf{R} \in \mathbb{R}^{2\times2}$ define balance between an aggressive control $\mathbf{u}$ and how well the system states $\mathbf{x}$ follow the state reference $\mathbf{r} \in \mathbb{R}^{6\times1}$ over the prediction horizon N. The equality constraint (4.2b) assures the correct prediction of the system's dynamics described in (4.1). The inequality constraints (4.2c) and (4.2d) are designed as box constraints on states $\mathbf{x}$ and control $\mathbf{u}$.

The relation $\mathbf{S}_k\mathbf{x}_k \leq \mathbf{s}_k$ in (4.2e) is a polytopic constraint on the position of $\mathrm{CoM_{ref}}$ within the support polygon. To ensure the optimization problem does not become infeasible if such constraint gets violated (which can happen during robot's locomotion) we introduce positive slack variables $\boldsymbol{\epsilon}_k$. If $\mathbf{S}_k\mathbf{x}_k \leq \mathbf{s}_k$ is satisfied, the optimal solution for slack variables is $\boldsymbol{\epsilon}_k = 0$ and they do not contribute to the cost function. Otherwise, they will have a positive value and will contribute to the cost function (4.2a). With relatively high values of the weight matrix $\mathbf{P}$ compared to others, the optimal solution will satisfy the condition $\mathbf{S}_k\mathbf{x}_k \leq \mathbf{s}_k$ whenever it is possible. Since we are dealing with a quadruped robot, which can at most have four legs on the ground at any moment (support polygon is quadrilateral), it is valid $\mathbf{S}_k \in \mathbb{R}^{6\times4}$, $\mathbf{s}_k \in \mathbb{R}^{4\times1}$, $\mathbf{P} \in \mathbb{R}^{4\times4}$ and $\boldsymbol{\epsilon}_k \in \mathbb{R}^{4\times1}$.

Since we assume it is possible to predict future leg motion based on their current state and high level inputs to the basic locomotion controller, we can predict support polygons over the horizon N. That is reflected in $\mathbf{S}_k$ and $\mathbf{s}_k$ changing over time and prediction horizon, allowing MPC to anticipate support polygon changes.

The i-th row of matrix $\mathbf{S}_k$ and vector $\mathbf{s}_k$ is formed as follows:

$$\mathbf{S}_k(i,:)=\left[\frac{p_{y,i,k}-p_{y,i+1,k}}{||p_{k,i}-p_{k,i+1}||} \quad 0 \quad 0 \quad \frac{p_{x,i+1,k}-p_{x,i,k}}{||p_{k,i}-p_{k,i+1}||} \quad 0 \quad 0\right], \tag{4.3a}$$

$$\mathbf{s}_k(i)=p_{x,i,k}\frac{p_{y,i,k}-p_{y,i+1,k}}{||p_{k,i}-p_{k,i+1}||}+p_{y,i,k}\frac{p_{x,i+1,k}-p_{x,i,k}}{||p_{k,i}-p_{k,i+1}||}-s_m\frac{k}{N}, \tag{4.3b}$$

where position of i-th leg in stance at moment k (where k is position in the prediction horizon) in robot frame of reference is $p_{k,i}=\left[p_{x,k,i}, p_{y,k,i}\right]^T$. The order of legs is: front left, front right, hind right, hind left and again front left to close the support polygon. The legs that are in swing phase are skipped, which results in the number of non-zero rows in $\mathbf{S}_k$ being less than 4. As we move through the prediction horizon, uncertainty grows. To compensate for errors in the support polygon prediction, the term $-s_m k/N$ is added to the computation of the vector $\mathbf{s}_k$, effectively reducing the size of the support polygon towards inside. In the final prediction step $k = N$, all the edges of support polygon will be moved inwards for the user specified constant s_m (measured in meters).

Practical considerations

The optimization problem described in (4.2) can be brought into a standard QP form [191]. To solve it we use a QP solver qpOASES [49]. Since solving a QP problem (4.2), specially with a longer prediction horizon N is a relatively complex task compared to the computation requirements of the locomotion controller, we decided to run the MPC and CoM_{est} at a lower frequency than the fast low level control loop. Values for sampling times of both control loops are shown in Table 4.2. Having a larger MPC discretization time dt_2 means that the prediction horizon reacher further into the future $(N \cdot dt_2)$ for a lower N, which simplifies the computation. From the implementation standpoint, the MPC control loop can be implemented in the separate thread from the locomotion controller, avoiding the computation bottleneck.

Although this MPC formulation allows to follow a time varying state reference r, in this case the reference is kept constant at the location of CoM when the robot stands still.

4.2.3 CoM follower

The MPC generates a position reference for the center of mass CoM_{ref} in the robot frame of reference (Fig. 4.2). An estimated position of the center of mass CoM_{est} is calculated within the locomotion controller by using the robot's forward kinematics and known masses of each of its links. To match the positions of CoM_{ref} and CoM_{est}, the robot needs to move its legs in the ground plane (x, y) for the value $\mathbf{v} \in \mathbb{R}^2$, which is generally in the opposite direction of the error $\mathbf{e} = \text{CoM}_{\text{ref}} - \text{CoM}_{\text{est}}$. However, such action also moves the robot's frame of reference. This means that for the most part[2], the shift of the CoM will not be visible in the variable CoM_{est}.

---

[2] Small change is caused by shifting the mass in the legs.

Therefore, during the tracking of the constant $\mathrm{CoM_{ref}}$, the error $\mathbf{e}$ does not converge to zero over time, yet it settles to a constant value which assures a perfect match of $\mathrm{CoM_{ref}}$ and $\mathrm{CoM_{est}}$ in the ground frame of reference. Knowing the dynamics of the $\mathrm{CoM_{est}}$ as a function of the posture adjustment $\mathbf{v}$, one can design a CoM follower controller which will act on the error $\mathbf{e}$ to get the shift $\mathbf{v}$. In our case, where we chose to have a 5 times faster update rate (Table 4.2) of the CoM follower compared to the rate of the $\mathrm{CoM_{ref}}$, a simple proportional controller with a gain of -1 provides satisfactory results.

4.3 The platform

The robot we use to test our control approach is Pleurobot: a sprawling posture quadruped with segmented spine. Its design is inspired by a salamander and allows for multimodal locomotion (walking and swimming) [98]. Since we focus solely on terrestrial locomotion and want to stay as close as possible to the general form of quadruped robots, we use it in its tailless form. The tail has has an important role for swimming, but at this point it is not used during walking. Each of the legs has four degrees of freedom and the spine has five, which allows the trunk bending in the lateral direction.

The low level inverse kinematics controller of Pleurobot is presented in [84], while in [86] we presented the spine controller which extends the legs' reach and improves turning capabilities of the robot. In the latter work we discussed the problem of defining a reference frame of such a robot with segmented spine. An effective control of the spine required us to define two local coordinate frames attached to the front and the hind girdle (attachment points for the legs). Now, we need to calculate and control the position of CoM in a single reference frame. Therefore, we define a third reference frame, whose origin is attached to the front girdle and x-axis passes through both girdles (such reference frame is used in [84]). The robot frame is illustrated in Fig. 4.2. In this chapter all values and variables are expressed in the robot frame of reference.

The robot is capable of executing a wide range of different walking gaits, although in our previous work we have mostly used low and wide posture gaits with relatively high duty factor of 0.8. In terms of maneuverability, in addition to moving forwards/backwards, the robot can turn with a turning radius of one IGD (inter-girdle distance) [86] and also walk sideways (sidestepping) without changing its heading direction (Fig. 4.2). Walking velocity (both linear and angular) and direction are high level commands ("user input" in Fig. 4.1) given by a robot operator (e.g. by using a gamepad).

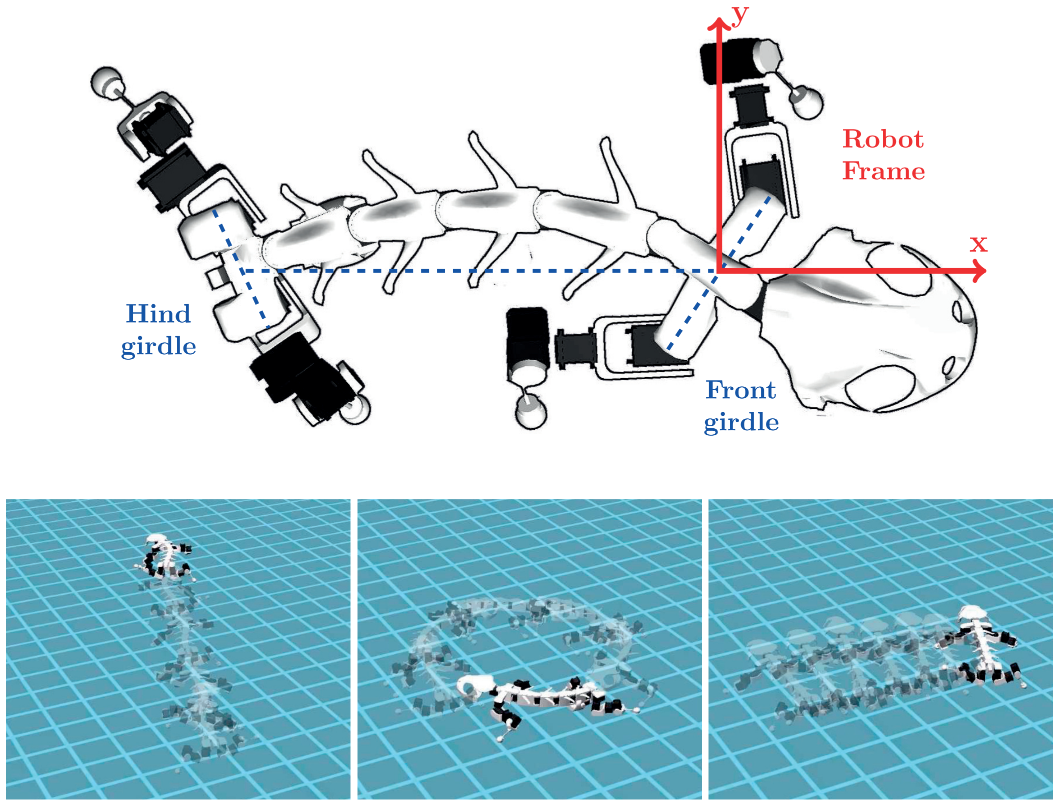


Figure 4.2 – (Top) The Pleurobot's reference frame. The x-axis passes through both front and hind girdle while the origin of the reference frame is attached to the front girdle. The z-axis points towards the reader. (Bottom) Maneuverability capabilities of the Pleurobot: forward/backward walking, turning and sidestepping. All three stages are used in the both simulation and hardware experiments.

4.4 Experiments

To evaluate the performance of the proposed controller, we carried out a series of experiments on both the simulated and real robot[3]. To show universal capabilities of the controller, we designed four different walking gaits whose characteristics, including gait diagrams [78], are shown in Table 4.1. Gait 1 is a very stable gait which Pleurobot does not have problems executing, even without controlling the CoM. Gait 2 is higher and narrower than gait 1 so balance becomes more important. Gait 3 has an unusual leg phase offsets which is not something we commonly use on Pleurobot. Benefits of the spine oscillations in this case are not so visible therefore the spine is kept straight for gait 3. Gait 4 has a lower duty factor which does not satisfy our starting assumption of always having at least three legs on the ground. However, we wanted to see how does the proposed controller handle more dynamic gaits. Gaits 1 and 2 can be classified as

---

[3] Video of the simulation and hardware experiments can be fount at: https://biorob2.epfl.ch/video/309

diagonal sequence walk, gait 3 as lateral sequence walk and gait 4 as walking trot [78]. All the experiments are performed twice, with and without controlling the CoM (to the latter we will refer as a basic controller).

The controller parameters, mostly MPC related are shown in Table 4.2. Weight matrices contained in the cost function 4.2a as well as constraints imposed on the CoM_{ref} dynamics were iteratively tuned by using the simulation across multiple gaits (not specifically for the ones used in the experiments). For the state reference (in the cost function 4.2a) we used constant value $r = \begin{bmatrix}-21.2 & 0 & 0 & 0 & 0 & 0\end{bmatrix}^T$. This means the MPC will try to position CoM_{ref} to the location of the estimated CoM (on the spine, 21.2 cm behind the front girdle) when the robot stands still, with straight spine and all four legs on the ground. The same parameters were used for the hardware experiments. Our choice of the prediction horizon length N was imposed by the time needed to solve the optimization problem 4.2. For longer prediction horizons, the solving time on the robot's on-board computer might take more than discretization time of the CoM_{ref} dynamics dt_2.

Table 4.1 – Gait parameters used in the experiments.

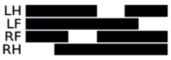
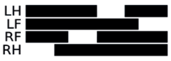
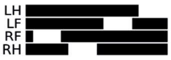
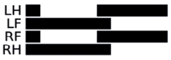
Gait	Duty factor	Leg phase offsets FL/FR/HL/HR	Stance L/W/H [cm]	Gait diagram
1	0.8	0/0.5/0.3/0.8	30/16/18	
2	0.8	0/0.5/0.3/0.8	30/14/24	
3	0.8	0.25/0.75/0/0.5	26/16/20	
4	0.6	0/0.5/0.5/0	30/20/18	

Table 4.2 – Controller parameters used in the experiments.

Parameter	Value
$\mathbf{Q}$	diag(5000, 200, 20, 5000, 200, 20)
$\mathbf{Q}_N$	diag(5000, 200, 20, 5000, 200, 20)
$\mathbf{R}$	diag(1,1)
$\mathbf{P}$	diag(10^6, 10^6, 10^6, 10^6)
s_m [m]	0.04
$x_{cr,\mathrm{min/max}}$ [m]	−0.3 / −0.15
$\dot{x}_{cr,\mathrm{min/max}}$ [m s^{-1}]	−0.15 / 0.15
$\ddot{x}_{cr,\mathrm{min/max}}$ [m s^{-2}]	−0.8 / 0.8
$\dddot{x}_{cr,\mathrm{min/max}}$ [m s^{-3}]	−100 / 100
$y_{cr,\mathrm{min/max}}$ [m]	−0.04 / 0.04
$\dot{y}_{cr,\mathrm{min/max}}$ [m s^{-1}]	−0.2 / 0.2
$\ddot{y}_{cr,\mathrm{min/max}}$ [m s^{-2}]	−0.8 / 0.8
$\dddot{y}_{cr,\mathrm{min/max}}$ [m s^{-3}]	−100 / 100
N	7
dt_1 [m s^{-1}]	10
dt_2 [m s^{-1}]	50

4.4.1 Simulation

Simulation experiments are made with a Pleurobot model in Webots™ robotics simulator. Each of the gaits is executed at two different frequencies: 0.25 Hz and 0.5 Hz. The frequency of 0.5 Hz corresponds to the fast walking of the real salamander animal after being dynamically scaled to the size of Pleurobot [98].

In the first set of experiments we let the robot walk for 180 s. For $0\,\mathrm{s} \leq t < 60\,\mathrm{s}$ the robot walks straight. For $60\,\mathrm{s} \leq t < 120\,\mathrm{s}$ the robot is commanded to keep turning right with a turning radius of 0.5 m. For $120\,\mathrm{s} \leq t < 180\,\mathrm{s}$ the robot is commanded to walk sideways to the right. The second set of experiments is meant to test the controller during a constant change of high level inputs which define turning radius and walking direction. We let the robot walk for 120 s. For $0\,\mathrm{s} \leq t < 60\,\mathrm{s}$ the turning command is constantly oscillating with a sine law between maximum turning to the left to maximum turning to the right. For $60\,\mathrm{s} \leq t < 120\,\mathrm{s}$ robot's locomotion direction is changing at the constant rate between 0° and 360° counter clockwise. During that period, the heading of the robot is kept constant (no spine-assisted turning) so the resulting motion is a combination of forward-backward walking and sidestepping. The frequency of turning command and walking direction oscillations are 10 times lower than the walking frequency (0.025 Hz and

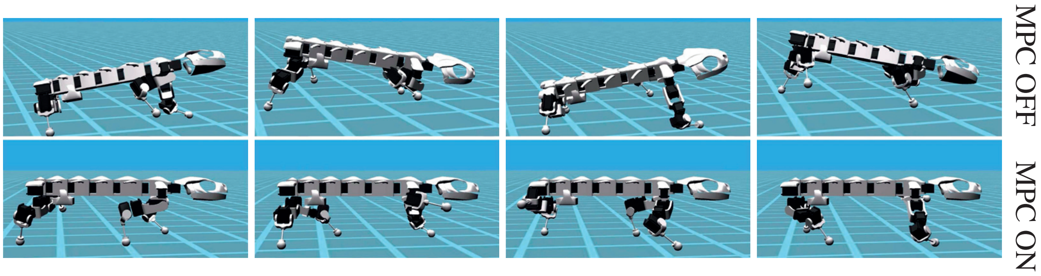


Figure 4.3 – Snapshots showing simulated robot using gait 2 (walking forward) at frequency 0.025 Hz without (up) and with (bottom) controlling the CoM. The snapshots show one walking cycle, with a time difference of 1 s between each of them.

0.05 Hz respectively). In the end, each set of experiments consists of 16 runs: four different gaits at two different frequencies with and without controlling the CoM.

To evaluate the controller's performance, we use several different metrics. First, we check how much the projection of estimated CoM leaves the support polygon:

$$J_{\text{spoly}} = \frac{1}{K}\sum_{k=1}^{K} \text{dist}(\text{CoM}_{\text{est},k}, \text{spoly}_k)^+, \tag{4.4}$$

where K is the number of samples and $\text{dist}(\text{CoM}_{\text{est},k}, \text{spoly}_k)^+$ measures the distance of CoM_{est} (while being outside) from the nearest edge of the support polygon at moment k. This metric reflects a violation of soft constraints in our MPC cost function. We chose it to see how well it translates into the actual improvement of the robot's stability.

To quantify the stability, we measure roll ϕ_{roll} and pitch ϕ_{pitch} angles of the robot. The measurements are performed by an IMU placed on the robot's front girdle (in both simulation and on the real robot). Due to the spine movements, we transform those measurements to the robot's coordinate frame. Since all the gaits have the constant height of the stance trajectories, in ideal case the robot's posture should be completely horizontal (roll and pitch angles are zero). Any tilting will be the result of losing balance and tipping towards the legs in swing phase. The metric is calculated as follows:

$$J_{\text{rp}} = \frac{1}{K}\sum_{k=1}^{K}\Big(|\phi_{\text{roll},k}| + |\phi_{\text{pitch},k}|\Big). \tag{4.5}$$

Since the maximum height of a swing leg is lower than the overall body height, we assume that in case of losing stability and tipping over, the foot of a swing leg will hit the ground. Therefore we define our final metric as an average force experienced by the feet force sensors during the swing phase of the corresponding leg:

$$J_{\text{force}} = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{4} ||F_{i,k,\text{swing}}||, \tag{4.6}$$

Where $F_{i,k,\text{swing}} \in \mathbb{R}^3$ is a 3D force measurement of i-th leg during its swing phase (in stance it is set to 0) and $||\cdot||$ is Euclidean norm. This metric does not only indicate ground contact of the swing leg, but also its intensity: a higher value indicates a higher impact force.

Results of the first set of experiments are shown in Fig. 4.4 and 4.3. The percentage above the blue bars indicate the performance difference of the CoM controller compared to the basic controller. Overall, the CoM controller scores well across all three metrics. The only exception is J_{rp} score for gait 4 at the lower frequency. J_{spoly} metrics shows that the support polygon violations are almost entirely eliminated for the gaits 1-3. However, that does not mean the robot is always stable during the run due to our estimation of ZMP with CoM. This is reflected in J_{force} metric which still shows some ground contacts of the swing legs although the amount of contact is decreased for at least 60% compared to the basic controller.

Results of the second set of experiments are shown in Fig. 4.5. Same as before, the overall performance is improved across all the metrics and gaits. An exception is the turning phase for the gait 2, where the CoM controller scores worse in the J_{rp} metric.

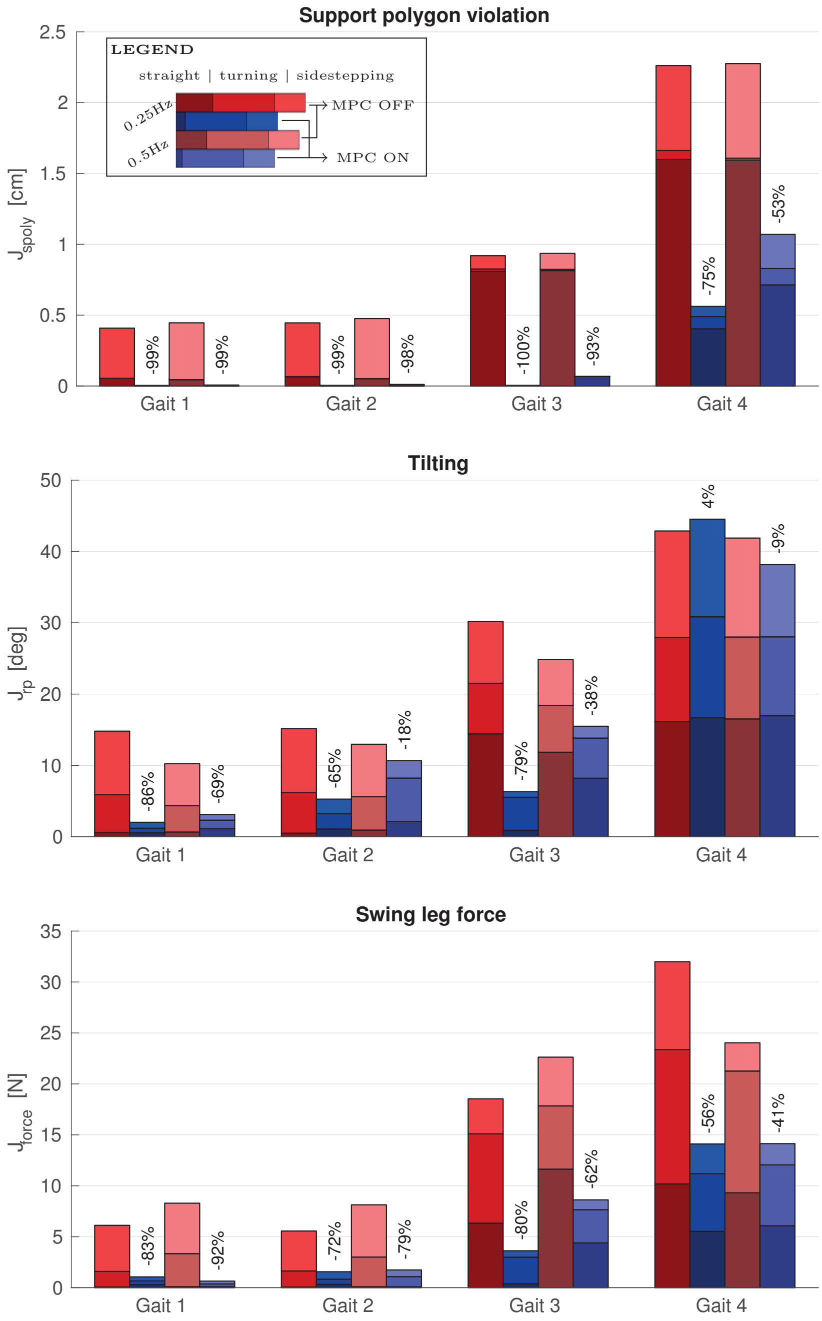


Figure 4.4 – The results of the simulation experiments (first set). The different shades in each bar indicate different stages in the walking sequence (forward, turning, sidestepping). A percentage above the blue bars (MPC enabled) indicates how much the metric cost is reduced compared to the first red bar on the left (MPC disabled).

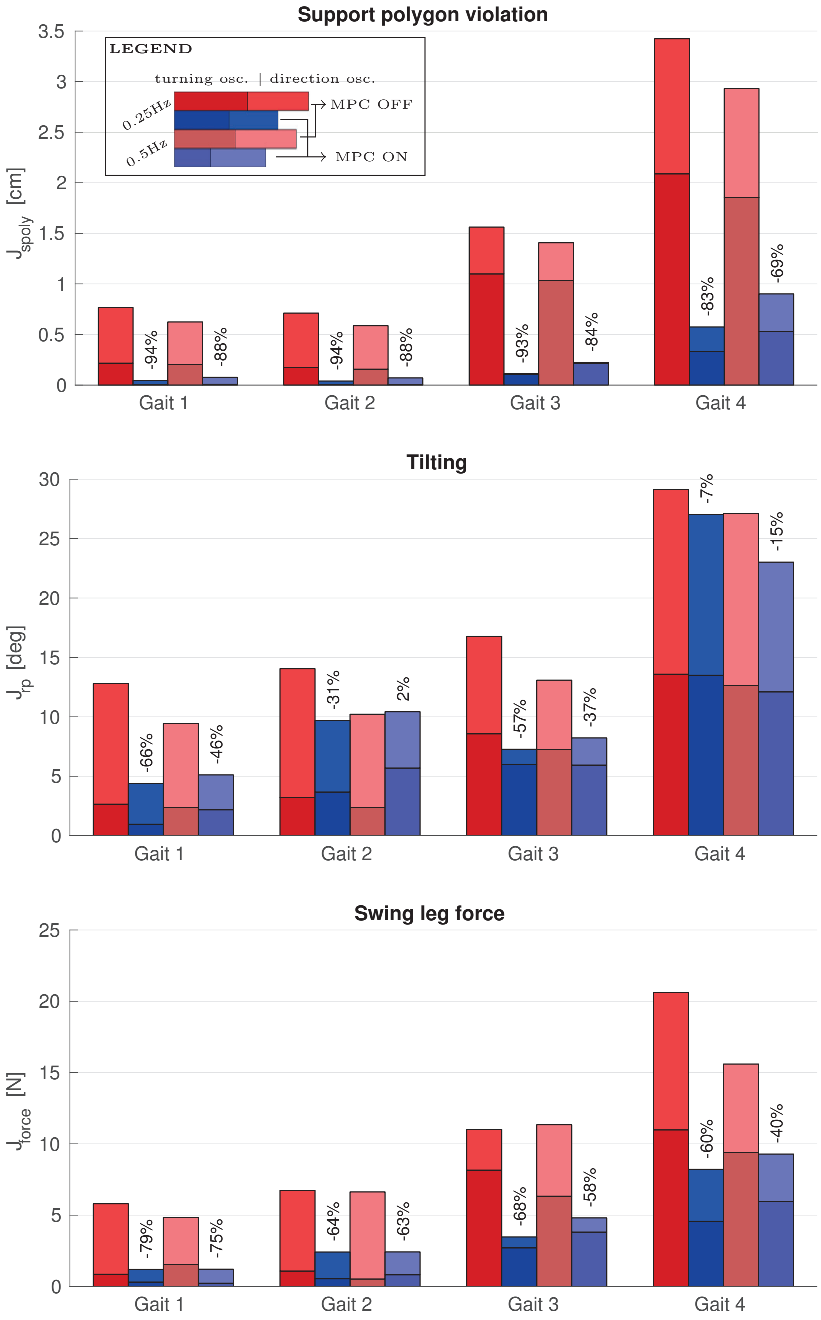


Figure 4.5 – The results of the simulation experiments (second set). The different shades in each bar indicate different stages in the walking sequence (slalom turning, rotating walking direction). A percentage above the blue bars (MPC enabled) indicates how much the metric cost is reduced compared to the first red bar on the left (MPC disabled).

4.4.2 Hardware experiments

To confirm the results of simulation experiments, we repeated a subset of them on the real Pleurobot. The experiments are similar to the first set of simulation experiments. However, due to the space and hardware restrictions, each of the segments (straight walking, turning and sidestepping) was run independently, until the robot made at least 5 steps. Furthermore, we tested all the gaits with the lower walking frequency of 0.25 Hz. The front girdle roll and pitch angles were recorded with a VectorNav VN-100 IMU. To record the swing feet forces we used Optoforce 3D force sensor. All the computation was done with the on-board computer - Odroid-XU4. The controller parameters were exactly the same as in the simulation.

The results are shown in Fig. 4.6. The dynamical behavior of the real robot was slightly different than in the simulation. This is mostly due to flexing across the robot's body and bouncing caused by rubberized force sensors that acted as feet. Neither of those effects were modeled in the simulation. However, the results show a similar trend as the simulation. The CoM controller improves stability, specially for the sidestepping locomotion. It is important to mention that the robot repeatedly failed to turn while using the gait 2 in both cases. Therefore, that segment is excluded from the results.

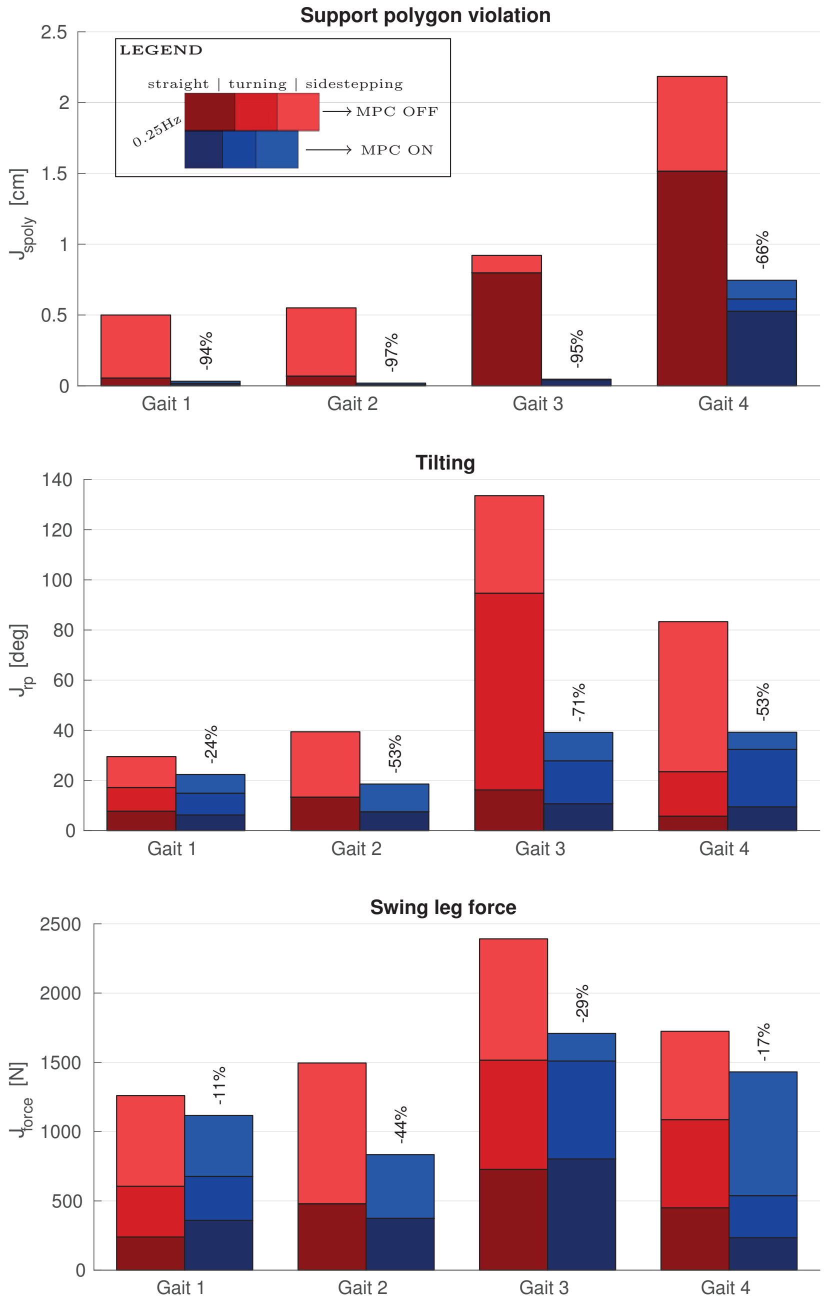


Figure 4.6 – The results of the real robot experiments. The different shades in each bar indicate different stages in the walking sequence (forward, turning, sidestepping). A percentage above the blue bars (MPC enabled) indicates how much the metric cost is reduced compared to the first red bar on the left (MPC disabled). The results of the turning stage for gait 2 are excluded since the robot kept tipping over in both cases (MPC enabled and disabled).

4.5 Discussion

We presented a center of mass control framework for position controlled quadruped robots using a static walking gait. The framework is built around MPC acting on a dynamical system representing a desired dynamics of the CoM. The output of the system is then used as reference for the estimated CoM, assuming the reference dynamics is slow enough for the robot to follow it. Such assumption holds true since the user has the power to model the dynamics of the reference. The real strength of using MPC is its capability of taking into account future states of the system. More precisely, we used predictions of the support polygon movements, which resulted in adaptations of CoM reference to the events like leg liftoff, before they occur. Knowledge of support polygons is incorporated in the MPC scheme as a constraint acting on the position of the CoM reference. However, the said constraint is formulated as a soft constraint, meaning the optimization solver will not fail in case of inevitable support polygon violation.

The proposed scheme is tested in both simulation and on the real robot. Through the tests it showed improvements in the robot's stability - reduction in body tilting and less force exerted on the feet during a swing phase. To demonstrate the controller's independence of the used gait, we demonstrated it on four different gaits, each of them being used at different frequencies and locomotion stages (forward walking, turning and sidestepping).

4.5.1 Future steps

In this chapter we made several assumptions that can be alleviated in the future work. Integrating an IMU into the control loop will improve the support polygon and center of mass estimation. At the moment, we assume the robot's body is perfectly horizontal during locomotion and all the projections to the ground plane are done accordingly. Using an IMU to detect an actual body attitude, the accuracy of such projections can be greatly improved. Furthermore, by estimating an inclination of the ground plane, the controller will also be applicable for locomotion on slopes.

To account for more dynamic gaits, where an inertia of the robot's body is more significant so the approximation of ZMP with CoM does not hold any more, the current framework can be adapted to generate the ZMP trajectories, and from them the CoM reference [20].

Science Part II

5 Reverse Engineering the Locomotion of an Extinct Stem Amniote

Chapter 2 presented a bio-mimetic robot designed to replicate the gait kinematics of an animal. A locomotion controller for such robot that allows the user to precisely craft the kinematics of the robot gait, was described in Chapter 3. Here we combine these two studies to build and control a new robot that mimics the morphology and kinematic structure of an extinct animal. The motivation behind such endeavor is to use the robot as a tool to reconstruct and evaluate the animal's locomotion capabilities. The required data for building the robot and constraining its gait is obtained from a well preserved fossil and a set of fossilized trackways belonging to the same species. The outcomes of this study are: (i) insight into the most plausible gait of the extinct animal and (ii) an interdisciplinary methodology that could be applied to other fossils.

The material presented in this chapter is adapted from:

NYAKATURA, J. A., MELO, K., HORVAT, T., KARAKASILIOTIS, K., ALLEN, V. R., ANDIKFAR, A., ANDRADA, E., ARNOLD, P., LAUSTRÖER, J., HUTCHINSON, J. R., FISHER, M. S., AND IJSPEERT, A. J. Reverse engineering the locomotion of an extinct stem amniote. *Manuscript submitted to Nature.* (2018).

My contributions: Designed and performed the dynamic simulations, designed and carried out the experiments with the robot, designed and implemented the interactive online visualization tool, contributed to the writing.

5.1 Abstract

Understanding the locomotion of extinct vertebrates offers insight into their paleobiology and helps to conceptualize major transitions of vertebrate evolution [14, 122, 64, 149]. Reconstruction of a fossil's locomotor behavior, however, remains problematic, because of the limited information preserved and lack of one-to-one correspondence between form and function [110, 116]. The evolution of advanced, i.e., more erect, balanced and mechanical power saving locomotion on land reflects the terrestrialization of tetrapods and was previously linked to the diversification of amniote lineages [182]. No quantitative and reproducible approaches to reconstruct locomotor characteristics of stem amniote fossils are available, and generally methods suffer from overreliance on anatomical features, ambiguous locomotor information preserved in ichnofossils, or unspecific modeling of locomotor dynamics. *Orobates pabsti*, of which fossil trackways have also been preserved [188], was a stem amniote close to the origin of the crown-group. Here we present an integrative methodological framework to reconstruct gaits of Orobates that is constrained by quantified metrics for energetic efficiency, balance, and precision of matching fossil tracks. Our framework uses in vivo assessment of locomotor mechanics in four extant species to guide an anatomically informed kinematic simulation as well as dynamic simulations and bio-informed robotics to filter the parameter space for plausible gaits (Fig. 5.1). The analysis indicates that Orobates exhibited advanced locomotion pointing to greater agility than assumed for earlier tetrapods. This also suggests that advanced terrestrial locomotion preceded the diversification of crown group amniotes. Readers can interactively explore filters constraining our simulations on an accompanying website also supported by open datasets. Our methodological approach can serve as an exemplar for similar research questions that address locomotion of key taxa to gain insight into evolutionary transitions. Importantly, our quantitative and dynamic reconstruction can be extended and revised according to future methodological advances.

Figure 5.1 – (Next page.) Flow chart of the basic steps of analysis. **DATA ACQUISITION**: digitizing holotype specimen of focus fossil, Orobates and trackways; conducting x-ray motion analysis and measuring ground reaction forces of extant animals. **SIMULATION**: kinematic simulation of a digital marionette of Orobates in Maya; dynamic simulation of OroBOT within the Webots environment; using constraints (anatomy of the holotype specimen, Orobates' trackway parameters, mechanical principles of sprawling locomotion of extant animals). **GAIT SOLUTIONS**: Combining kinematic and dynamic filters to exclude unlikely gaits to narrow down the parameter space. **ROBOTICS**: Demonstrating validity of gait solutions. 1) holotype specimen *Orobates pabsti* (MNG 10181); 2) x-ray motion analysis of an iguana; 3) digital marionette of Orobates, 4) dynamic OroBOT simulation, 5) OroBOT locomoting on a treadmill. Weights given to filters can be manipulated and representative videos of the animals, all simulated permutations of the dynamic OroBOT and the kinematic Orobates simulations can be explored using the interactive online visualisation tool[1] accompanying this work.

---

[1] https://biorob2.epfl.ch/users/thorvat/OrobotProjectWeb/

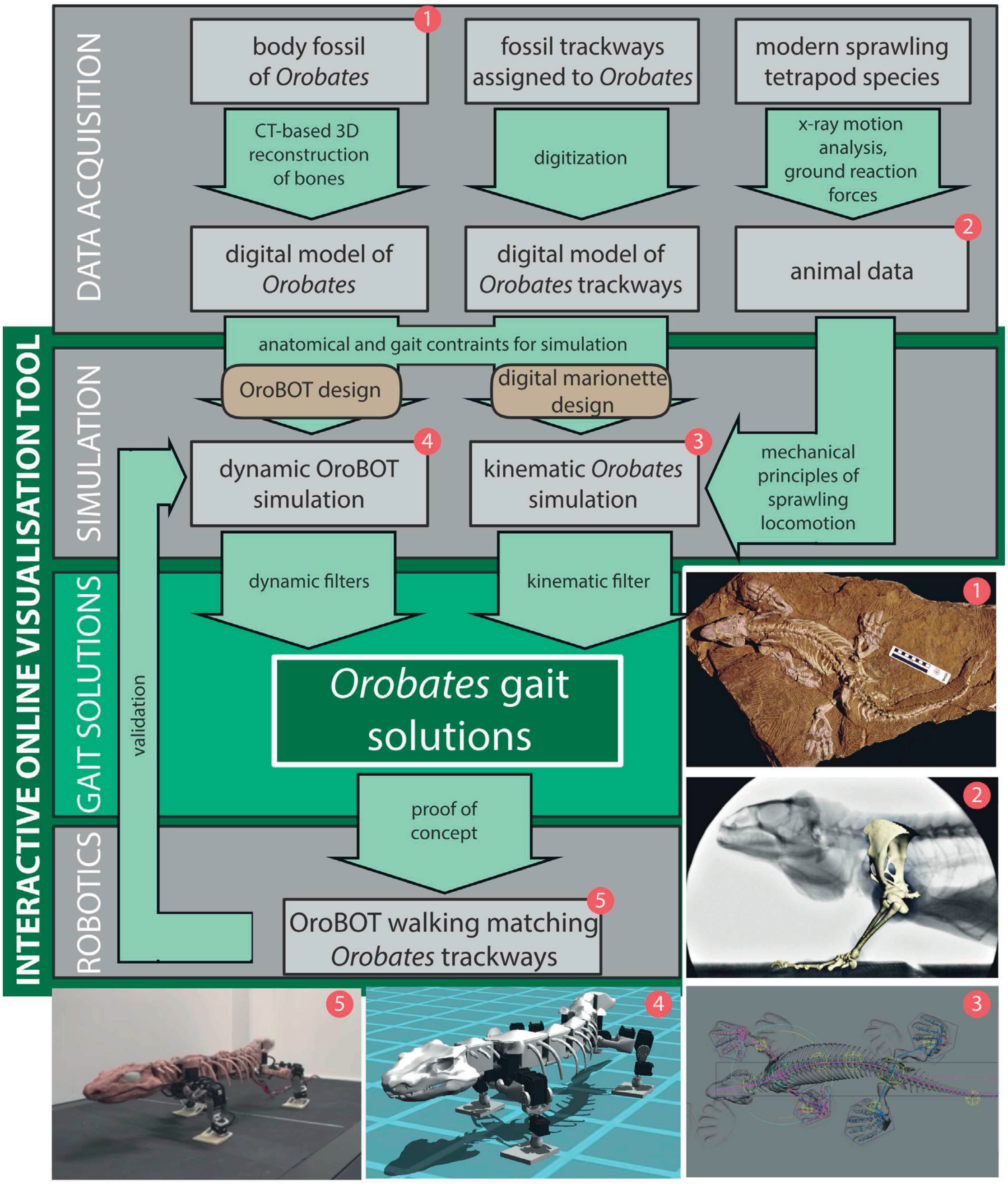

DATA ACQUISITION
body fossil of *Orobates*
1
fossil trackways assigned to *Orobates*
modern sprawling tetrapod species
CT-based 3D reconstruction of bones
digitization
x-ray motion analysis, ground reaction forces
digital model of *Orobates*
digital model of *Orobates* trackways
animal data
2
INTERACTIVE ONLINE VISUALISATION TOOL
SIMULATION
anatomical and gait contraints for simulation
OroBOT design
digital marionette design
dynamic OroBOT simulation
4
kinematic *Orobates* simulation
3
mechanical principles of sprawling locomotion
GAIT SOLUTIONS
validation
dynamic filters
kinematic filter
Orobates gait solutions
ROBOTICS
proof of concept
OroBOT walking matching *Orobates* trackways
5
1
2
5
4
3

Figure 5.1 – Flow chart of the basic steps of Orobates locomotion analysis.

5.2 Introduction

Reconstructing the locomotion of key vertebrate fossils is critical for inferring major transitions in vertebrate evolution. Often, locomotor behavior is reflected in the musculoskeletal anatomy of vertebrates, however, reconstruction attempts solely focusing on an anatomical analysis of fossil remains have been criticized to suffer from joint ranges of motion that are much larger than what is used during locomotion and redundancy (excess degrees of freedom) in appendages [137, 62, 57]. Additionally, these studies tend to neglect the biomechanics of the entire organism [175] and analyses of extant species demonstrate that diverse anatomical conditions can potentially enable similar function. Thus there is no one-to-one correspondence between form and function [110, 62]. In contrast, reconstruction attempts relying entirely on mechanical modeling and engineering are limited, because they often neglect anatomical detail. Finally, reconstruction attempts relying solely on interpretation of fossil trackways suffer from uncertainty of different movements or gaits that might produce nearly identical trackways [180]. Here we propose an integrative approach that makes use of the advantages of these different strategies to reconstruct locomotion of tetrapod fossils, focusing on an attempt to infer the potential gaits of the stem amniote *Orobates pabsti* (Fig. 5.1). Orobates, a basal diadectid from the lower Permian, is an ideal candidate fossil for the reconstruction of locomotor behavior, because of its complete and articulated preservation [10] and because also fossil trackways have been assigned to this species, constituting the oldest tetrapod track-trackmaker association [188]. The an-amniote to amniote transition documented by stem amniotes such as Orobates involved completing the tetrapod transition from water to land, because their eggs were equipped with semi-permeable shells and nutrients to allow the development of terrestrial hatchlings, thereby avoiding an aquatic larval stage critical for terrestrialization [162].

5.3 Results

The anatomically precise digitized fossil [139](Fig. 5.2) was used to design three types of models: a digital marionette used for kinematic simulation, a model for dynamic (i.e. physics-based) simulation, and a physical model (OroBOT). A fossil trackway assigned to Orobates was idealized and used as a hard constraint in the kinematic simulation and as a precision metric in the dynamic OroBOT simulation. The purpose of the kinematic Orobates simulation was to identify anatomically plausible kinematic parameter combinations. The purpose of the dynamic OroBOT simulation was to quantify the physics of locomotion, and to assess physical aspects of gaits such as mechanical power, the ability to walk without excessive tilting; and the precision of matching the fossil trackway. The purpose of OroBOT was to validate the results of the dynamic simulation under real-world physical conditions. In our acquisition of animal data and in our simulations we focused on body lift and those kinematic aspects that have been demonstrated to be the most important contributors to overall progression during sprawling tetrapod locomotion (Fig. 5.5): lateral bending of the vertebral column, long-axis rotation (LAR) and retraction in the proximal limb joints (shoulder and hip) [46, 5].

The mechanics of locomotion and track-making of four metamorphosed Mexican salamanders, two blue-tongued skinks, two green iguanas, and two spectacled caimans (Appendix B.1) were analyzed in this study. Species sampling accounted for the phylogenetic bracket of the an-amniote to amniote transition [196, 111], represented different major taxa within amniotes that usually exhibit sprawling locomotion (squamates and archosaurs), species with very different ecologies (e.g., a desert dwelling and a arboreal species), and diverse gross morphologies such as different limb length to snout vent length ratios. It was not our goal to find an analogue for Orobates. Instead, we aimed to identify principles of sprawling tetrapod locomotion that apply to most if not all sprawling taxa. Given the limited number of species that can practically be studied in detailed analyses of locomotor mechanics, our species sampling covered a reasonable portion of the mechanical disparity exhibited in the 6000+ tetrapod species using quadrupedal sprawling locomotion.

With the freedom to choose their preferred gait and speed, extant species exhibited walking trots and lateral sequence gaits (limb phase (LP): 0.42 ± 0.04 (mean ± standard deviation); duty factor (D): 0.67 ± 0.1; stride frequency: 0.83 ± 0.43). Comparative x-ray motion analysis with simultaneous ground reaction force (GRF) measurement revealed that Salamanders and skinks (both limb pairs) as well as iguanas (only hindlimbs) exhibited relatively less body lift (i.e., a hyper-sprawled limb posture) while showing more pronounced LAR and less retraction of the proximal skeletal element (Fig. 5.3). In contrast, more erect limb postures in caimans (both limb pairs) and iguanas (only forelimbs) displayed more retraction while showing less LAR of the proximal skeletal element. Forelimb vertical GRF patterns were very similar in our sample of extant species, with a peak force of almost 0.5 body weight units (BWUs; 0.46 ± 0.02) and occurrence of peak force at $\sim 2/3$ of contact time ($62.5\% \pm 6.45\%$) indicating that dynamic similarity was maintained in these aspects. Hindlimb vertical GRF profiles were less similar, but resembled each other in the timing of peak force at $\sim 1/3$ of contact time ($32.5\% \pm 6.45\%$). Based on these observations of key mechanical aspects of sprawling locomotion in our diverse sample, we conclude that in sprawling locomotion the amount of LAR in the shoulder and hip is inversely related to retraction; and that this relationship depends on body lift. Despite the differences in the kinematic patterns between the hyper-sprawled and more erect limbs, the kinetics of sprawling locomotion remain similar in terms of the timing of peak vertical GRF. Trackway parameters of Orobates (blue in Fig. 5.3) suggest intermediate kinematics, but given the observed principles in extant species similarity in kinetics can be inferred for the fossil. To facilitate the linkage of extant animal data with both simulations (kinematic *Orobates*; and dynamic OroBOT simulation) (Fig. 5.4), our simulations also focused on body lift, spine bending and LAR . Both simulations were used to systematically vary each parameter. We varied all parameters from "none" to "exaggerated" to cover a much larger parameter space than was exhibited in the extant species (with validation of this workflow using the caiman: see methods). To rule out anatomically implausible kinematics, 200 permutations of parameter combinations (100 for forelimbs and hindlimbs each) were evaluated in terms of the occurrence of bone collisions in the proximal limb joints and disarticulations of the wrist or ankle during a limb's ground contact (Fig. 5.5). For each bone, collisions were scored on four levels with a maximum score for perfect plausibility (no joint

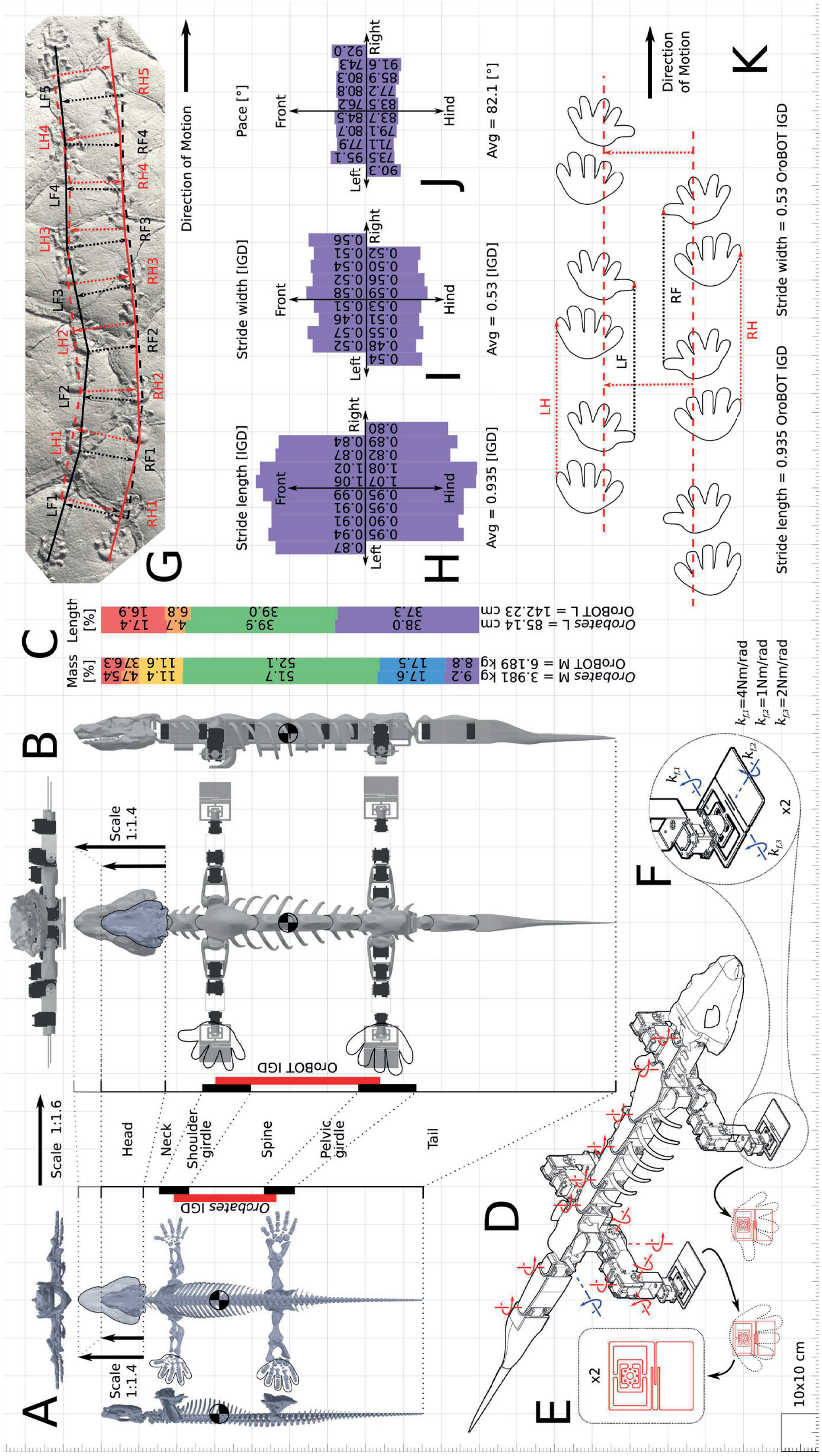


Figure 5.2 – Fossil, robot, and trackway detailed description.

Figure 5.2 – (Previous page.) (**A**) Orobates fossil 3D views in a grid of 10 × 10 cm. Position of center of mass (from previous publication [139]) and lengths of different segments including inter-girdle distance (IGD) in red bar. (**B**) Scaled ($s = 1.6$) robotic reconstruction of Orobates fossil called OroBOT. 3D views in a grid of 10 × 10 cm. Position of center of mass, segment lengths and scaled IGD in red bar. Details of head secondary scale for housing the main processing unit volume. (**C**) Mass and length distributions and comparison between different segments of fossil used for the robot design. Fossil masses and lengths percentages matching in the robotic replica. (**D**) Isometric view of OroBOT robot specifying the joints location. Active joints in red and passive joints in blue. (**E**) Passive compliant foot pattern and dimensions. Scale from fossil tracks and physical implementation in the robot. (**F**) Detail of the passive compliant foot with values of stiffness and passive bending axes. (**G**) Original Orobates - associated trackway in a 10 × 10 cm grid. (**H**) Detail of stride lengths, (**I**) stride widths and (**J**) pace measurements for front, hind, left and right feet. (**K**) Idealized trackway for OroBOT. Stride length and stride width correspond to averaged values of the data in (**H**), (**I**), and (**J**).

disarticulations and bone collisions). Using a combined score for both limb pairs (eight levels), we found a domain of anatomically plausible parameter combinations within the parameter space marked by little to moderate LAR (and hence pronounced retraction), intermediate to high body lift, and moderate lateral spine bending at the girdles (Fig. 5.4). Videos of all parameter combinations can be explored on the accompanying interactive website.

OroBOT made use of off-the-shelf electromechanical equipment such as actuators and sensors and was built using 3D printing [89, 66]. Control and design was based on a previous biomimetic platform, Pleurobot, which successfully replicated kinematics and dynamics of a walking salamander [98]. OroBOT closely mimicked the fossil in terms of anatomy, mass distribution of body segments, and position of center of mass (Fig. 5.2). The dynamic simulation of the physical model OroBOT was used to systematically test 512 (8 × 8 × 8) combinations of kinematic parameters analogous to the kinematic simulation. Three biologically meaningful metrics were obtained and subsequently used to exclude unlikely gaits from the parameter space to identify dynamically plausible gaits: 1) Efficiency (in terms of mechanical power), which was measured as the reciprocal of positive torque times velocity summed over the 28 actuated joints for one stride of OroBOT. Animals are expected to choose gaits that minimize energy expenditure to produce torque in their joints [57]. While passive mechanisms to regain energy were probably already present in early tetrapods [160], these cannot be replicated in the position-controlled servos of OroBOT and were thus not considered. 2) Balance, which was quantified as the reciprocal of the body's roll and pitch rate. Animals are expected to minimize rapid tilting of the body to not compromise optical and vestibular perception [44]. High scores in balance correspond to gaits that have low rates of tilting. 3) Precision, measured as the accuracy of the robot's foot placement within the fossil trackways. All parameter combinations and metric thresholds can be explored on the website. Here we present an exemplar analysis with all metrics' cut-off thresholds set to to 50%, which excludes the 50% lowest-ranked solutions of each metric. The remaining parameter combinations of the three individual metrics were then combined to create a combined dynamic score, which assumes the highest value for parameter combinations that passed the

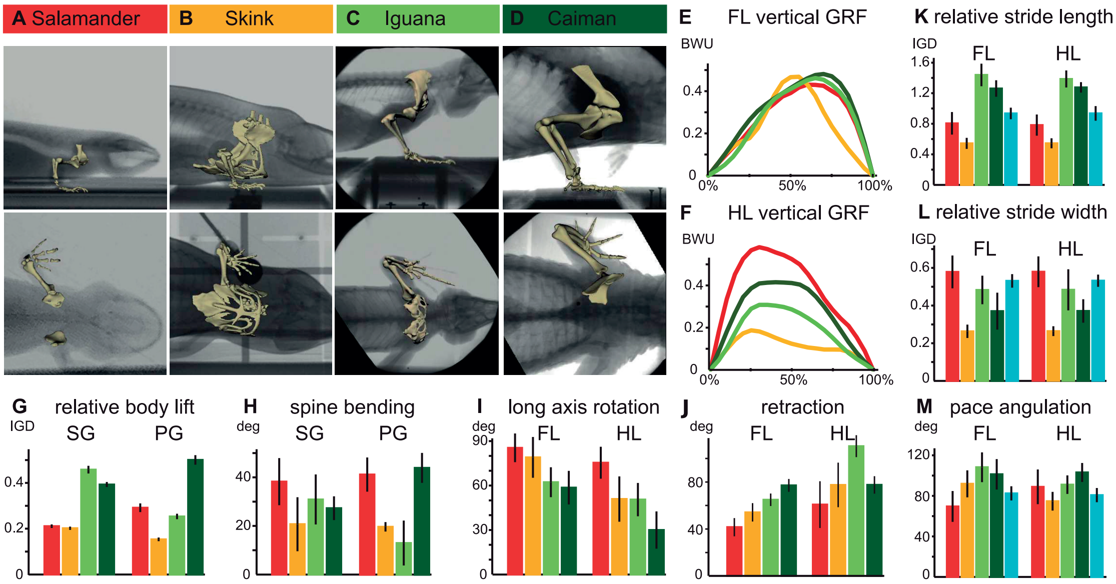


Figure 5.3 – Extant animal data. **A**-**D**: Screenshots of x-ray motion analysis from lateral (upper row) and ventro-dorsal (lower row) projections. Color code applies to all other plots (blue represents available data for Orobates). Note that data are normalized to inter-girdle distance (IGD) or body weight units (BWU) to allow comparison. **E**,**F**: vertical ground reaction force (GRF) profiles. **G**-**J**: Comparison of mean data (± standard deviation) for key kinematic aspects of sprawling locomotion for the shoulder girdle (SG) or pelvic girdle (PG) and forelimb (FL) and hindlimb (HL), respectively (Appendix B.1). **K**-**M**: comparison of mean trackway data (± standard deviation) plotted together with available data for Orobates (blue).

cut-off threshold of all individual metrics. The final step to arrive at plausible gait solutions for Orobates combined these dynamically plausible gaits with the anatomically plausible gaits. With the suggested exclusion settings, the overall parameter space was narrowed down to one parameter combination that achieved the highest score (Fig. 5.4). That most plausible solution corresponds to a gait with moderate LAR and spine bending combined with considerable body lift (similar to caiman; see website). The highlighted plausible gait solution may not be the only parameter combination yielding the maximum high score, depending on the weight given to the individual metrics, but is representative of a domain within the parameter space with a clustering of gait solutions indicative of more erect (adducted) limb posture (the reader is advised to try, e.g., a cut-off threshold of 30% or 70% for each metric on the website). As a proof of concept, the physical robot using parameter combinations within this domain is capable of reproducing trackway parameters associated with Orobates. For the robot, 15 different gaits[2] were tested and produced good matches to the dynamic simulation, i.e., the gait solutions could effectively be produced by the robot with foot placements very close to the idealized fossil trackway.

---

[2] The videos can be fount at https://biorob2.epfl.ch/users/thorvat/OrobotProjectWeb/

5.4 Discussion

A more erect locomotion of Orobates is consistent with previous inferences based solely on fossil trackways [188] and was found here to be balanced and mechanical power saving. More erect limb postures are today found in sprawling species living in open habitats and are linked with greater capacity for speed and agility [58], reduced torsional stresses occurring at limb long bone midshafts [14, 161, 15], and reduced expended power to accelerate the body in the direction of travel [164]. Orobates locomotion was more agile than assumed for earlier tetrapods [101]. It is parsimonious to assume the presence of these advanced terrestrial locomotor properties in the last common ancestor of diadectids and amniotes, i.e. within the amniote stem lineage and preceding the subsequent rapid radiation of crown group amniotes in contrast to previous suggestions [182, 162]. Future studies may critically re-evaluate the proposed metrics here and refine our results. Similar integrative approaches may be adopted for comparable research questions concerning major transitions in vertebrate evolution.

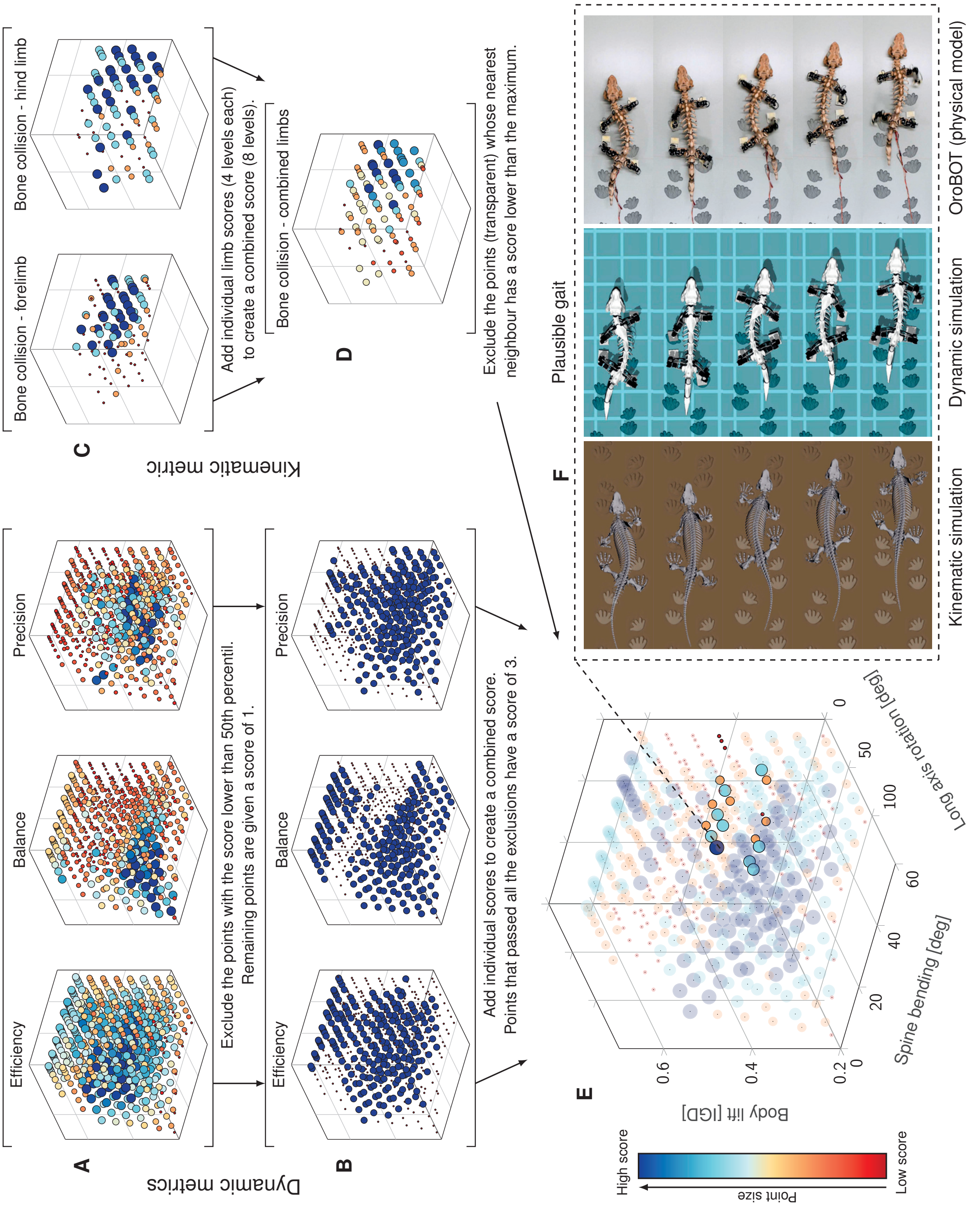

A
Dynamic metrics
Efficiency
Balance
Precision
Exclude the points with the score lower than 50th percentil.
Remaining points are given a score of 1.
B
Efficiency
Balance
Precision
Add individual scores to create a combined score.
Points that passed all the exclusions have a score of 3.
C
Kinematic metric
Bone collision - forelimb
Bone collision - hind limb
Add individual limb scores (4 levels each)
to create a combined score (8 levels).
D
Bone collision - combined limbs
Exclude the points (transparent) whose nearest
neighbour has a score lower than the maximum.
E
Body lift [IGD]
0.6
0.4
0.2
0
20
40
60
Spine bending [deg]
0
50
100
Long axis rotation [deg]
High score
Low score
Point size
F
Plausible gait
Kinematic simulation
Dynamic simulation
OroBOT (physical model)

Figure 5.4 – Procedure of identifying plausible OroBOT gaits.

Figure 5.4 – (Previous page.) Procedure of identifying plausible OroBOT gaits. (**A**) Individual metric scores in the gait space. (**B**) Binary threshold (50th percentile) applied to the individual scores. A and B comprise the dynamic metrics set. (**C**) Forelimb and hind limb bone collision scores (4 levels). (**D**) Summed bone collision score (8 levels). C and D comprise the kinematic metric. (**E**) Summed scores of B. The points whose nearest neighbour in D has a score lower than the maximum (8) are excluded (transparent), providing a region for the most plausible gaits. (**F**) Snapshots[3] of a gait with the highest score within the plausible region.

5.5 Methods

5.5.1 Comparative motion analysis of extant species

All procedures involving live animals adhered to animal welfare regulations and were authorized by the authorities in Thuringia, Germany (registration number: 02-008/11). All specimens were adult and did not show any abnormalities. The experimental design combined biplanar, high-speed x-ray videography with the simultaneous measurement of single limb ground reaction forces (GRFs) in order to study locomotor mechanics. Additionally, trackway production was quantified during locomotion on soft clay (i.e., in a similar situation as preserved fossil trackways assigned to Orobates [188]). Kinematic analysis was conducted using the fully digital, biplane, high-speed x-ray facility of the Institut für Zoologie und Evolutionsforschung at the Friedrich-Schiller-Universität in Jena, Germany. The specifics of the facility have been described in previous publications [51, 141]. Synchronized x-ray recordings from ventral and lateral projections of locomotor trials in which the animals could choose the speed of progression were captured using 38 cm diameter image intensifiers at a resolution of 1536 x 1024 pixels. Locomotion, if not exhibited spontaneously, was motivated by gentle touches with a stick on the tail resulting in a series of several consecutive strides. For the smaller species (salamanders and skinks) we used an instrumented trackway (1.0 × 0.3 m) with two custom-built radiolucent force plates which were placed side by side in the field of view of the two x-ray projections (for details see [4, 140]). To study locomotion of the larger species (iguanas and caimans), a larger trackway was built (3.0 × 0.75 m) with a radiolucent board mounted on two Kistler force plates arranged to the left and right of the trackway to allow ventral x-ray projection. Both the small and the large trackway were constrained by an acrylic glass enclosure. Recording frame rate varied from 300 frames per second (fps) for salamanders to 500 fps (other species).

For x-ray motion analysis, we used x-ray reconstruction of moving morphology (XROMM [17]). Specifically, we used the non-invasive variant of XROMM (no need to implant metal markers into bones of interest) termed "scientific rotoscoping" [61]. The general procedure for SR was identical for the analysis of all sampled subjects and was described in detail for the blue-tongued skink in a previous publication [140]. It will therefore only be summarized here. In SR a digital 3D model of the subject's skeletal elements of interest is manually positioned to match (i.e., overlay) these elements' x-ray shadows in both x-ray projections. For this, the 3D animation software Autodesk Maya® 2015 (Autodesk, San Rafael, Ca, USA) was used. The manual matching of 3D bone models with x-ray shadows was repeated for key frames (usually 15-20 keyframes each for the stance and swing phase) of the x-ray video and was then cubic-spine interpolated between keyframes to produce smooth movements that closely approximate the recorded skeletal kinematics. Raw x-ray videos were corrected for distortion prior to SR [61] and the distance/orientation of the x-ray image intensifiers in relation to the subject was determined by using a calibration object (0.2 × 0.12 × 0.12 m) placed within the biplanar field of view. We used the MATLAB calibration routine developed at Brown University, Providence,

---

[3] Accompanied video can be found at https://biorob2.epfl.ch/video/315

USA (www.xromm.org). To obtain bone models, CT scans were done either on a CT scanner belonging to the Friedrich-Schiller-University Hospital, Jena, Germany (skinks, iguanas, and caimans), or on a microCT scanner at the Fritz Lipmann Institute, Jena, Germany (salamander cadavers). Living skinks and iguanas were cooled to ca. 15 °C to reduce movement and caimans were scanned inside a dark transport box. Sedation was not necessary. Bone surface models were obtained using the segmentation editor in the Amira software package (VSG, Burlington, MA, USA). In SR, bone models are linked to form a hierarchical marionette [61]. Anatomical coordinate systems were implemented into the shoulder and pelvic girdle models of all species to measure the rotation of each girdle around a dorso-ventral axis as well as into the shoulder and hip to measure LAR, retraction, and abduction/adduction as described previously [140]. Movements in the shoulder and hip were measured relative to the respective girdles. We quantified the rotations (3 degrees of freedom) in the shoulder and hip joints relative to a reference pose. This pose was aligned to a right-handed global coordinate system placed in the trackway with positive x pointing in the direction of movement, positive y pointing to the animals' left, and positive z pointing upwards. All bone model coordinate systems were aligned to the axes of the global coordinate system (see [141]). To obtain anatomically meaningful data we used non-physiological fully extended reference poses (for both the fore- and the hindlimbs; with the humerus and femur pointing laterally [141, 140, 181, 142]. To avoid the singularity problem, the rotation order in each joint was set to have the largest expected movement as the dominant axis [181]. All data were exported into Microsoft Excel (v. 2014, Microsoft, Redmond, WA, USA) and each trial was normalized to the same duration (101 points). LAR/retraction of the shoulder and hip joints and spine bending was quantified as the range occurring during a limb's ground contact. Body lift (at midstance measured as distance from the spine at the shoulder or hip to the ground) and track parameters (stride length and width) were normalized to inter-girdle-distance (IGD) to allow comparison across all specimens (also including the kinematic and dynamic simulations).

We ran multiple linear regression with LAR (in shoulder and hip) as dependent variable and retraction (in shoulder and hip) and body lift as independent variables in Microsoft Excel using the "data analysis" add-in to statistically assess our qualitative observations (main text). *P* values were considered significant if below 0.05.

Custom-built force plates made from carbon fibre with 6 DOF force-torque transducers (ATI, Industrial Automatization, Apex, NC, USA) were used to minimize metal in the x-ray projection while measuring vertical GRFs in the smaller species (skinks and salamanders). For the larger species (caiman and iguana) vertical GRFs were collected using a radiotranslucent board mounted on two Kistler force plates which were placed at both sides of the trackway to allow ventro-dorsal x-ray projection. Both instrumented trackways were built with the force plates surface integrated flush with the trackway. Details of the design of the force plates used and GRF data analysis are documented in [4] (smaller species) and [197] (larger species). Vertical GRFs were collected at 500 Hz using customized software for LabView 2009 (NI USB-6229, National Instruments Germany GmbH, Munich, Germany). GRFs were normalized to body weight units (BWUs).

To document track production during walking while being recorded by a high-Hz camera, trackways were custom built for each species (see [36]). In the center of each trackway a bed of fine pottery clay (100 × 30 × 2 cm) and a measuring tape were placed within the field of view of the camera. The camera (Photron FASTCAM-X 1024PCI, Photron USA, Inc., San Diego, CA, USA) was positioned to capture a dorsal view. After each trial the imprints were photographed with a resolution of 4000 × 3000 pixels (Canon Powershot G9, Canon Inc., Tokyo, Japan). Stride length, stride width, and pace angulation were determined from photographs using ImageJ software (http://rsbweb.nih.gov./ij/). Only trials on clay with relatively equal moisture contents (using qualitative categories see [36]) were used for data analysis. Video data were used to determine locomotor speed (determined by measuring the displacement of the snout in direction of travel over a stride cycle).

5.5.2 Digital marionette design and kinematic simulation of Orobates

Fossil trackways were digitized and imported into Maya. Trackways were idealized so that two subsequent stride cycles were repeated infinitely (using an animated loop) and animated as if on a treadmill (Fig. 5.5). The digital holotype specimen of Orobates [139] was also imported into the scene. The digital marionette was scaled using the ’scale’ tool so that the hands and feet of the fossil visually matched the size of the imprints. To allow simulated movement of the digital skeleton, it was rigged using Maya’s ’joints’ and ’bones’ tools to form a digital marionette in which in principle all anatomical joints can be controlled by the user via the underlying Maya ’joints’. A generic (default) locomotor sequence was achieved by using kinematics observed in the extant species and accounting for measurements taken in the motion analysis of extant species for spine bending, LAR and retraction at the shoulder and hip joints. Previously estimated joint range of motion in the shoulder and hip [139] were not violated by the default sequence. We simplified animation of most joint movements using Maya’s inverse kinematics solvers (see [137, 192] for general introduction into inverse kinematics in 3D animation) for the spine, each limb, and roll-off motion of the hands and feet qualitatively approximating movements observed in extant species. Hands and feet were forced to contact the ground with LP of 0.5, D of 0.75, and a stride frequency of 0.75 (i.e. an idealized walking trot as was occasionally exhibited by all extant species studied). Critically, the simulation allowed user-specified systematic variation of the kinematic parameters body lift (from belly-dragging to more erect postures via abduction/adduction in the shoulder and hip), spine bending (from none to exaggerated via rotation about a dorsal ventral axis in the girdles), and LAR and retraction in the shoulder and hip joints (Fig. 5.5). For each resulting parameter combination, anatomical plausibility was evaluated by checking for bone collisions in the shoulder and hip joints and the spine, as well as for disarticulation of wrist and ankle joints during the stride cycle (if at all, then usually occurring at the beginning or end of a limb’s ground contact). We used a graded score ranging from 1 (disarticulation of joints and/or definite bone collisions) to 4 (no disarticulations and bone collisions) with 2 and 3 scored for minimal and moderate bone collisions, respectively. Soft tissues were not modeled. Since collisions in joints not controlled by user specification may occur due to unspecified joint kinematics found by the inverse kinematics solver, all other bone collisions were not taken into account. Similarly, swing phases were animated, but not taken into consideration.

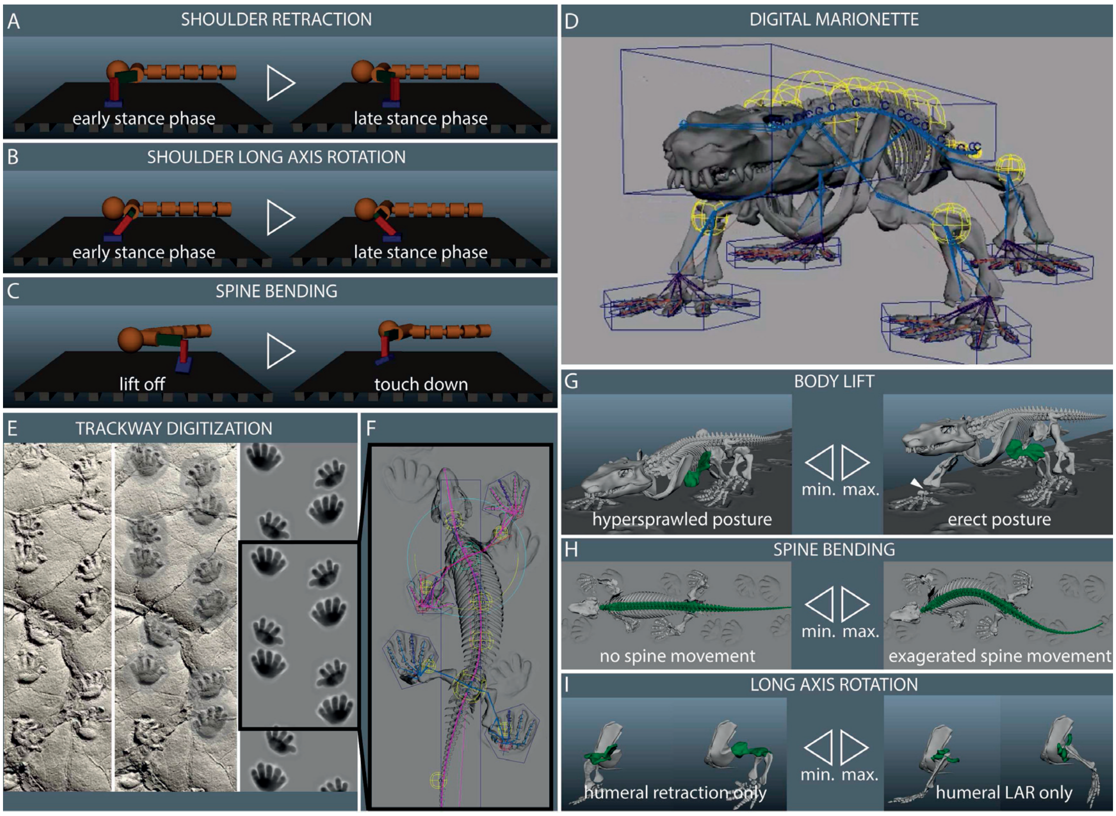


Figure 5.5 – Kinematic simulation of Orobates. **A**-**C**: The generation of body propulsion during sprawling tetrapod locomotion (exemplified for a forelimb). **A**: The humerus (dark green) is retracted in the shoulder joint. **B**: The humerus is rotated about its long-axis in the shoulder joint. Both mechanisms also apply to the hind limb (femoral movement relative to the hip). **C**: Spine bending during the swing phase contributes to step length. **D**: Fully "rigged" version of the digital Orobates reconstruction allowing for systematic variation of body lift, LAR and retraction in the shoulders and hips, and spine bending. **E**: Digitization and idealization of trackways (specimen MNG 1840) for kinematic simulation. Manus (hand) and pes (foot) imprints were idealized and superimposed to fossil trackways to retain stride length, stride width, pace angulation, and manus and pes rotation. **F**: Enlarged portion of the idealized trackway with digital reconstruction of the *Orobates pabsti* holotype specimen placed into the trackway. **G**-**I**: Systematic exploration of the kinematic parameter space. Plausibility of a parameter combination was ruled out if it resulted in bone collision within the spine or within the shoulder and hip joints as well as when disarticulation of limb joints occurred (see white arrowhead in G). **G**: Body lift. **H**: Spine bending. **I**: Long axis rotation (LAR).

5.5.3 Validation of kinematic simulation workflow

To validate the inference of anatomical plausible gaits for Orobates, we followed the same workflow for a disarticulated *Caiman crocodylus* specimen (inventory no.: PMJ Rept 665) housed at the Phyletisches Museum Jena, Germany, and compared the results to the actual kinematics measured during our caiman x-ray motion analysis. First, CT-scans of the specimen were obtained. ’Watertight’ bone surface models were generated using Amira software and the high-detail meshes were reduced using the ’remesh’ tool in ZBrush (Pixologic, Los Angeles, CA, USA). Trackway information (from a photograph) of a representative trial of the female caiman specimen was imported from the trackway analysis of the extant species into Maya and digitized as described above for trackways assigned to Orobates. After import of both models (i.e. digital skeleton and digitized trackways) into Maya, the caiman skeleton was ’rigged’ and animated in the same way as Orobates. Note that for the default stride cycle the available x-ray videos of the caiman were not used. In analogy to the Orobates workflow, 100 parameter combinations of body lift, spine bending and LAR/retraction were tested for anatomical plausibility in the hind limbs (identical scoring as used during the Orobates kinematic simulation; see above). Comparison to the actual kinematics during slow high walks of caimans during the x-ray motion analysis (the caimans did not exhibit any low walks) demonstrated that the anatomically plausible gait parameter combinations identified by our workflow encompassed the actual quantified motion data of living caimans, thus validating the kinematic simulation workflow (Fig. 5.6).

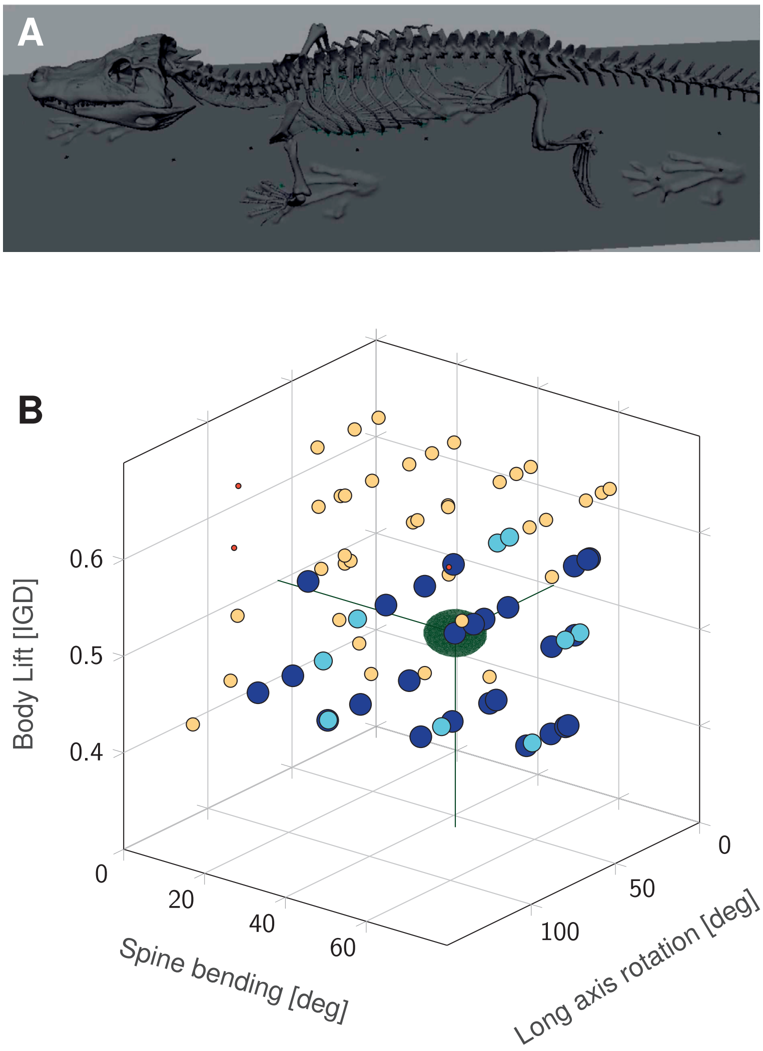


Figure 5.6 – Kinematic simulation workflow validation using caiman. **A**: Maya screenshot of caiman digital marionette walking within digitized caiman trackways. **B**: 100 hindlimb parameter combinations of body lift, spine bending, and LAR were tested (in the same way as described for Orobates kinematic simulation). Scores for each combination were coded by size of the dots (largest dots assigned to perfect plausibility) and color (dark blue assigned to perfect plausibility). Green ellipsoid depicts mean measured kinematics of caiman hindlimb from the x-ray motion analysis ± standard deviation. Note that body lift less than 0.4 resulted in the body moving through the ground and spine bending over 60° resulted in bone collisions within the spine. Actual caiman kinematics (green) are encompassed within the domain identified as anatomically plausible, demonstrating the validity of the kinematic simulation workflow.

5.5.4 Robot design

We followed the same robot design methodology of Pleurobot in our previous work [98], but designed OroBOT to account for the anatomy of Orobates (Fig. 5.2). To recreate the spine we segmented it in 8 actuated joints: 2 for the neck, 4 for the trunk, and 2 for the tail. All joints were coplanar and rotated in the coronal plane (parallel to the ground). The tail was connected through a passive joint in the sagittal plane and suspended in a way it did not touch the ground. This is due to no evidence of tail dragging found in the trackways [188]. Each limb contained 5 actuated joints: 3 shoulder/hip joints in a spherical configuration (protraction-retraction, abduction-adduction, LAR), an elbow/knee joint (extension-flexion), and a wrist/ankle joint for foot rotation.

Each OroBOT foot consisted of three passive compliant joints. The palm and fingers were approximated by a rectangular shape. The size was chosen so the rectangle lies within the footprint area (Fig. 5.2). Width, palm and finger lengths were 8.7 × 5.7 × 4.5 cm for the front foot (manus), and 9.1 × 6.1 × 5.1 cm for the hind foot (pes). The passive joints corresponded to the wrist and ankle joints (foot dorsiflexion-plantar flexion and eversion/inversion) and the metacarpal-phalangeal and metatarsal-phalangeal joints, respectively. These joints were elastic to make the foot comply with the ground, guaranteeing a full foot contact, and also to be able to reset its original state between the steps (see Appendix A.9).

The foot, including the three passively compliant joints, was designed in Autodesk Inventor® 2017 and fabricated using three layers of 2 mm polyoxymethylene (POM), cut with a Trotec Speedy400 Flexx™ laser cutter. The three layers were bonded together with a stretched rubber layer and fixed to the robotic limb with screws. The cutout profile on each foot layer was designed to allow bending in the required rotation axes. Coefficients of elasticity were primarily approximated with finite element analysis (FEA) in Autodesk Inventor® 2017, and then iterated to find a good trade-off between matching the expected stiffness value and the admissible bending without breaking or plastic deformation failure of the POM material.

Because of actuator constraints, the robot was scaled to 1.6 times the size of the fossil (85.14 cm), for a length of 136.22 cm. An additional scaling of 1.4 was applied to the fossil head (11.14 cm) to provide space for the computer and peripherals, increasing the total length to 143.35 cm. Except for the head, all the lengths were geometrically scaled, respecting the same aspect ratios of the fossil reconstruction (Fig. 5.2). Correct geometrical scaling was validated by comparing relative ratios of OroBOT segment lengths with respect to body length, to those in Orobates. Discrepancies remained less than 2.05% (Appendix B.2).

OroBOT's 3D printed parts were fabricated using selective laser sintering (SLS) with polyamide plastic as material. The parts were designed in Autodesk Inventor® 2017. The 3D printed parts provided the structural attachments for the servo motors. We used 24 Robotis Inc. Dynamixel MX-64R servomotors (8 in spine and 4 for each limb) and four Dynamixel MX-28R servomotors for the wrist and ankle rotation. All motors were connected using the standard Dynamixel cables and communication protocol, except for distributed power loops to prevent voltage drops from a 14.8 V power supply. To control the robot, we used a Hardkernel Odroid XU4 Linux Computer.

5.5.5 Mass distribution

Mass should scale with the power of three of the length scale to achieve a correct dynamic scaling [2]. In OroBOT, this results in a four-fold mass increment. As the servo motors' maximal torque was not enough to smoothly execute gaits with such a weight, we decreased OroBOT's mass (6.189 kg) to be only 1.5 times Orobates' mass (3.981 kg), yet maintaining the same mass distribution. We averaged maximum and minimum reported plausible masses [139] of Orobates' head, neck, front limbs, trunk (comprised by pectoral girdle, spine segments and pelvic girdle), hind limbs and tail. We computed relative mass of each segment against the total mass (Appendix B.2). The 3D printed parts of OroBOT were designed, and its servomotors located, in order to comply with the same mass distribution ratios. Discrepancies were found to be less than 1% (Appendix B.3).

5.5.6 Simulated robot

The simulated OroBOT was created in Webots robotics simulator, version 8.5.4. The simulation was tuned to represent real physical quantities of the robot-like size and mass distribution. The ground was represented as a horizontal plane. The coulomb friction coefficient between robot's feet and the ground was set to 0.4, which resulted in a reasonable amount of slipping across multiple gaits. The integration time step of the built-in physics engine was 2 ms. The controller loop ran at a frequency of 100 Hz which corresponded to the control setup on the real robot. The simulated robot was equipped with the following sensors: inertial measurement unit (IMU), joint encoders, joint torque sensors and position trackers (simulated GPS). The IMU was placed in the front (pelvic) girdle and it provides body roll and pitch angles. The position trackers were placed on each foot, providing their position in the world frame. The sensor readings were logged to a text file at the rate of the controller loop execution. Due to the complexity of simulating an elastic material, the manus and pes were approximated with two rigid bodies, representing the proximal elements (carpus/tarsus) and digits, connected by a passive joint. It was attached to the leg via two passive joints with axes of rotation depicted in Fig. 5.2D. The passive joints were modeled as spring-damper systems with tunable parameters (Appendix A.9).

5.5.7 Walking frequency

As it is impossible to measure the speed or stride frequency in Orobates, we used the dynamic similarity hypothesis [3] to define a walking speed for OroBOT. Data from diverse animals with a sprawling posture (Appendix B.5) was used to compute the Froude number ($F_r = h_c(f)^2 g^{-1}$). We used the stride length as characteristic length h_c, the gait frequency f and gravity g. Unlike other approaches, we did not use the hip height as characteristic length [3], due to the sprawling posture nature of the gaits. Froude numbers from the 19 specimens (8 species) analyzed ranged from 0.4 to 3.92 with a mean of 1.42. Similarly, we calculated the Froude number for OroBOT at admissible frequencies within the speed range of the servomotors $f = \{0.25,\ 0.5,\ 0.75,\ 1\}$

Hz. The respective Froude numbers $F_r = \{0.31,\ 1.22,\ 2.75,\ 4.88\}$ were found. We chose to use 0.5 Hz and 0.75 Hz to be walking frequencies for our gait reconstruction because they were in the Froude number range of the analyzed extant species. In order to test whether the stride frequencies used for OroBOT resulted in realistic frequencies in Orobates, based on the Froude numbers used in OroBOT we calculated the corresponding frequencies for Orobates. We found $f = \{0.3, 0.6, 0.9, 1.3\}$ Hz, which are in the range of the data observed in extant sprawling species. As a final check, we also verified the relationships between the Froude number and both the duty factor and limb phase differences as suggested by the dynamic similarity hypothesis [3]. The data obtained when these relationships were calculated for the 19 specimens fall consistently into similar range of values of those of the robot.

5.5.8 Locomotion control of OroBOT

OroBOT was controlled by providing position reference signals to its 28 servo motors. The control was based on solving the inverse kinematics of the legs and the spine in order to produce a desired gait motion. Each of the legs formed a kinematic chain that started at the girdle and ended at the foot. The trunk was a planar kinematic chain that connected the front and hind girdles. The trunk connected anteriorly with the neck and head and posteriorly with the articulated tail to form the spine. The synchronization between legs and the spine was provided by an upper layer of the controller, which was a trajectory generator; thus the inverse kinematics of each kinematic chain could be solved separately.

To solve the leg inverse kinematics, we used an iterative Jacobian pseudoinverse method [19] and formulated the problem as a Quadratic Program allowing us to include joint limits as constraints. Since the number of leg (excluding the wrist/ankle) degrees of freedom (four) was higher than a number of coordinates defining the foot position (three), the solution of inverse kinematics was not unique. The extra degree of freedom was used to adjust the leg posture, which indirectly affected the amount of LAR during walking. The problem was solved numerically using qpOASES solver [49] (Appendix A.1).

The same method was not suitable to solve the inverse kinematics of the spine due to a presence of kinematic singularities (straight spine). Thus we used the same method as in Chapter 3 that relied on spline approximation and fine adjustments through optimization by using a nonlinear solver from the Dlib library [106] (Appendix A.2).

A foot trajectory was described as a parametrized closed curve in the robot's frame of reference. As the feet were complex, with multiple points touching the ground simultaneously, the trajectory referred to the proximal attachment point of a foot (i.e. wrist/ankle). The overall robot motion was the consequence of all feet executing such trajectories at their respective timings. The spine trajectory described the orientation of the girdles in time, which resulted in spine bending. The details about robot trajectories can be found in Appendices A.3-A.5 and in Fig. 5.7.

Measurements acquired from the trackways were used to determine values of the parameters

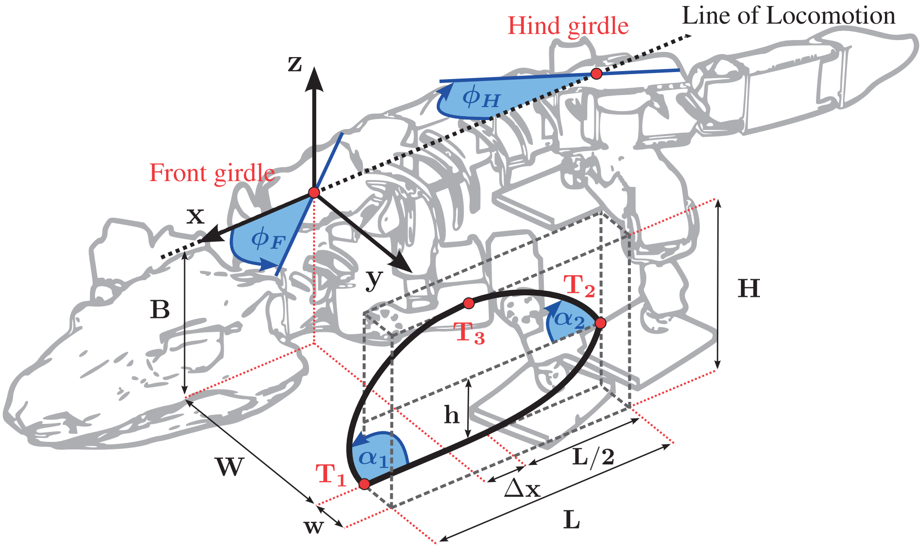


Figure 5.7 – OroBOT's reference frame and kinematic gait parameters. The foot trajectory, composed of a stance phase ($T_1 \rightarrow T_2$) and a swing phase ($T_2 \rightarrow T_3 \rightarrow T_1$), was defined in the reference frame of the corresponding girdle. The spine motion was determined by rotation of the girdles around their vertical axis.

linked with gait kinematics like stride width, length and distance between ipsilateral footprints (Appendix A.7). Values of the remaining gait parameters (e.g. limb phase offset or duty factor D), that could not be inferred from the trackways, were chosen to be biologically reasonable taking into account quantities present in extant animals (Appendix A.8).

5.5.9 Gait exploration and evaluation

Each of the three main gait parameters (Appendix A.6) was assigned eight different values, resulting in 512 evaluated gaits. The evaluation consisted of (i) simulating the robot for a specific parameter combination, (ii) logging the data relevant for the metrics, (iii) processing the data to evaluate the individual metrics (efficiency, balance and precision) for each gait (Appendices A.10 and A.11). All data processing was done in Matlab R2016b.

To draw conclusions about overall performance of each gait, the individual metrics had to be combined. The easiest way would be to use an average score of each gait across the three (normalized) metrics. Such an approach could skew the results towards gaits with a disproportionately high score (an outlier) compared to others in a single metric. To avoid giving too much relevance to a single metric or small variations within it, we adopted and modified the constraint-based exclusion approach [62]. To achieve this, each metric was also assigned an exclusion score. The exclusion score was formed by giving a score of 1 to the gaits that performed better than a user-defined threshold in a respective metric. The remaining gaits were "excluded" by giving them a score of 0. The threshold for each metric was calculated as a K-th percentile to assure each metric had the same score distribution. In Fig. 5.4 we used $K = 50$, which corresponds to the median value.

To combine the metrics, the exclusion scores were summed. The gaits which passed thresholds in the all three metrics had the maximum score of 3. Such approach favored gaits that scored "well enough" across all the metrics (as opposed to gaits that score highly in one metric but not in others). In the final step, the identified anatomically plausible gaits (kinematic simulation) and the dynamically plausible gaits (OroBOT simulation) were combined to find gait solutions for Orobates (Fig. 5.4).

Application Part III

6 Testing Bio-Robots in African Wilderness Prepares them for Disaster Response Missions

All the robots described so far were primarily designed for a specific task confined to a laboratory environment. The goal to tackle disaster response missions requires robots oriented towards a field deployment. With this in mind, we designed a new Krock class of sprawled posture robots. They are still bio-inspired but have a different design approach compared to their predecessors, relying on simple and rapid manufacturing techniques, aimed to increase the robots' robustness and make them easy to repair. The first version Krock-1 was built to be featured in a documentary series filmed in Africa. Such unique opportunity provided us with field experience which shaped the next generation of our robots. The result is Krock-2 - a sensorized robot designed for disaster robotics.

Parts of the material presented in this chapter are adapted from:

MELO, K., HORVAT, T., AND IJSPEERT, A. J. Testing bio-robots in african wilderness prepares them for disaster response missions. *Manuscript in preparation.* (2018).

My contributions: Designed and implemented the robots' software and control framework, performed system integration, developed the long range communication system, designed and implemented the smartphone user interface, contributed to the writing.

6.1 Introduction

Animals display a vast diversity of forms and sizes that are generally adapted to the media in which they locomote. The ways animals move are very rich. They can swim, crawl, run, swing, jump, fly, and use different types of locomotion to escape from predators, attract partners or search for food in very efficient, yet amazing ways [2]. We want to study animal locomotion to reveal fundamental principles of nature, to understand our own body and to give us insights into how to design better technologies.

Biologists and engineers have recently joined forces to create robots used as systematic tools to validate scientific hypotheses about physical interaction with the environment and neuromechanical control of animal locomotion [89, 66]. Robotic replication of animal morphologies and locomotion [98], use of computational models and robots to explain parts of the nervous system [91], and comparison of animal/robot body dynamics during jumping [113], are examples that prove the convenience of robots in scientific research. While robots help us to study nature, the knowledge we obtain while doing it, guides us to make our machines sturdier in tackling perceived and unperceived terrains [194], which is an animal skill desirable to be transferred to robotics.

Disaster response activities often require rescuers or rescue dogs to put themselves in danger in order to preserve the life of others. Robots have a potential to become surrogates or companions to human rescuers when it is required to trudge into complex and hazardous debris and rubble [127]. The adoption of robots as a viable technology for disaster response could have an impact on society, benefiting not just victims but also the rescuers.

The disaster event often produces as a consequence an unstructured environment difficult to locomote through, even for skilled humans and trained dogs. The robots tackling such challenge require advanced locomotion characteristics not present in other robotic applications [87, 50]. In particular, a machine that can imitate the morphology and locomotion of certain animals, while combining the precision and repeatability of conventional robots, would be an indisputable asset for any rescue team. For an example, when a building collapses, different stories will be reduced to flattened layers of rubble material, where pipes and other sanitary and electrical infrastructure are exposed. Tight spaces between the rubble might rapidly fill with water, creating a complex mixed-media scenario. Narrow openings and partially flooded cavities are the only possible ways of getting inside to assess the status of the structure or look for victims. Apart from state of the art sensing, actuation, perception and planning technologies, such scenarios require us to build machines and control algorithms that are engineered to withstand these harsh conditions during operation.

Whereas robustness and reliability are often synonyms for complexity in robotics, simple design approaches proved they can also be effective in complex scenarios [172]. Leveraging simple fabrication techniques and off-the-shelf components, we can achieve satisfying results. Quick fabrication allows for fast iterating through prototypes. Using widely available materials, maintenance costs can be dramatically reduced and repairs can get faster with minimal tools and materials. This is specially useful when time-critical field applications like disaster response require it. Moreover, lower price and rapid construction would make it more acceptable to damage or destroy the robot during an operation. Ideally, rapid prototyping techniques would not only make robots sturdy enough for real world applications, but the affordability of their fabrication could get a broader community get engaged in field robotics by open sourcing, sharing, adapting and reusing robot designs.

To get prepared for disaster response missions, we believe the robots should go through extensive field testing. If the robots are inspired by an animal and they are trying to replicate some of its locomotion capabilities, a logical location for the field tests would be the same environment inhabited by the animal. The experience and lessons from such tests should enable us to adapt our robots to be better prepared for real world applications. Additionally, confrontation of these bio-robots with the natural environment of their living counterparts could propose new research opportunities like studying animal-robot interaction, ethology, evolution and biomechanics.

6.1.1 Challenges

Observing animals in their own habitats is indeed a great source of information to create robust and skilled robots. Designing and testing these robots in such natural scenarios in order to validate their capabilities, is however still a very challenging enterprise.

There are numerous examples of bio-robots designed to mimic and study animal locomotion [89, 66]. However, many of these robots are being tested under controlled laboratory conditions. Furthermore, they often remain in their first prototyping stage and even after their successful role in the advancement of science, they miss the transfer to other real world applications.

Systematic field tests of bio-robots, and especially, tests in natural habitats of the animal that influenced the robot's design are not easy to document. Only a few examples of robots brought to the animal habitats are reported [100]. As it is difficult to systematically collect data outside the laboratory environment, such experiments often serve as mere demonstrations.

Apart from control and biomechanical design, building robots that behave like animals is still challenging in terms of the actuation and sensing technologies. State of the art advances in actuation and sensing are still far away from reaching the power and energy density of muscle actuation [148] and neural sensing. This brings the following questions. Can affordable, rapid robot prototyping and of-the-shelf components provide enough reliability for a bio-robot to be able to endure harsh conditions of a natural environment? With a simple yet robust design, does the bio-robot still keep the ability to reproduce the motion of its associated animal in its own environment successfully? And finally, do these field experiences really improve the design of robots for real world applications like disaster response, whose scenario also imposes new challenges and specific task constraints?

6.1.2 Approach

In order to respond to these challenges, we introduce a new class of sprawling posture robots named Krock (Fig. 6.1). Unique in their kind, these robots are - to the best of our knowledge - the first amphibious robots whose design is informed by animals, have simple, robust and affordable construction, and whose locomotion was tested in the habitats of the same animals for two weeks. The first version Krock-1 (Fig. 6.1A and 6.1B) was embodied in two slightly different robots that resemble the morphology of a crocodile (Fig. 6.1C) (named SpyCroc Fig. 6.1D) and a monitor lizard (Fig. 6.1E) (named SpyLizard Fig. 6.1F). The current and now improved version Krock-2 (Fig. 6.1G) was built based on our experience from testing these robots in the wild. Krock-2 makes the technology transfer from mimicking the animals, to a field application in disaster response by featuring several key improvements.

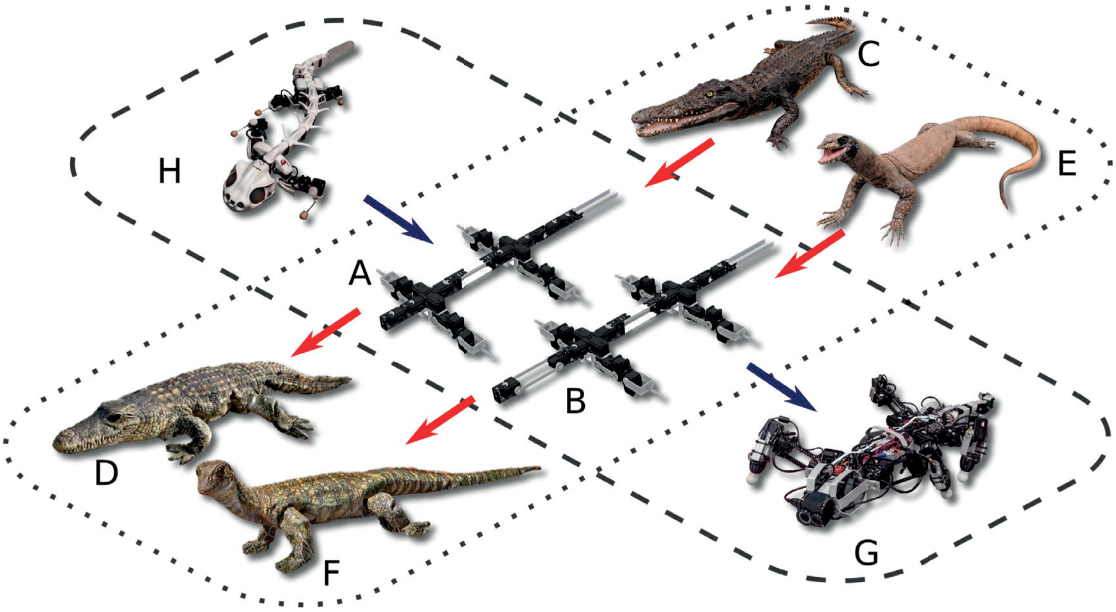


Figure 6.1 – Krock - a new class of sprawling posture robots. The first version of Krock, Krock-1 is present in two variants (**A**) and (**B**). They resemble a crocodile (**C**) and a monitor lizard (**D**) respectively. After covering them with a latex skin, the robots became SpyCroc (**E**) and SpyLizard (**F**), used in the BBC's documentary "Spy in the Wild". An improved version Krock-2 (**G**), intended for disaster response tasks is a product of a technology transfer that started with Pleurobot (**H**).

Krock robots are lightweight and more compact machines compared to their predecessor Pleurobot [98] (Fig. 6.1H). Their design follows a simple fabrication technique that allows rapid prototyping of these robots. The parts these robots are made of are simple, but their simplicity goes along with robustness and easy replicability. Off-the-shelf electronic components, including the computer, actuators and other peripherals, enabled a simple and robust machine integration.

We developed a unique control framework, required to command the different locomotion modes associated with the particular morphology of these robots. The control software provides the robots with a series of capabilities that include leg reflexes [84], leg - spine motion coordination [86], and center of mass control [85].

To test our design approach we went to the same natural environment where the species that informed the design live. Such opportunity came along filming a wildlife documentary with BBC for two weeks in Africa. Besides the requirements on endurance and robustness, the robots' gaits had to resemble real crocodiles and lizards. A latex skin was used to disguise the robots as animals (Fig. 6.1D and 6.1F). The purpose of the robots was to carry high definition cameras to film as closely as possible the daily routines and behavior of the animals.

Operating the robots in the wild, walking in locations like muddy river banks, dusty crocodile nests and swimming in a river, demanded multimodal locomotion capabilities of the robots. High temperatures and complex terrain were motivators to adjust our hardware and controllers on the fly to fulfill the task. A collection of locomotion challenges and integration hardships were turned into requirements for the next Krock-2 version.

The improved locomotion capabilities in Krock-2 are aimed to respond to specific requirements in disaster response. Some of them are the ability to operate upside-down if the robot flips over, intuitive robot control suitable for inexperienced users, full robot state visual feedback for an operator as well as additional sensors and cameras (wide-angle and thermal) to analyze the environment.

The request from BBC

In November 2015, British Broadcasting Corporation's (BBC) producers - John Downer Productions (JDP) in Bristol, UK, presented us with a challenge and a funding opportunity. Due to our experience with bio-inspired sprawling posture robots (i.e. Pleurobot [98] in Fig. 6.1H), they funded the conception of two animal-like robots. The robots were used for filming the wildlife documentary "Spy in the Wild", shot across the Murchison Falls National Park on the Nile River in Uganda, Africa [9]. The robots were supposed to mimic a crocodile and a monitor lizard (Fig. 6.1C and 6.1E). These are very common species found on the African river banks sustaining an ecological rival relationship (the lizards often feed on crocodile eggs [7, 117]). At the same time we were considering to build a new sturdy amphibious sprawled robot to fulfill the expectations of the Swiss NCCR-Robotics Rescue Challenge [133]. Therefore, we proceeded to design and build Krock-1, a field-ready platform with multimodal locomotion capabilities that simultaneously fulfills the requirements of BBC.

The offer from the TV producers came with a series of requirements and constraints that shaped our robots. Besides a mandatory robustness to operate in the field, the robots should carry a pair of high definition cameras, be able to continuously operate for long periods of time (~ 30 min), look and move as close as possible to the real animals, and be waterproof. The robots were required to walk and swim, potentially next to the real animals in their exact terrain and aquatic conditions.

The shooting took place across several locations in the park. We were required to keep the robot prepared at all times, for quick and repeated deployments, before the lighting conditions change or the wild animals around us move away. To ensure the robot operator being at safe distance from the wild animals, a remote communication link was mandatory (the range up to 100 m).

6.2 Results

6.2.1 Krock - a new class of sprawling posture robots

The two animal species we based our work on were a juvenile Nile crocodile (*Crocodylus niloticus*) and a large adult monitor lizard (*Varanus niloticus*) (Fig. 6.1C and 6.1E). We chose these developmental stages in our reference animals due to their similarity in size, sharing similar snout-to-vent length (SVL, a length measured from an animal's nose to its tail base) ~ 76 cm and of total length of ~ 1.3 m [136, 38]. This allowed us to reuse the same base structure (i.e. spine and girdles) for both robots (Fig. 6.1A and 6.1B). Although these species share many morphological and biomechanical characteristics with salamanders (i.e. our previous model animal [98, 91]), some details of their locomotion, including higher postures, leg trajectories, spine leg timing and body balance, were completely new. We had to adapt both the kinematics and locomotion controllers to match the new animals motion [158, 168]. This was an iterative process, involving advice from zoologists and wildlife experts, where we were challenged to achieve visual resemblance to the real animals.

We leveraged the bio-informed legacy of our salamander-like robot Pleurobot, to determine the morphology of Krock robots. The four DoF limbs of Pleurobot are designed to closely replicate salamander gaits. However, they offer kinematic redundancy and a large enough work space so only a small adjustment of the limb kinematics was needed in order to mimic gaits of the new animals [84, 63]. Similarly, a highly actuated spine was necessary to accurately reproduce walking and swimming gaits, however a satisfying similarity can be achieved with simpler spine kinematics. A lower number of spine joints (two in Krock) reduces the weight and power consumption, while improving robustness and increasing payload volume.

We followed a simple, yet scientifically grounded procedure for the creation of our robots (Fig. 6.2A). We informed the design of Krock robots by studying the morphology of the animals we wanted to reproduce. Two taxidermy specimens (Fig. 6.1C and 6.1E), a *Crocodylus niloticus* (SVL = 92 cm) and a *Varanus niloticus* (SVL = 56 cm) were carefully photographed and measured (Fig. 6.2B). Further understanding of their body structure was obtained form the data belonging to similar species (for more details see Section 6.4.1).

To obtain the final dimensions of the robots (SVL ≈ 76 cm), the dimensions of the taxidermy specimens had to be scaled (crocodile scaled down and lizard scaled up). Achieving dynamic

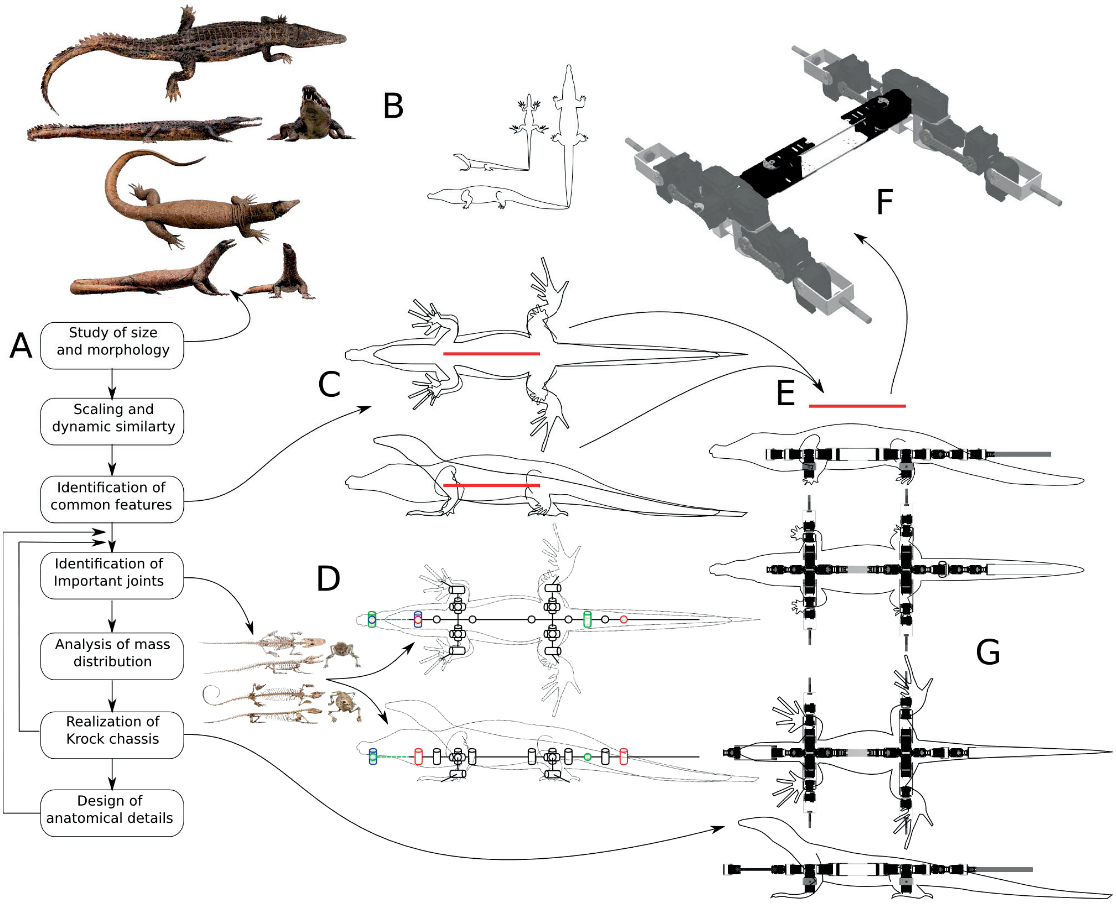

A
Study of size and morphology
Scaling and dynamic similarty
Identification of common features
Identification of Important joints
Analysis of mass distribution
Realization of Krock chassis
Design of anatomical details
B
C
D
E
F
G

Figure 6.2 – Bio-informed design of Krock. A simple and scientifically grounded procedure for the creation of animal-like robots. (**A**) Steps of the design approach. (**B**) Different spatial views of the taxidermy specimens used to create anatomical silhouettes of the animals. (**C**) Superimposition of animal silhouettes, scaled to match the inter-girdle distance IGD of 42.56 mm (in red). (**D**) Identification of the main joints by studying the animals' skeletal structure. Placement of the joints in the silhouettes (black are shared by the two robots, red exclusive for SpyCroc, blue for SpyLizard and green are passive joints). (**E**) Realization of Krock chassis. (**F**). Details of the common chassis, featuring two DoF spine and limbs for the two animal robots. (**G**) Addition of anatomical articulated details including neck and tail.

similarity in the gait of a bio-robot is crucial when locomoting in the same environment of the robot's biological analogs [98, 120, 40]. Proper scaling will guarantee correct dynamical interactions of the robot with the environment. For that, the robots require to move at speeds dictated by the relationships between inertial and gravitational forces (i.e Froude number [2]). In the case of Krock, as it is not replicating agile gaits like running, but simple walking and slow swimming, it was still able to use speeds similar to those observed in nature. In addition, we studied allometric growth[1] relationships of crocodile [24] and lizard [25] growth. Positive allometry is observed in lizards as they develop longer limbs in larger-bodied species. SpyLizard

---

[1] Different body parts grow at different rates, as opposed to isometric growth.

is relatively large compared to the average size of monitor lizards [38]. Their positive growth allometry allowed us to relax the constraints on the final limb dimensions to be slightly larger than the given lengths in the isometric scaling. This was convenient as the same limbs could be used for SpyCroc, simplifying the fabrication and control of the robots.

Common features between the two animals were identified by graphically matching their morphologies (Fig. 6.2C). The comparison guided the dimensioning, joint placement and mass allocations of the different robot body parts. Both robots share the same chassis (Fig. 6.2F), consisting of the pectoral girdle, the spine and the pelvic girdle (trunk). The dimension of these body parts matched both animal morphologies (Fig. 6.2E), with exceptions of the limb size, which were guided by the allometric growth data [24, 25]. Finally, necks and tails were designed to provide required motion versatility. The necks were responsible to provide animal-like motions of the head containing the cameras. Role of the tail actuators was to produce undulatory motion required for swimming.

6.2.2 Born in a lab, to live in the field

To prepare for the field deployment, we had to consider the average terrain topology and environmental conditions the robots were going to face. These characteristics guided a concurrent design of the gait, fabrication, choice of materials, communication devices and system integration. This process was entirely carried out inside the laboratory.

Working first under controlled laboratory conditions provided large flexibility in choosing the devices and systems we can use in our robot for sensing, communications and power. Likewise, locomotion testbeds (treadmills, man made rough terrains etc.) can easily be adjusted or reconfigured for experimental purposes. The Krock class of robots was created to robustly interact with an environment deprived of any comfort provided by the facilities we had in the laboratory. Constraints for the operation of the robots on the field lead to requirements that can be classified as (i) limited bandwidth, long range communication, (ii) power trade-offs, (iii) robust walking performance. Being able to comply with these requirements in the laboratory made the robots ready for the intended tasks.

Given the dangerous nature of the operating environment and the proximity to wild animals, we set a conservative range for our communication distance of 100 m. As this range was higher than previously used Bluetooth gamepad, a new long range RF system had to be developed. Details about this system can be found in Section 6.4.4.

As quick and frequent robot deployments were an important requirement, we made sure all essential robot functions were controllable by the gamepad. The robot's computer was configured to allow for automatic connection of the gamepad (upon pressing the Bluetooth pairing button) as soon as the computer loads the system. Furthermore, essential functions like starting or stopping the locomotion controller and shutting down the computer were mapped to specific button combinations. This made it possible to control the robot without an additional computer.

However, in case the controller had to be modified (e.g. change the gait parameters), we set up a portable wireless router to access the robot's computer from an external computer, connected to the same network. Furthermore, the gamepad enabled the operator to seamlessly choose between different operating modes (i.e. walking, inspection and swimming).

During extensive testing in the lab, the robots are usually connected to an external power supply. The average power requirements are ~10 A @ 14.8 V when the robot is in operation and ~2 A @ 14.8 V during standby (~1.5 A consumed by the actuators and ~0.5 A by the robot's computer). Based on the observed power consumption and the requirements on the robot autonomy, we chose a 5800 mAh Lithium-polymer battery (612 g, 145 × 48 × 38 mm) to power the robot in the field. Further laboratory tests with the on-board battery demonstrated that the autonomy of the robot was at least 30 min under the same testing conditions as when the external power supply was used.

Krock-1 robots and other similar sprawling posture legged robots are not known for being very dynamic compared to other legged robots [87, 176, 146, 16]. Nevertheless, reduction in their total mass (compared to its predecessor Pleurobot, with 7.8 kg excluding batteries), boosts their walking performance in terms of increased ground clearance and endurance while reducing the actuator power expenditure. By carefully choosing the construction materials and keeping a minimalistic structure, the total robot weight was reduced down to 4.6 kg. The reduced weight provided a needed increase in the payload capacity as the robots had to carry a high capacity battery, heavy latex skin and the cameras.

Anticipating the conditions the actuators will encounter in the field, we ran laboratory tests with the robot to understand the overheating, overloads and other actuator related issues. For example, we ran experiments where an operating actuator was submerged in water, in order to predict possible damages, set failure diagnostics policies and be prepared to repair or perform a maintenance when needed.

It was required to place the actuators into a robot's structure that resembles the morphology of the animals according to the bio-informed design guidelines. Since the actuators were used as structural parts, additional structural connection elements were chosen to (i) respect the volume and mass distribution, and (ii) provide structural strength to hold all pieces together while the robot robustly interacts with the environment. This led us to the selection of high strength lightweight materials like carbon fiber and aluminum. Although carbon fiber is rather expensive, the strength to weight ratio makes it the preferred material for the limbs, as they suffer from the highest strain during locomotion. Aluminum was preferred for more complex (volumetric) parts where bending was required. The fabrication process was simple as most of the connecting pieces were either flat or made by bending flat elements. This allowed us to directly transfer the structural features (shapes, holes, bend lines) from CAD to the material by e.g. laser engraving (for the details see Section 6.4.2). Since aluminum flat bars (of different width options) are common goods in hardware stores, using them reduced material cutting times and made the fabrication possible with simple tools like a drill, saw, vise and hammer.

The shared spine and limbs in SpyCroc and SpyLizard were intended to be highly modular, allowing fast disassembly by removing few screws. This allowed us to pack the robots in a very efficient way for transportation - they can be carried in a normal backpack and made operational within 15 min.

6.2.3 Testing in natural scenarios

The filming in Uganda took place on shores of the Nile across two weeks[2]. Each day started by loading the robot on a small river boat and searching for an appropriate filming location. As we filmed across multiple locations each day it was crucial for the robot to be easily handled and operated, even by a single person. Our usual deployment sequence had the following steps:

1. Connect the battery and close the outer suit (~ 1 min).
2. Wait for the computer to load the system and connect the Bluetooth gamepad (~ 2 min).
3. Start the locomotion controller via the gamepad.
4. Perform walking or inspecting tasks as needed for filming purposes (5-20 min). Between the shooting sequences, the locomotion controller can be stopped to reduce the battery drain.
5. Turn off the computer (by the gamepad) and disconnect the battery (~ 1 min).
6. Replace the battery (~ 1 min).

Different filming locations required the robot to walk over grass (Fig. 6.3A), rocks (Fig. 6.3B), and even dirty (Fig. 6.3C) or muddy terrains (Fig. 6.3D). Across all the terrains, the robot used the same gait - a walking trot with a duty factor of around 0.8. While the aesthetic latex skin gave the robot a realistic look, it had a negative impact on the performance. The skin, support jackets mounted around the girdles and bubble-wrap padding tucked under the skin, used to provide a realistic volume, hindered the range of motion of the limbs and increased the weight. Similarly, the padding and stiff plastic underbelly plate, used to close the skin, constrained the spine motion. To avoid damaging the skin, the walking gait had to be adjusted by decreasing the stride length and swing height, which decreased the walking speed and the balance (e.g. the stride length had to be reduced up to 30%). The increased trunk volume caused the underbelly to occasionally drag on the ground, however a wide support base of the sprawling gait made flipping over not an issue.

For filming purposes, the robot often performed an inspection mode combined with a short walk. In the inspection mode, the robot kept all the feet on the ground while the operator controls the front girdle position (within the kinematic limits) and head movements. While the resulting motion looks lifelike, it also gives a good overview of the area for the head mounted cameras.

---

[2] The video compilation can be fount at: https://biorob2.epfl.ch/video/316

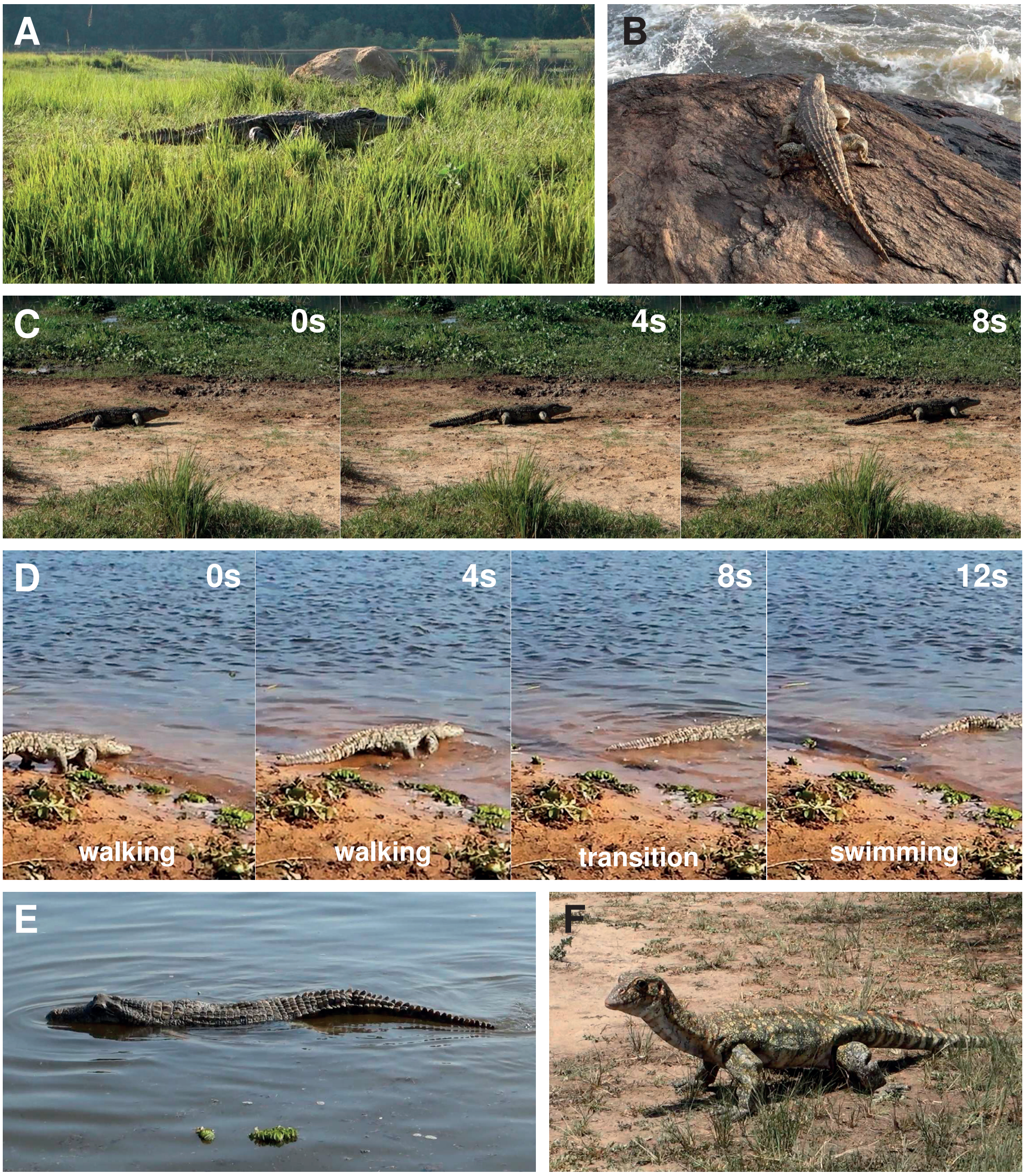


Figure 6.3 – Different terrains encountered in Uganda. (**A**) - (**C**) SpyCroc walking over grass, stone and dirt. (**D**) Transitioning from walking to swimming. (**E**) A close up of SpyCroc swimming. (**F**) SpyLizard performing inspection mode.

The combination of additional weight the robot had to carry and the heat accumulated within the skin, caused the shoulder and hip actuators, supporting most of the load, to sometimes shut down due to an overload protection (i.e overload admissible limits decreased due to increased temperature). To prevent it, the robot was continuously operated only for short amounts of time

$(5-20$ min), with cooling breaks in between. During the break, we opened the skin underneath to improve the heat dissipation. Even though the robot computer was also placed inside the skin, we have not experienced any heat-related issues with it.

To swim in the Nile, the robot used an anguilliform swimming gait (Fig. 6.3E), where the spine and tail produce traveling waves propagated along the body, propelling the robot in water [98]. To control the body undulations we used an open-loop controller sending sinusoidal position references to the spinal actuators with predefined phase offset between consecutive joints. The performance was modest (compared to Pleurobot swimming [98]) as the spinal movements did not transfer well through the suit and padding. The tail skin was loosely fitting the tail and fin, meaning they could move inside without effectively translating the motion through the skin. Moreover, the plastic bags attached and sealed with hose clamps and silicone glue proved to be a cumbersome and unreliable method of waterproofing.

The transition between walking and swimming was triggered manually via the gamepad. The switching between inverse kinematics based walking controller, and joint space swimming controller happens instantaneously (Fig. 6.3D). To prevent discontinuities in position references for the actuators, all the joint reference signals are passed through a low pass filter. In the moment of the state switching, the filter time constant is set to a nominal value of 2 s, after which it decays towards 0 s. Such mechanism allows smooth transition between the states and it does not introduce delays once the transition has finished.

Due to the narrative of the BBC's documentary, we mostly used SpyCroc, while SpyLizard was only tested on one occasion (Fig. 6.3F). Constant use of SpyCroc required necessary maintenance and repairs on the field. Exchanging a single actuator was easy and could be done within an hour. However, when multiple actuators got damaged by water, we were able to quickly turn SpyLizard into SpyCroc. Since they share the same base structure (trunk and legs), only the head and tail had to be switched exploiting the modular character of the robot.

After two weeks of continuous use, we were left with one fully operational robot, while the other had several broken actuators due to the water damage. We also noticed an increased internal friction (reduced backdrivability) of spinal actuators due to often overloads. As the actuators are not dust proof, and the waterproof layer was only equipped while swimming, some actuators had noticeable amounts of dirt and sand inside the gearbox after each filming session.

6.2.4 Addressing the disaster response challenges

Our experience with Krock-1 robots in Africa gave us valuable insights on how to design an improved version of the robot — Krock-2 (Fig. 6.4). The spinal actuators often suffered from being overloaded and would regularly shut down, or even get damaged. Since the more powerful Dynamixel MX-106 already proved to be quite resilient in Krock-1's shoulder/hip joints, in Krock-2 they were chosen to also actuate the spinal joints. In a similar manner, we decided to replace Dynamixel MX-28 actuating Krock-1's tail with MX-64. The smaller and lighter MX-28 was chosen to reduce weight of Krock-1, however their lower output torque reduced the tail's performance. As the tail is important for swimming, Krock-2 uses the same type of tail actuators that were already tested in swimming experiments with Pleurobot [98].

The waterproofing solution of Krock-1 used in Africa is suitable only for short instances of swimming and it requires a new application of the sealant each time it is deployed. Since Krock-2 does not have to wear a decorative skin, it can reuse waterproofing solution of Pleurobot that relied on wearing a waterproof suit (Fig. 6.4).

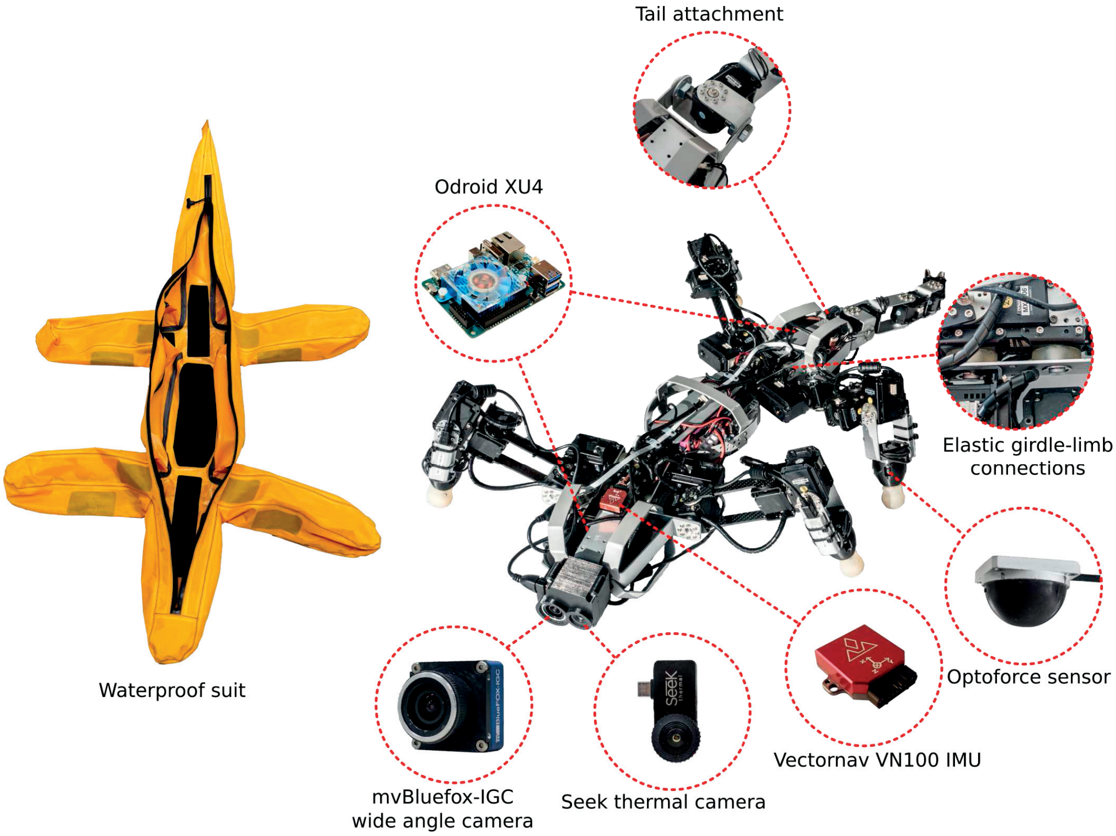


Figure 6.4 – Krock-2 and its main features. The newly added sensors, not featured in Krock-1, are IMU, force sensors and two cameras located inside 3D printed head. Due to increased amount of peripheral devices and increased requirements on computational power (video processing), a second Odroid is added. The modular tail can be quickly detached by removing only two screws. To dampen impacts and their transmission through robot's body, the limbs are attached to the girdles through elastic elements (urethane, shore 50). Furthermore, the robot is equipped with a new professionally tailored waterproof suit with a waterproof zipper.

The battery autonomy of Krock-1 was sufficient for our requirements in Africa, however it can be further improved without switching to a higher capacity battery. Even though the actuators are turned off when the locomotion controller is not running, they are still connected to the battery and the entire motor chain is draining ~ 1.5 A at 15.4 V. To prevent an unnecessary battery drain, the motor power line in Krock-2 has an electrical switch, with the control input connected to the robot's computer. To easily toggle the switch state, its control is linked to a gamepad.

The limb morphology, as expected from our experience with Pleurobot, provides a wide workspace suitable for different terrains. Thus we reused a similar, but slightly modified morphology in Krock-2. Changing the order of the first two shoulder joints does not significantly change the limb workspace, but it makes the robot symmetric in coronal plane. Such a feature makes the robot invariant to being flipped over. Moreover, a flatter body allows it to move under low obstacles and passages (Fig. 6.5). Both features are interesting for applications in disaster scenarios.

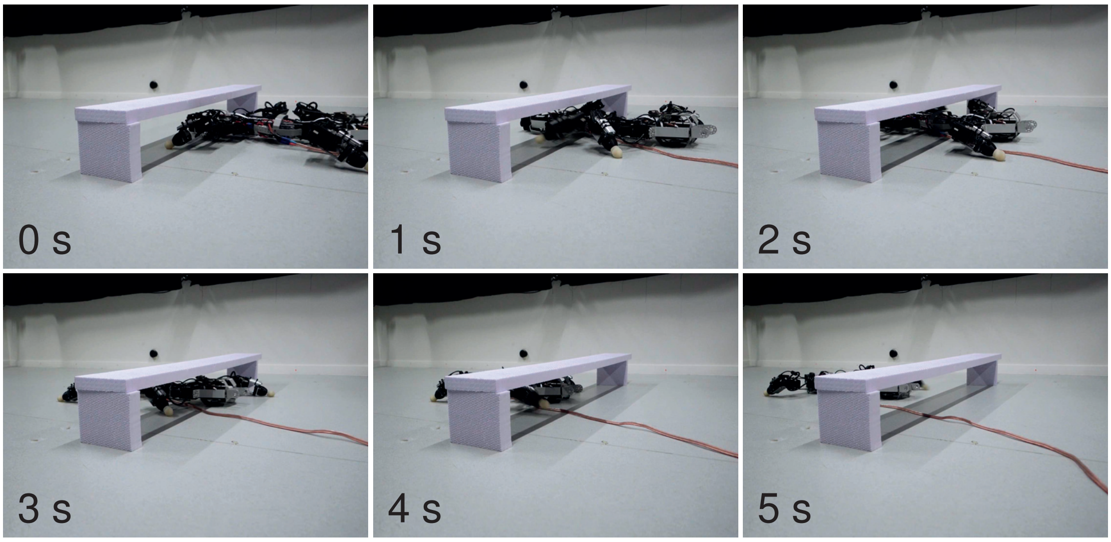


Figure 6.5 – Snapshots of Krock-2 passing through a low passage. The passage height is 13 cm.

An another modification was a compliant element between the limbs and girdles. Recurrent stepping impacts damaged the structure over time in Krock-1 and contributed to the attachment screws getting loose. The compliant element can potentially decrease such wear over time and prevent ground impacts to get transmitted to the gearboxes of the actuators.

While Krock-1 is designed to be simple, lightweight and reliable, Krock-2 is supposed to be a more complex machine, specialized for search and rescue challenges. As a consequence, Krock-2 is equipped with additional sensors: an IMU, force sensors mounted between a foot and an elbow/knee and a set of conventional and thermal cameras (Fig. 6.4).

6.3 Discussion

The design flow of Krock class robots was a mix of two directions. In one direction (blue arrows in Fig. 6.1), there is a technology transfer that starts from fundamental scientific research (creating robots to study animal locomotion). It passes through the phase of creating sturdy animal-like robots and finishes by incorporating both bio-informed design and exhaustive field testing into robust platforms. These platforms are created with a demanding field application in mind: disaster response. On a transversal direction (red arrows in Fig. 6.1), we demonstrated that our scientifically rigorous informed approach in designing robots, using real animal data was successful. It enabled us to achieve an aesthetically realistic look and movements of the animal inspired robots (i.e. morphologies, mass distribution, dynamic similarity, gaits), which was used for wildlife filming. In addition, it also showed that the simple fabrication and integration of the various systems was robust enough to allow the robots to overcome intensive field testing. Such testing was done in the environment that shaped the morphology and locomotion strategies of the real animals. Recreating parts of their locomotor behavior with the robots, serves as validation of our methodologies and as a starting point for proposing new scientific hypotheses.

The challenges of testing in natural habitats of the animals that inspired the robots and the requirements that came along filming the documentary series pushed us to build better machines. Design flaws that could be neglected in the laboratory environment, had to be dealt with in the field. Convenient assumptions and simplifications that are often very tempting to use in our advantage, were less common in this scenario. For example, we had to use the batteries instead of an external power supply, deal with an additional payload that hindered the robot's motion, rely on a simple gamepad to fully control the robot and be prepared for fast deployment and repairs. As a consequence, a systematic data collection was a challenge (as discussed in Section 6.1.1) and it mostly relied on recording video footage.

The unique experience taught us about advantages and disadvantages of the used technologies. Proven robustness of the off-the-shelf components and their integration into robotic system, gave us confidence to rely on them in the future projects. The observed limitations provided valuable lessons how to mitigate them in the next iteration of Krock robots. The result was Krock-2 robot, which reuses the sturdy base of Krock-1 and expands it with additional features, sensors and subsystems to make it suitable for disaster response missions.

In the end, we would like to propose new scientific avenues based on the work presented here. The bio-informed methodologies and simple fabrication techniques provide the scientific community with potential tools to test hypotheses related to animal locomotion. Synthetic studies of animal locomotion require a physical platform in order to capture the real world physics[3] of the animal interaction with its environment. With our contributions in this chapter, we would like to make such platforms more affordable while remaining precise enough for scientific purposes. This paves the way for biologists and roboticists to join their efforts to use bio-informed platforms in a variety of research topics. For example, designing simple robotic mechanisms aimed to recreate biomechanical data from animals would enable a wider range of comparative morphology studies. In evolution, bio-informed robots can be used to recreate the locomotion of extinct animals, which was demonstrated in the previous chapter. Finally, animal-like robots, similar to SpyCroc and SpyLizard, could bring new opportunities in research of animal behavior, e.g. to study animal-robot interactions.

---

[3] Numerical simulations are useful too, but interactions with a complex environment (e.g. water and mud) are difficult to simulate properly.

6.4 Materials and methods

6.4.1 Bio-informed robot design

Two taxidermy specimens were used in the design of our robots: a *Crocodylus niloticus* (Catalog No. MZL-8631) and a *Varanus niloticus* (Catalog No. MZL-1267) from the Cantonal Museum of Zoology, Palais de Rumine, Lausanne, Switzerland [21]. The specimens were photographed and measured. The data was complemented with illustrations of crocodile's skeletons [118], monitor lizard's CT scan data [174] and photographs and measurements of an alligator (*Alligator mississippiensis* Catalog No. MZL-24013) and a varan (*Varanus griseus* Cat. No. MZL-24012).

The dimensions of all the studied specimens were isometrically scaled in order to match the final dimensions of our robots (SVL ≈ 76 cm). There was no need for further dynamical scaling using the Froude number because both the robots and the studied species share relatively similar sizes and velocities when performing slow walking gaits.

To compare the animal morphology we made silhouettes of the top, side and front views of the animals, which were further scaled to the desired length of the robots. Superposition of the corresponding silhouettes revealed that both animals' trunks (i.e. pectoral girdle, spine segments and pelvic girdle) shared similar lengths (42.6 cm). Therefore it was possible to design the girdles and the spine that were interchangeable between the crocodile-like and lizard-like robots.

After dimensioning the various body parts, the silhouettes and the skeletal data helped to identify important bone articulations. The previous work on sprawling posture animals [98] was reused to determine the placement of the limb joints required for reconstructing the locomotion. Additionally, this methodology was used to ensure a valid segmentation of the animal's trunk with a minimum number of joints. An assumption that the thickness of the animal's body parts roughly represents their mass (uniform density) guided a placement of the actuators to achieve a similar mass distribution.

Despite being highly segmented in vertebrates, the spine was designed to have only two joints for three main reasons. (i) In walking gaits, the spine is responsible for rotating the girdles to maximize the stride (shown in Chapter 3). This is also achievable with a reduced spine segmentation. (ii) In swimming, the spine and tail, used as a single kinematic chain, are the main undulatory mechanisms that propel the animals. The final robot segmentation with a two joint spine was not ideal for efficient swimming [151] but was good enough to ensure the wave propagation. (iii) A reduced number of joints in the spine leaves more volume for a payload.

The crocodile's big head and strong neck are viewed as a small articulated system. As the head mobility in the vertical plane is limited [52], the SpyCroc's neck featured two actuated joints that allowed the sideways motion of the head up to 90° in both directions. This was aligned with the fact that the robots were mobile platforms for the on board cameras, requiring a good panoramic range. The ratio of head size and neck length of the crocodile differs from that of the lizard, featuring a small head and long, flexible neck [155]. Thus, SpyLizard's neck had an extra joint to drive a parallel four bar linkage lifting the neck while keeping the head parallel to the ground (Fig. 6.2). While resembling the neck of the animal, it also brought a higher perspective for the cameras.

To have the same number of actuators in both robots, the tail mechanics was designed to balance the number of neck actuators. Following the robot requirements and the animal morphology, the SpyCroc's tail featured three joints that allowed it to swim. In the case of the SpyLizard, which was not required to swim, the tail had two joints. Both tails featured a passive joint in the sagittal plane, which was implemented with an inactive actuator (used as spare part). The tails ended with an elastic bending plastic bars (Fig. 6.2). Both heads were made of a high density polystyrene foam housing the cameras where the animal's eyes would be. The head foam was made at John Nolan studio in London, UK.

The limbs, attached to the girdles, consisted of a three-joint shoulder/hip (yaw, pitch, roll), a single elbow/knee joint and ended with a rubber ball-foot. The joint configuration was the same as in Pleurobot (Fig. 2.1).

6.4.2 Rapid fabrication

All the structural parts of Krock class robots were constructed using two types of lightweight material. The design combined the formability of aluminum with the strength of the carbon fiber. For aluminum parts we used extruded flat bars (30 × 2 × 100 mm) of anodized Aluminum 6061, with density of $2.7\,\mathrm{g\,cm^{-3}}$, commonly used in architectural applications and easily found in any hardware store. The bars had T4 temper which allowed a bending radius as low as 1 mm. The carbon fiber parts were made of HT prepreg carbon fiber fabric with epoxy resin matrix panels (150 × 3 × 340 mm) from Swiss-composite (Fraubrunnen Switzerland) with density $1.56\,\mathrm{g\,cm^{-3}}$ and 1050 MPa of flexural strength.

All the used actuators were the Dynamixel MX series servo motors from Robotics Inc. Seoul, Korea. Both SpyCroc and SpyLizard used four MX-106 for each of the limbs' pitch joints and 14 MX-64AR for the remaining limb and trunk joints. The neck and tail used six MX-28AR (five active and one passive). In Krock-2, the two MX-64AR used in the spine got replaced with more powerful MX-106 increasing their total number to six. The tail became stronger featuring three MX-64AR actuators and it could be easily attached/detached through a passive sagittal plane joint that connects it with the girdle. As the actuator's enclosure is made of durable aluminum and plastic, they were also used as structural parts for the robots. The eccentric output shaft of the actuator's body allowed a flexibility in using the extra volume to reduce the amount of structural material required to connect the joints.

The aluminum bending parts were produced by a simple, cheap and relatively rapid manufacturing technique (Fig. 6.6). All the robot parts share a common width (30 mm) and thickness (2 mm), therefore we were able to use the flat aluminum bars directly from the store. All the pieces were designed in the CAD software Autodesk Inventor® 2017. Some of the pieces were simple flat bars with hole extrusions (similar to the carbon fiber pieces) while the others had to be bent (Fig. 6.6A). The surface features were represented as 2D technical drawings, including bend line and hole center annotations (Fig. 6.6B). The drawings were engraved on the aluminum flat bars by using Trotec Speedy 4000 laser cutter (Marchtrenk, Austria) (Fig. 6.6C). If the aluminum is not anodized the engraving procedure will not work. In that case, we recommend to put a transparent tape over the material, as the laser cutter will be able to cut into it and imprint the desired pattern. An alternative low cost solution, not requiring a laser cutter, is to print the pattern on an adhesive paper and stick it to the material. Once the cut lines, bend lines and hole centers were properly marked, the individual pieces were cut with a saw (Fig. 6.6D). Hole centers were punched by hand (Fig. 6.6E), drilled (Fig. 6.6F) and cleaned (Fig. 6.6G). Alternatively, all the cuts and holes can also be produced by using a CNC-Mill or a waterjet cutter, as it was done with the carbon fiber. To produce bends, the aluminum bar is placed in a vise, aligning the first bend line with the edge of the vise. The bar was hammered until it reached half of the final bending angle (Fig. 6.6H). Then, the second bending line the bar was aligned with the edge and hammered until reaching the final bending angle. The comparison of the finished piece with the blueprint is shown in Fig. 6.6I. The fabrication of a single piece like the one depicted in Fig. 6.6 took only 9 min. Production of multiple parts at once would result in even lower manufacturing time per piece.

Waterproofing

Krock-1 robots SpyCroc and SpyLizard featured a double layered skin. Both layers were developed by John Nolan Studio in London, UK. The inner layer was shared by the two robot versions, while the external one was the skin that characterized each of the animals. The inner layer counted two 3D printed parts ("jackets") with holes for the limbs, spine and neck or tail. Each jacket, composed of two connecting halves, wrapped around each girdle. The role of the jackets was twofold: to provide an extra volume and attachment points for the outer layer, and in the case of SpyCroc to enable waterproofing. The round holes in the jackets allowed us to place custom made plastic bags and rubber tubes around each appendage. They were connected and sealed to the jacket with a large worm-drive hose clamps. To improve the sealing properties of those joints, they were additionally covered with a liquid silicone, making the whole setup not reusable. The outer skin layer was made of specially painted latex to realistically resemble the animals. The main part covering the body was devoid of limbs. The robot slid in through an opening on the bottom side of the skin. The opening was also used to easily access the battery and cameras when no inner waterproof bags were used. The skin covering limbs was made of painted latex gloves with long fabric sleeves, used to tie it around the jacket.

For Krock-2, we developed a custom waterproof suit made of flexible ripstop coated fabric of 0.5 mm thickness, sealed with thermal-seam-sealing tape. The suit is closed with a waterproof Tizip® Masterseal 10 zipper, spanning from head to tail. The suit materials were cut, sewed and sealed by Outdoor Repair GmbH Uster, Switzerland. The suit reaches the IP67 standard (IP Code,

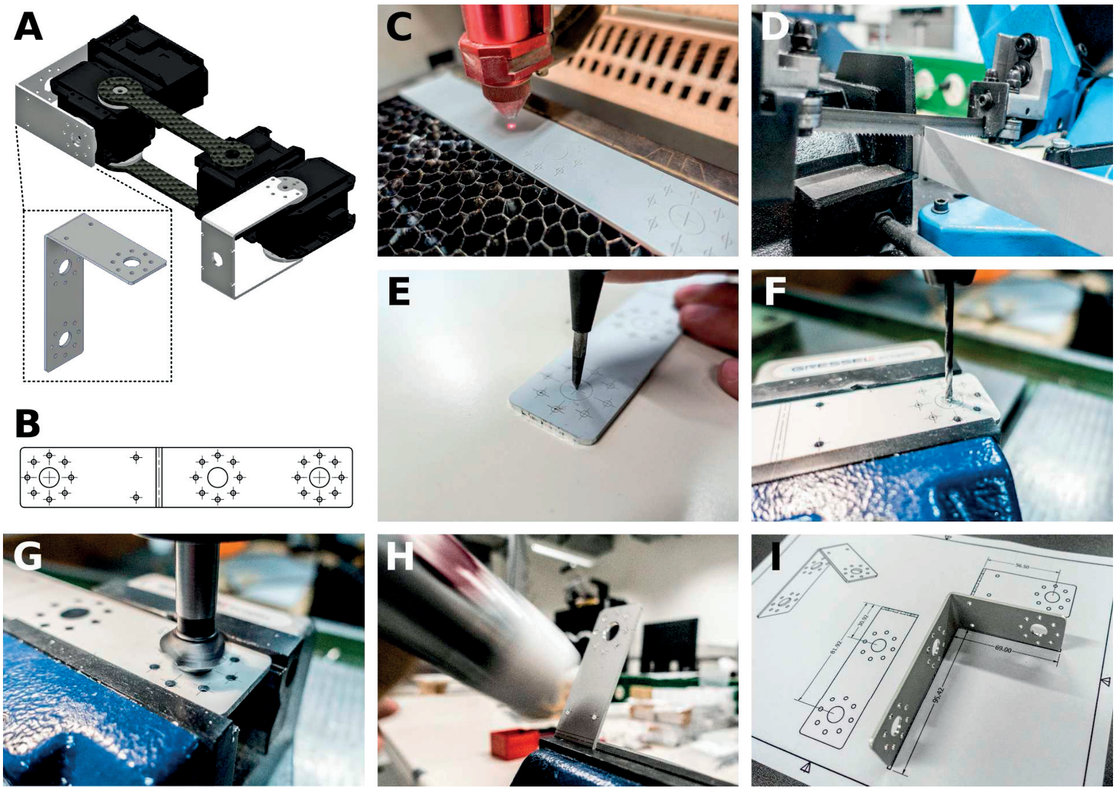


Figure 6.6 – An example of producing an aluminum part of Krock robot. (**A**) The location of the part in Krock-2's leg. (**B**) 2D technical drawing of the part. (**C**)-(**H**) Laser engraving, cutting, hole punching, drilling, cleaning and bending steps. (**I**) The produced part.

International Protection Marking, IEC standard 60529), although prolonged use can damage the waterproof coating and cause leakage. Minor damage can be repaired by applying heat to the internal thermic coating.

6.4.3 Control and integration

Core robot architecture

Krock-1 and Krock-2 share the same core architecture: actuation system based on Dynamixel servo motors and Linux based single board computer Odroid-XU4 as the computational unit. The servo motors and the computer are connected via USB-to-RS485 adapter, while the low level communication between them is based on a library provided by ROBOTIS Inc. for DARWIN Electronics Kit 2. As an input device, a Sony PlayStation 3 gamepad is used, allowing a user to provide high level commands to the robot. The gamepad connects to the computer via Bluetooth.

The core software modules are Locomotion controller and Interconnect module. Both are separate programs written in C++. The Locomotion controller implements control algorithms and provides position reference for the servo motors based on the feedback signals and high level commands.

The core function of the Interconnect module is to handle communication with the gamepad and forward the user commands to the Locomotion controller using a shared memory object (SHM). Furthermore, the Interconnect module can start and stop the Locomotion controller based on the user commands. The core architecture modules are emphasized in red in Fig. 6.7. The entire control loop uses a time step of 10 ms.

Architecture add-ons for disaster response

To make Krock-2 suitable for disaster response, the robot is equipped with additional sensors, cameras and software modules providing new functionalities (Fig. 6.7). Inclusion of the force sensors gives the robot an ability to feel its immediate environment which can be used to improve its locomotion capabilities like described in following Chapters 7 and 8. The used force sensors are OMD-D30 from Optoforce, that provide 3D force measurements. The sensors are located inside an enclosure located between a foot and knee joint. An alternative would be to mount them directly on the feet. However the robot's feet orientation while in ground contact can vary a lot which might result in non sensitive parts of the sensor achieving contact. Furthermore, the enclosure provides an additional protection from wear and tear. The main drawback is reduced measurement accuracy, however as it will be seen in Chapter 7, it is sufficient to detect contacts. The accuracy problem is addressed in Chapter 8. The robot is also equipped with an IMU (Vectornav VN100) that is placed above the front girdle.

The vision system is a crucial element for Krock-2's potential niche in S&R applications - an exploration of hardly accessible areas. Thus the robot is equipped with a camera system consists of a wide angle grayscale camera (Bluefox) and a thermal camera (Seek Thermal Compact), located within a 3D printed enclosure attached to the front girdle. The Bluefox camera's controllable exposure length makes it capable of providing reasonable picture quality even in badly illuminated scenes. The thermal camera is great for detecting heat sources like humans (possible victims), however its low resolution makes it difficult to distinguish between objects in the scene with a low temperature gradient. To improve clarity, we detect and extract edges from the grayscale image of the wide angle camera and overlay them over the thermal image. A similar technique can be seen in FLIR® thermal cameras. Both cameras have a USB interface and available software libraries to interface with Linux/C++.

Due to the increased number of devices to interface with and higher computational load of image processing, a second Odroid-XU4 is added. The Odroids are enclosed within the robot's structure and are located next to the girdles. The first Odroid is located in the back, while the second is placed behind the head, making it easy to physically connect with the cameras. To reduce the number of cables passing through the spine all the sensors are connected to the nearest Odroid as shown in Fig. 6.7.

The software add-ons running on the second Odroid are Interconnect module, Camera module and Visualization module. The second Interconnect module handles the data flow between the

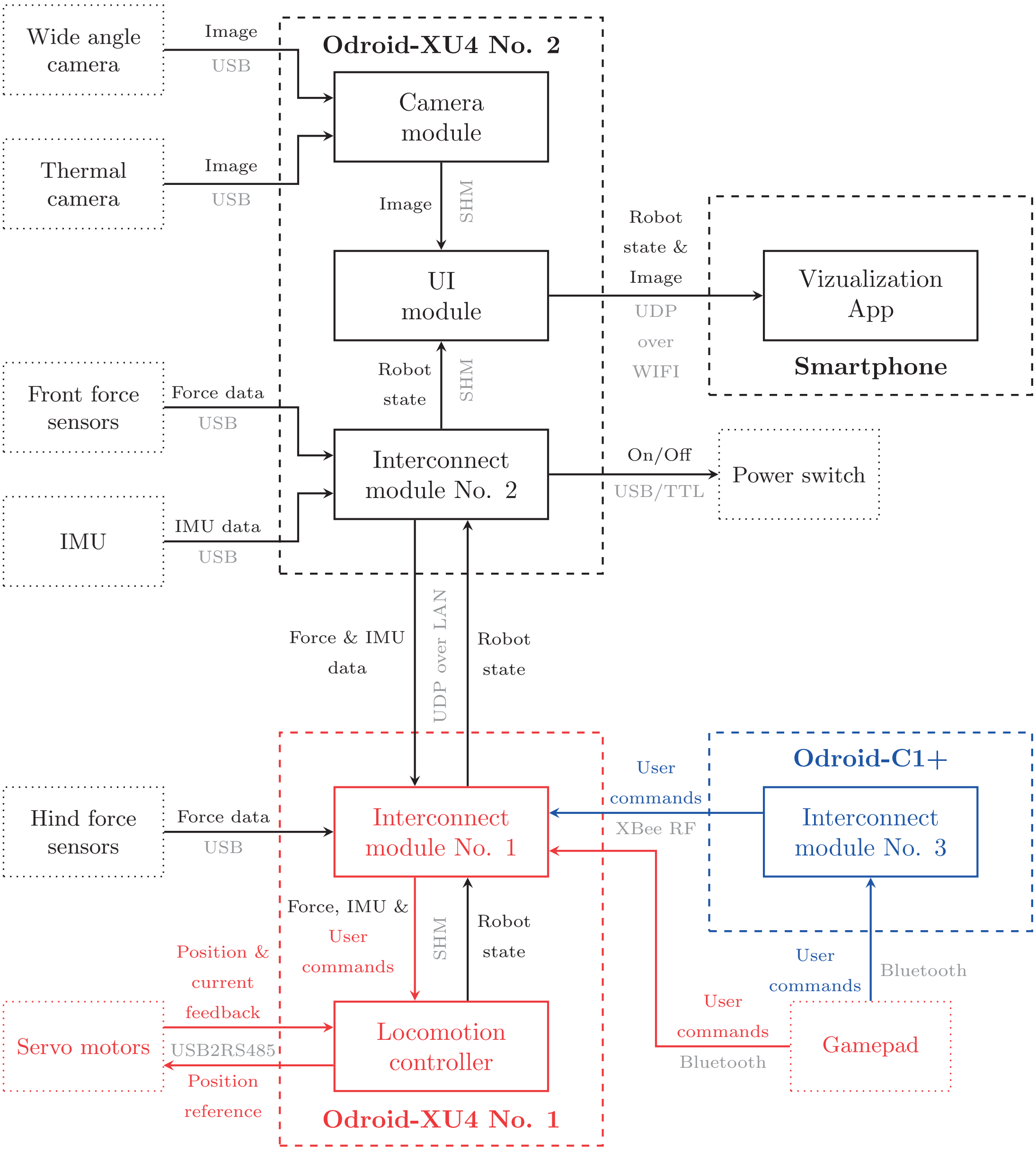


Figure 6.7 – System integration scheme of Krock robots. The core robot architecture, common for both Krock-1 and Krock-2 is shown in **red**. Its main component is Odroid-XU4 computer that runs the locomotion controller. The Krock-2 features an extended architecture (shown in **black**) that contains a second Odroid, additional sensors and accompanied visualization application for a smartphone. Krock-1 robots had an optional long range communication system based on XBee RF modules and a hand-held Odroid-C1+ (shown in **blue**). The arrows represent the data flow within the system. The **gray** labels indicate the data transfer methods.

two Odroids that are connected with a LAN cable. It also controls the power switch attached to the power line of the servo motors. The user can remotely cut the power by using the gamepad thus reducing battery drain during standby. The Camera module implements image acquisition from both cameras and aforementioned image processing. The image data is transmitted to the UI module which sends it together with the sensor data describing robot's state to a smartphone application. The transmission is done over WiFi by using a UDP protocol. The application is specially designed to visualize the robot's state and show camera feed.

Control

The Locomotion controller is based on solving the robot's inverse kinematics in order to follow desired feet trajectories. The main gait parameters like duty factor, stride length and width are predefined, however the controller adapts the feet trajectories online based on the issued user commands like walking speed or turning radius. Leg inverse kinematics is solved by using iterative Jacobian inverse method. The spine inverse kinematics and spine-limb coordination is solved like in the case of Pleurobot. Both methods are described in Chapter 3. Although Krock robots have a reduced number of spinal joints, the same control algorithm is successfully applied. The additional sensors of Krock-2 allow us to reuse attitude correction and contact based reflex control modules described in Chapter 7.

6.4.4 Long range communication system

The long range RF communication system of Krock-1 was based on two XBee-PRO® 868 modules. Despite the device specifications reporting ranges up to 1.5 km line of sight and 550 m in indoor/urban settings, we observed a considerably smaller range. However it was sufficient to cover the 100 m specification. These restrictions make a simplification in the communication mandatory. The first XBee module, mounted on the robot, connects to the Odroid-XU4 via USB-to-TTL adapter. The second module connects in the same way to a computer which serves as an input device for a robot operator. To minimize the size of such system and ensure great portability, instead of using a laptop, we built a hand-held system Odroid-C1+ Linux computer. Odroid-C1+ was preferred over more powerful Odroid-XU4 as it comes with an optional 3.5″ touchscreen. The touchscreen is used as a simple input device to control the Odroid and report basic diagnostics (e.g. communication and robot status). To issue robot commands, we reused the Bluetooth gamepad. However instead of connecting directly to the robot, it connects to the hand-held Odroid (Fig. 6.7). The gamepad commands are further sent through the RF communication link to the robot. To house the Odroid-C1+, its battery (1800 mAh LiPo @ 7.4 V), the XBee module and antenna, we built a special 3D printed hand mounted enclosure, shown in Fig. 6.8.

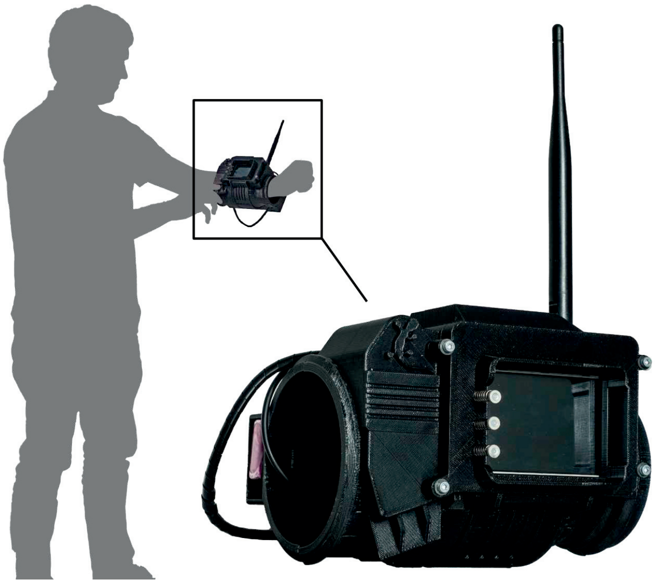


Figure 6.8 – The long range communication system. The 3D printed enclosure contains Odroid-C1+ with a touchscreen and three tactile buttons, XBee module with an antenna and a battery. Although bulky, the system can be carried around a wrist.

6.4.5 Rapid development of the user interface

An operator only provides a high level commands to the robot while the coordination of its high number of DoFs is done autonomously. Even then, it is useful for the operator to have a visual feedback on the robot's state to issue correct high level commands. This is specially true when the robot is within cluttered environment as the possibility of unwanted contacts and robot getting stuck increases. At the same time such environments can obscure the direct line of sight between the robot and its operator. This creates a need for a system that can transmit the data from robot's sensors that represent robot's state and interaction with surrounding environment to the user.

Such system should visualize the data in a user friendly manner that can be quickly interpreted. It has to be portable and easy to handle in order to avoid requiring additional manpower to operate the robot. Taking these requirements into account, using a smartphone to visualize the data seems like a good option. It is small, almost everybody has one and it can be attached to a gamepad with commercially available mounts. A drawback is the need to develop a specialized mobile application for data visualization, which might be time consuming for robotics researchers and engineers without a prior experience with such tasks. A possible alternative that we have explored is to use an existing game engine as a development tool for data visualization.

Unity game engine

The game engine we have selected for our task is Unity. Using Unity rather than developing a native mobile application brings several advantages:

- Multiplatform support and easy cross-compilation. Unity Editor is available for Windows and Mac OS while the final application can also be built for many other operating systems including Linux, Android and iOS.
- Possibility to import CAD models allows for fast and easy yet realistic construction of a virtual robot.
- Focus on visual experience. Since Unity is aimed towards game development, it contains a variety of tools for building an interactive user interface, while providing a great performance across a variety of devices.
- C# as its primary scripting language.

The interface application

The application should visually represent the data from all the available sensors: joint encoders and current sensors, IMU, feet force sensors and two cameras. A possible approach of visualization is to graphically plot time evolutions of all the sensors data, however due to its high dimension (mostly because of many servo motors), it would be impractical for a user to interpret it in real-time. The approach we take is inspired by robot simulators, however instead of building a dynamic model, we create a kinematic model representing a digital avatar of Krock-2, shown in Fig. 6.9, with the same joint configuration as the robot. To improve quality of visual representation, meshes describing different body segments are imported from the CAD drawing of the robot.

The data is linked to the avatar by using custom C# scripts. The communication between the application/smartphone and the Odroid, which should be connected to the same WiFi network, is handled by the UDP protocol. The joint encoder data is used to rotate the corresponding avatar joints. Since the reference frame is fixed to the front girdle, the avatar's orientation in the world space has to be constantly adjusted to emulate the robot's floating frame of reference, described in Chapter 3. The IMU data (roll and pitch) is used to set the global orientation of the entire avatar. The joint current, which is proportional to the joint torque is visualized by rendering curved arrows around corresponding joints. The arrow size is proportional to the magnitude of the sensed current, while its sign determines the arrow orientation. In a similar way, the feet force is represented by straight arrows attached to the avatar's feet, with their length being proportional to the magnitude of the sensed force (Fig. 6.9A). The visualization of individual sensors can be turned on or off in a simple settings menu (Fig. 6.9B).

A camera feed is visualized in a window overlaid over the main scene. A user can chose between

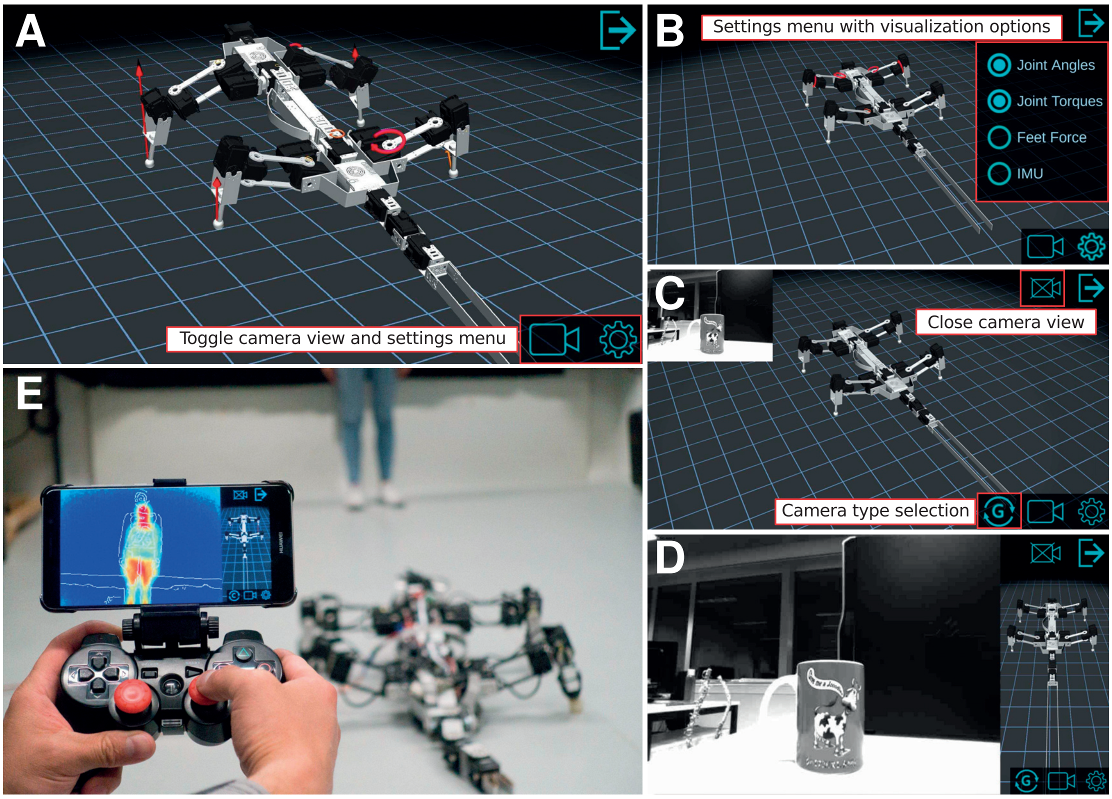


Figure 6.9 – The user interface application made in Unity. (**A**) The visualization of robot's encoder, joint torque, IMU and contact force data. The arrow size is proportional to the force and torque magnitude. (**B**) The settings menu used to toggle the visualization of the individual sensors. (**C**) A small camera window showing the feed from the grayscale camera. (**D**) A large camera window covering most of the screen. (**E**) A smartphone running the interface application, mounted on a gamepad used to control the robot. The application is visualizing the thermal camera feed.

two different sizes of the camera window (Fig. 6.9C and D) or turn it off completely (Fig. 6.9A). The camera data can be toggled between the thermal image, wide angle grayscale image or the combined view where the edges from the grayscale image are placed on top of the thermal image (Fig. 6.9E).

7 Reflex Based Controller for Walking on Uneven Terrain

The locomotion controller used so far assumes a robot is walking on flat ground. The utilized control methods are not relying on any sensory feedback that contains information about robot interaction with an environment. This is a problem when walking over rough or unstructured terrain as unperceived irregularities can hinder motion of the robot. Here we tackle the problem by introducing new control modules — reflex and attitude controllers — that adapt the robot's foot trajectories based on sensory feedback. The in-depth analysis of the new control scheme is made on Pleurobot, however the reflex module is also demonstrated on Krock-2.

Parts of the material presented in this chapter are adapted from:

[84] Horvat, T., Karakasiliotis, K., Melo, K., Fleury, L., Thandiackal, R., and Ijspeert, A. J. Inverse kinematics and reflex based controller for body-limb coordination of a salamander-like robot walking on uneven terrain. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (2015), IEEE, pp. 195–201.

My contributions: Designed the controller and programmed the robot, designed and carried out the experiments, writing.

7.1 Introduction

Developing and controlling robots for search and rescue applications is becoming a hot topic in the robotics community. Search and rescue scenarios usually involve heavily cluttered, on the verge of collapse (e.g. after an earthquake) or even flooded areas which are unsafe and not easily reachable for human rescuers. That is why the use of robots is necessary. It can shorten the response time of the rescue team and increase their efficiency.

However, not all robots are suitable for the task. The suitable candidates should be able to handle rough terrain and have a low profile in order to be able to reach cluttered areas and small holes. In addition, swimming capabilities might be useful in case of flooded areas. One robot matching all the criteria is RHex [172]. Its robust design, rotational legs (whegs) and the capability to continue the locomotion after flipping over make it a highly versatile robot [138]. It can also swim, but for fast swimming it requires a human operator to exchange its whegs with fins.

Another class of robots gaining popularity within the search and rescue community are snake-like robots. In [125] and [121], researchers are exploring snake robots that can be easily deployed into a collapsed building and navigate through cluttered environments, making use of its shape and low profile. The snake robot ACM-R5 successfully handles swimming due to its waterproof design [82]. Moreover, it has passive wheels which in combination with snake-like movements allow it to move over flat terrain. Salamander-like robot Salamandra Robotica II [33] combines segmented spine with whegs. The robot has good swimming and basic ground locomotion capabilities.

The next iteration of the salamander-like robot, Pleurobot, mixes interesting features: segmented spine, segmented legs, low center of mass, sprawling posture, ability to quickly reconfigure the robot (attach/detach the tail) and it is able to use a wide range of different gaits (e.g. low posture or high posture). While the robot was initially designed as a tool for neuroscience in particular, to test hypotheses about the interconnection of neural circuits controlling locomotion in the salamander spinal cord, we think it has also an interesting potential for search and rescue applications. For such applications, a key requirement is the ability to tackle rough terrains. Ajallooeian *et al.* [1] explored the problem of dynamic locomotion of a quadruped robot over unperceived rough terrain. The proposed framework involves the use of fast reflexes and adjustments of the robot's attitude. Similar reflex based strategy was used on a HyQ robot [53].

In this chapter, we address the question of how to design the reflex and attitude controller modules to improve the locomotion over uneven terrain. We did a series of tests over different terrains and with different robot configurations (tailed vs. tailless) to evaluate the performance of the proposed control framework and see the effect of each of the control modules.

7.2 The setup

Pleurobot is a quadruped salamander-like robot with 27 active degrees of freedom, 11 in the spine and 4 per each leg (Fig. 7.1). The spine can actively bend in the horizontal plane, while the tail has 2 passive degrees of freedom in a vertical plane (one joint behind the hind girdle and one before a fin). For actuation, it uses Dynamixel MX64 servo motors (max torque of 7.3 Nm). Body parts are 3D printed in polyamide strengthened with fiberglass (spine segments) or aluminum (leg segments).

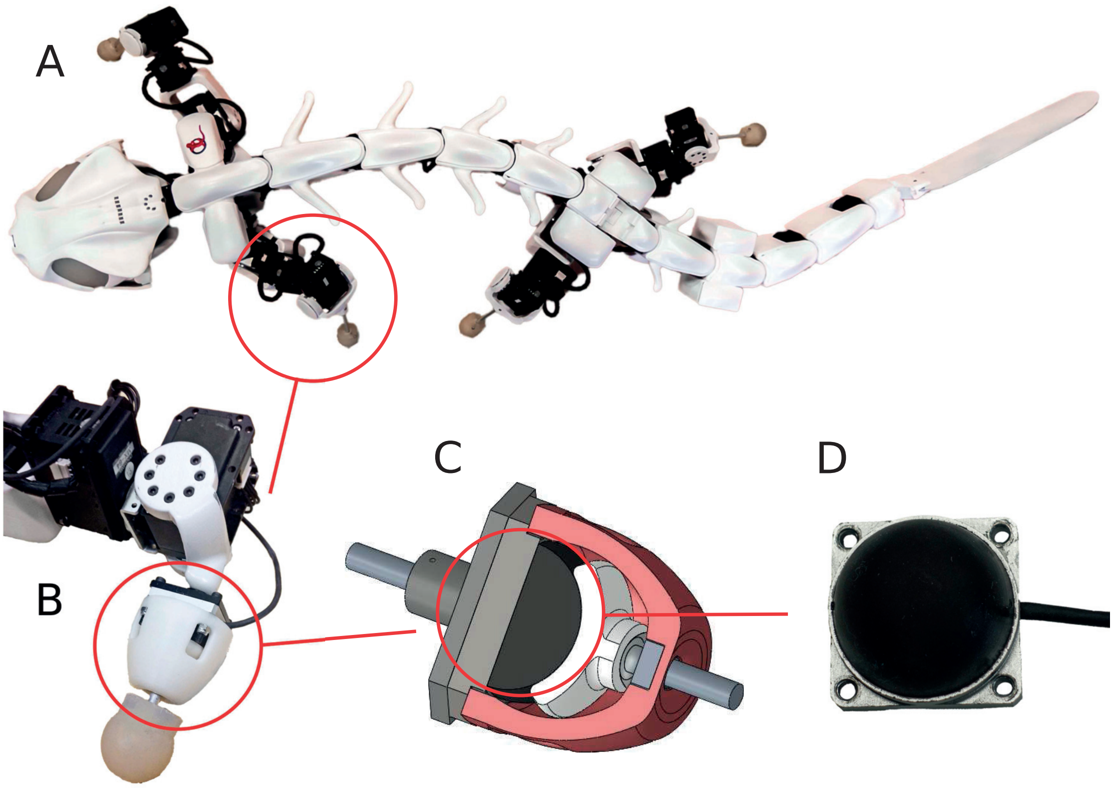


Figure 7.1 – The force sensor setup. (**A**) Pleurobot. (**B**) 3D printed casing including the sensor, mounted between the knee and the foot. (**C**) The schematics of the casing. The force is transmitted to the sensor by a rod with a socket on its end. (**D**) A dome-shaped Optoforce sensor.

Between the knee and the silicone ball foot, lies a 3D force sensor (Optoforce). The sensor relies the force estimation on a deformation of the silicone ball (Fig. 7.1). It can be mounted directly on the foot, replacing the silicone ball, but we chose to put it in a casing in order to prevent abrasion of the sensor's surface and increase the sensing area. In this configuration, we only use the magnitude of the force readings (i.e. 1 dimension), but not the directional information.

While locomoting, the robot uses sprawling posture, occasionally touching the ground with its hips. The tail is constantly being dragged on the ground. However, the tail can be easily and quickly removed (within a few minutes), allowing a higher posture while walking, but removes the swimming capability. In this paper, we will compare walking performance with and without the tail. The goal is to have the same controller architecture for both configurations, and modify its morphology at the start of a mission, depending on the terrain.

7.3 Reflexes and attitude controller

Due to its low center of mass and the sprawling posture, Pleurobot can handle rough terrain even with a basic locomotion controller, i.e. with the leg and spine controllers generating trajectories in open-loop. To improve its performance, we introduce a set of reflexes based on force feedback from the feet and an attitude controller based on the IMU (inertial measurement unit) readings placed on the front girdle.

7.3.1 Leg extension reflex

Predefined leg trajectories assume that the ground beneath the robot is flat. While walking over rough terrain, that assumption is no longer valid and some of the legs could lose contact during the stance phase. That could result in the robot hitting the ground with its hips or even losing balance while having gaits with a higher posture. Inspired by [37], we introduce a leg extension reflex $\boldsymbol{r}_{\text{ext},i}$ which modifies the i-th foot trajectory $\boldsymbol{\xi}_i$ based on the feedback from the attached force sensor. If the sensed force in the stance phase is lower than a predefined threshold (missing contact), the foot is being extended downwards (in the direction of negative z axis) until the force readings reach the same threshold (the foot is in contact with the ground). The reflex trajectory is described as a first order linear dynamical system with different time constants $T_{\text{ext,ON}}$ and $T_{\text{ext,OFF}}$ for the activation and the deactivation of the reflex:

$$\dot{\boldsymbol{r}}_{ext,i} = \begin{cases} -\dfrac{\boldsymbol{r}_{\text{ext},i}}{T_{\text{ext,ON}}} + \dfrac{\boldsymbol{\zeta}_{\text{ext}}}{T_{\text{ext,ON}}} & \text{,reflex ON} \\ -\dfrac{\boldsymbol{r}_{\text{ext},i}}{T_{\text{ext,OFF}}} & \text{,reflex OFF} \end{cases} \tag{7.1}$$

If the reflex is triggered, a constant input $\boldsymbol{\zeta}_{\text{ext}} = \begin{bmatrix} 0 & 0 & -\Delta z \end{bmatrix}^T$, is fed to the dynamical system where $\Delta z > 0$ determines the maximum leg extension.

7.3.2 Stumbling correction reflex

Another problem to overcome while traversing rough terrain is hitting an obstacle with a leg that is in a swing phase. If the robot is in a sprawling posture, even a relatively low obstacle, which is higher than the leg's swing trajectory, cannot be traversed. While in a higher posture, stiff contact and pushing against an obstacle can cause the robot to lose balance. Inspired by [56], we introduce a stumbling correction reflex which, like the leg extension reflex, modifies the foot trajectory. If the sensed force during swing is higher than a predefined threshold (contact with an obstacle), the foot retracts backwards (the direction of negative x axis) and reaches up (the direction of positive z axis). The backwards retraction prevents the leg from pushing against the obstacle, and increasing the height of the swing allows the leg to overpass the obstacle. The reflex trajectory has the same description as the one for the leg extension reflex, with the activation and deactivation time constants $T_{\text{stu,ON}}$ and $T_{\text{stu,OFF}}$:

$$\dot{\boldsymbol{r}}_{\text{ext},i} = \begin{cases} -\dfrac{\boldsymbol{r}_{\text{stu},i}}{T_{\text{stu,ON}}} + \dfrac{\boldsymbol{\zeta}_{\text{stu}}}{T_{\text{stu,ON}}} & \text{,reflex ON} \\ -\dfrac{\boldsymbol{r}_{\text{stu},i}}{T_{\text{stu,OFF}}} & \text{,reflex OFF} \end{cases} \tag{7.2}$$

The only difference from the leg extension reflex is the constant input fed into the system while the reflex is turned on: $\boldsymbol{\zeta}_{\text{stu}} = \begin{bmatrix} -\Delta x & 0 & \Delta z \end{bmatrix}^T$, where $\Delta x > 0$ and $\Delta z > 0$ determine how much the leg retracts and reaches up.

7.3.3 Attitude controller

Traversing an obstacle or stepping on a stair can lead to significant pitching and rolling of the body, even with available reflexes. If the obstacle is too high or the robot has taken a higher posture with a lower intrinsic stability, it can significantly affect the robot's balance. To tackle the problem, we introduce a simple feedback rule on the roll ϕ_{roll} and pitch ϕ_{pitch} angle readings from the IMU. The angles are represented with respect to the robot's body frame. First, we pass the readings through a first order dynamical system, which acts as a low pass filter:

$$\begin{aligned} \dot{\phi}_{\text{filt,roll}} &= -\frac{\phi_{\text{filt,roll}}}{T_{\text{roll}}} + \frac{\phi_{\text{roll}}}{T_{\text{roll}}}, \\ \dot{\phi}_{\text{filt,pitch}} &= -\frac{\phi_{\text{filt,pitch}}}{T_{\text{pitch}}} + \frac{\phi_{\text{pitch}}}{T_{\text{pitch}}}, \end{aligned} \tag{7.3}$$

where T_{roll} and T_{pitch} are filter time constants. Based on the filtered pitch and roll angles, we calculate modifications $\boldsymbol{\kappa}_i$ to the leg trajectories as follows:

$$\begin{aligned} \kappa_{1,z} &= k_{\text{roll}} \cdot \phi_{\text{filt,roll}} + k_{\text{pitch}} \cdot \phi_{\text{filt,pitch}}, \\ \kappa_{2,z} &= -k_{\text{roll}} \cdot \phi_{\text{filt,roll}} + k_{\text{pitch}} \cdot \phi_{\text{filt,pitch}}, \\ \kappa_{3,z} &= k_{\text{roll}} \cdot \phi_{\text{filt,roll}} - k_{\text{pitch}} \cdot \phi_{\text{filt,pitch}}, \\ \kappa_{4,z} &= -k_{\text{roll}} \cdot \phi_{\text{filt,roll}} - k_{\text{pitch}} \cdot \phi_{\text{filt,pitch}}, \end{aligned} \tag{7.4}$$

where k_{roll} and k_{pitch} are tunable gains. Notice that we only modify the z component of the trajectory: $\boldsymbol{\kappa}_i = \begin{bmatrix} 0 & 0 & \kappa_{i,z} \end{bmatrix}^T$.

Finally, we combine reflexes and attitude controller to get a modified leg trajectory $\boldsymbol{\xi}_{\text{mod},i}$, which is then used in the inverse kinematics controller to calculate the joint angles:

$$\boldsymbol{\xi}_{\mathrm{mod},i} = \boldsymbol{\xi}_i + \boldsymbol{r}_{\mathrm{ext},i} + \boldsymbol{r}_{\mathrm{stu},i} + \boldsymbol{\kappa}_i. \tag{7.5}$$

Fig. 7.2 shows how the new modules are added to the basic locomotion controller (BLC) described in Chapter 3. The internal BLC structure remains unchanged as the only way the reflex and attitude controllers modify its behavior is by giving an extra input to the foot trajectory generator.

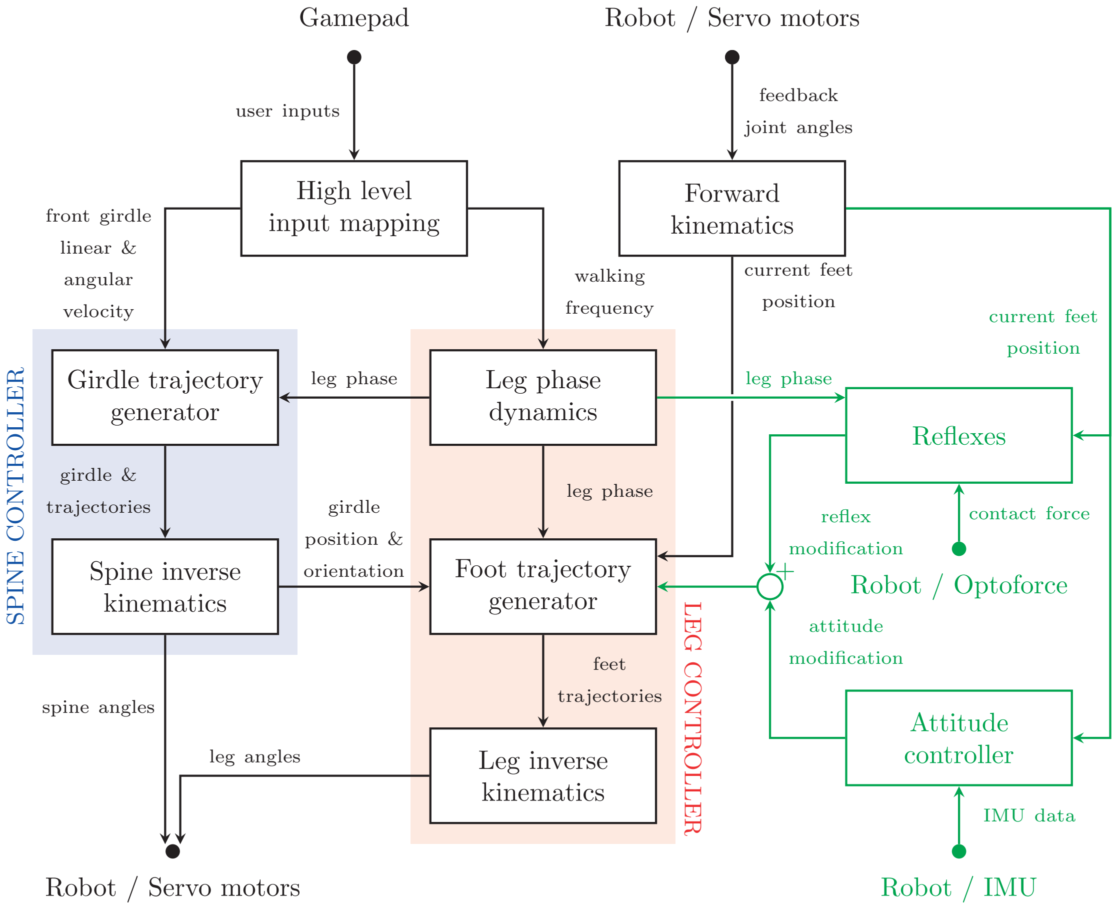


Figure 7.2 – Reflex and attitude modules (green) added to the basic locomotion controller. The new modules modify the foot trajectories. An additional feedback, required by the expanded controller, is provided by the Optoforce sensors and the IMU.

7.4 Experiments

7.4.1 Simulation

The robot is being modeled and simulated in Webots™ robotics simulator. The control loop and physical simulation use a time step of 4 ms.

We evaluate robot's performance by letting it walk over 3 different terrains (*stairs*, *holes* and *rough terrain*) with 3 difficulty levels (*easy*, *medium*, *hard*), measuring its success rate (it did not flip over and, in case of the *stairs*, robot climbed on the first stair) and comparing forward velocity to its flat ground velocity while utilizing the same gait. The obstacles and irregularities are parameterized according to robot size and its ground clearance as described in the following.

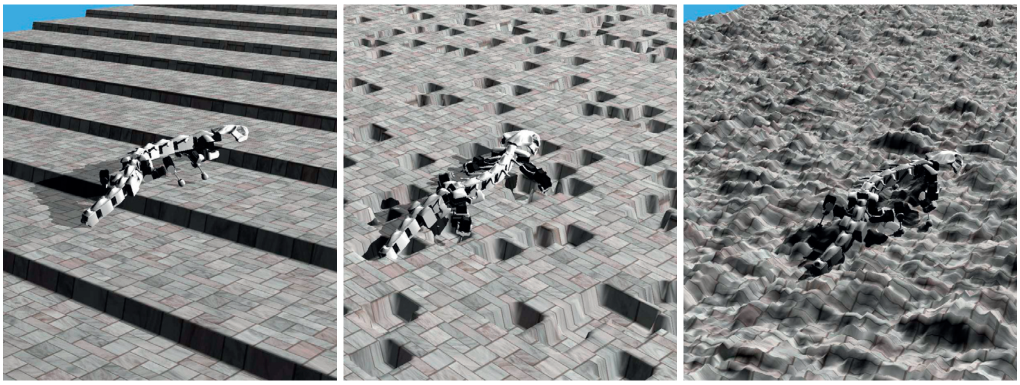


Figure 7.3 – The terrains used in the simulation experiments: *stairs* (left), *holes* (middle) and *random* terrain (right).

The task in the *stairs terrain* is to climb the stairs. The length of a single stair is 70 cm, which is approximately the length of Pleurobot without its tail. To vary difficulty, the stair height was changed: 5 cm for *easy*, 7.5 cm for *medium* and 10 cm for *hard* difficulty. The numbers correspond to 50%, 75% and 100% of the robot's lower arm length. We repeat the experiment 100 times with different initial conditions (distance to the first stair).

The *holes terrain* consists of randomly placed square-shaped holes of a specific depth: 5 cm for *easy*, 7.5 cm for *medium* and 10 cm for *hard* difficulty. The experiment is repeated 100 times, where we varied the robots initial position in both lateral and longitudinal direction.

The *random terrain* is created by initializing its grid height with random values coming from a normal distribution of a certain variance: 10 cm for *easy*, 30 cm for *medium* and 50 cm for *hard*. The terrain surface is then smoothened with a 2D Gaussian filter, decreasing the difference between the lowest and highest points. The experiment is repeated 100 times, where we varied the robots initial position in both lateral and longitudinal direction.

In the experiments[1] Pleurobot had two different configurations: with and without tail. The tailed version used a low posture gait with a stride length of 42 cm. The heights of the front and hind girdles were 13 cm and 15 cm respectively. Those parameters were hand-tuned in order to obtain natural looking gaits, similar to the gait that the salamander uses [97]. The tailless version uses higher posture with height of both girdles set to 20 cm and stride length of 40 cm. In both cases duty factor was set to 0.75 and the walking frequency was 0.3 Hz.

Parameters of the reflexes and the attitude controller were hand-tuned on a single stair experiment until we have got a stable behavior. Once tuned, the parameters were kept constant during all the experiments. Their values are shown in Table 7.1.

Table 7.1 – Reflex and attitude control module parameters.

Extension reflex		Stumble reflex		Attitude control	
$T_{\text{ext,ON}}$	0.5 [s]	$T_{\text{stu,ON}}$	0.033 [s]	k_{roll}	0.1 [m/rad]
$T_{\text{ext,OFF}}$	2.31 [s]	$T_{\text{stu,OFF}}$	1.6665 [s]	k_{pitch}	0.1 [m/rad]
$\boldsymbol{\zeta}_{\text{ext}}$	$\begin{bmatrix}0 & 0 & -0.15\end{bmatrix}^T$ [m]	$\boldsymbol{\zeta}_{\text{stu}}$	$\begin{bmatrix}-0.15 & 0 & 0.2\end{bmatrix}^T$ [m]	T_{roll}	0.2 [s]
				T_{pitch}	0.2 [s]

7.4.2 Hardware experiments

The reflex behavior was tested on the real robot with both configurations: tailed and tailless version. As in a simulation, the tailless version used a higher posture gait. The gait parameters (except for the height of the tailed version, which was 2 cm higher to compensate for the extra compliance), were the same as in the simulation. The time step of the control loop was set to 10 ms.

Due to the limitations of the platform and building the rough terrain, the robot was tested in simplified scenarios: a single obstacle, a single stair and a hole. The height of the stair/obstacle and the depth of the hole was approximately 7.5 cm which corresponds to the *medium* difficulty of the terrain used in the simulation.

7.5 Results

Fig. 7.4 shows the simulation results for the tailed version for all three terrains. The *Easy stairs* terrain takes advantage of the robot's low posture and high stability since the swing of the leg is higher than the height of a single stair. However, in the *Hard stairs* terrain, the BLC completely fails since it did not overcome the first stair. The reflexes are compromising the stability and causing lower performance and success rate (robot flipped over). The attitude controller counteracts the bad property of the reflexes (the leg extension reflex specially) and

---

[1] The video of the experiments can be fount at: https://biorob2.epfl.ch/video/310

increases the overall performance and the success rate.

In the *holes terrain* adding the reflexes and the attitude controller increases the overall performance. The attitude controller also increases the success rate of the reflexes. In the *random terrain* the attitude controller did not improve the performance of the reflexes. This shows that we need an attitude controller mostly when the robot's pitch and roll are high in order to prevent falling.

Fig. 7.5 shows the results of the tailless version. We observe similar effects of both reflexes and reflexes + attitude controller. Although the performance of the basic controller is the lowest, it still has a high success rate in the *holes* and *random* terrains, which emphasizes the advantage of the platform for the rough terrain locomotion due to its high stability. Compared to the tailed version, overall on the *high* difficulty the tailless robot has higher performance.

Experiments on the real robot confirmed our insights from the simulation results. The obstacle and the stair experiment showed the importance of the stumble reflex. Without it, the robot got stuck and could not traverse the obstacle or start to climb the stair (Fig. 7.6). Even with the reflexes, the tailed version could not lift its hind girdle over the stair due to the current limitations of the platform: high load around the hind girdle, torque limits of the used servo motors, sharp edges of the hips and battery holder which can easily get stuck against the irregularities in the terrain. The tailless version, with the high posture gait could climb up the stair, but it was pitching significantly while doing so. The leg extension reflex in this case increased the pitching (front legs were activating it) so we kept it turned off. Based on the simulation results we expect it to be improved with the attitude controller. The extension reflex helped the robot move over the hole. With the reflex, the leg reached the ground and provided extra push, while without it, the leg had a stance phase in the air (Fig. 7.7).

As in simulation, the performance of the tailless version was higher as the tail does not actively help the ground locomotion, yet it only provides an extra weight. The reduced weight allows the robot to use a gait with a higher posture. Higher ground clearance made it easier to traverse higher obstacles, which was particularly visible in the hardware experiments.

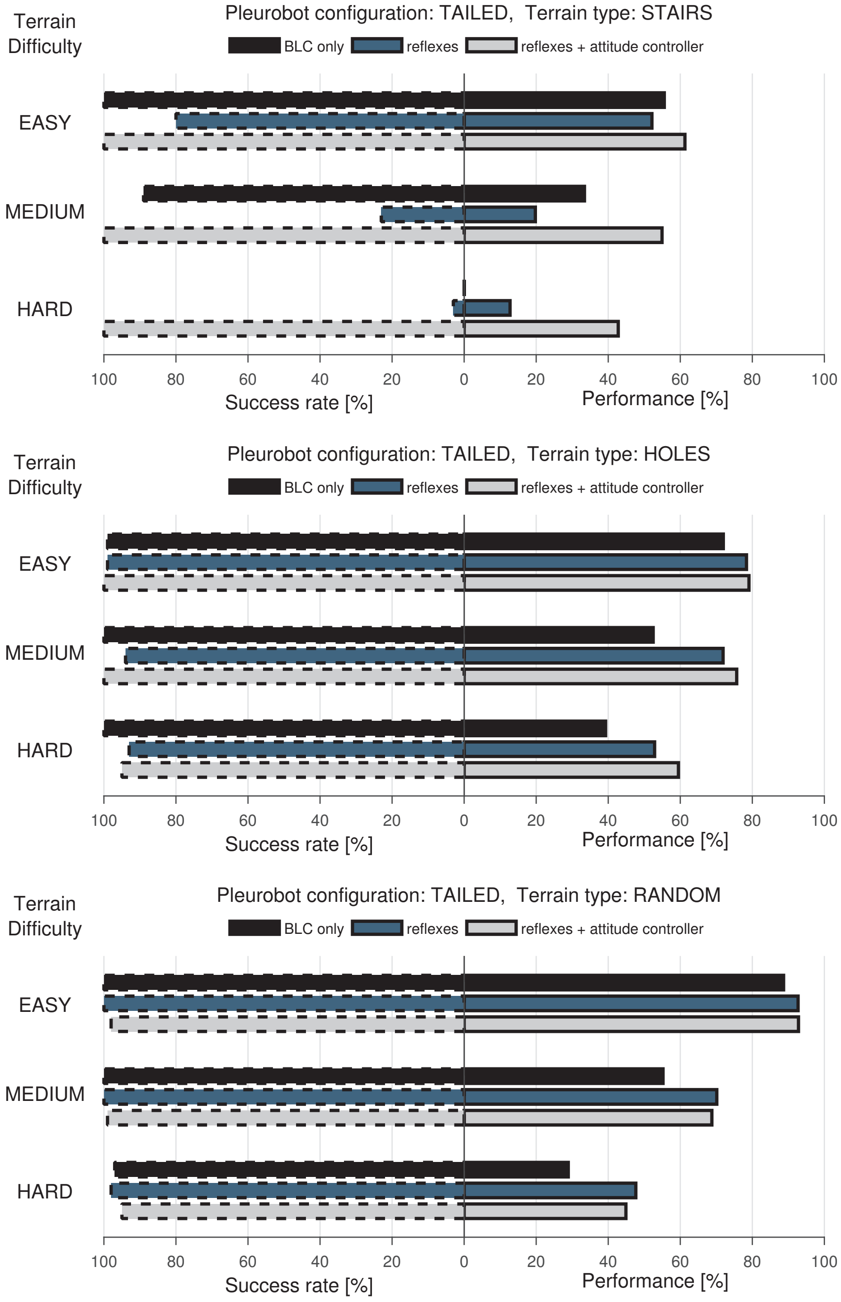


Figure 7.4 – Performance of the *basic locomotion controller (BLC)*, *BLC + reflex* and *BLC + reflex + attitude controller* for the tailed version of Pleurobot over *stairs* (up), *holes* (middle) and *random* terrain (down). For the easy difficulty, BLC provides great performance and high success rate. How the difficulty increases, the advantage of reflexes and attitude controller becomes more visible. The attitude controller has the highest impact in the *stairs* terrain, where it is crucial to control the pitch angle of the robot when the front legs step on the next stair and the hind legs are still on the previous one. The performance is expressed as the ratio between the robot's walking velocities achieved on the rough terrain and a flat ground (while using the same gait).

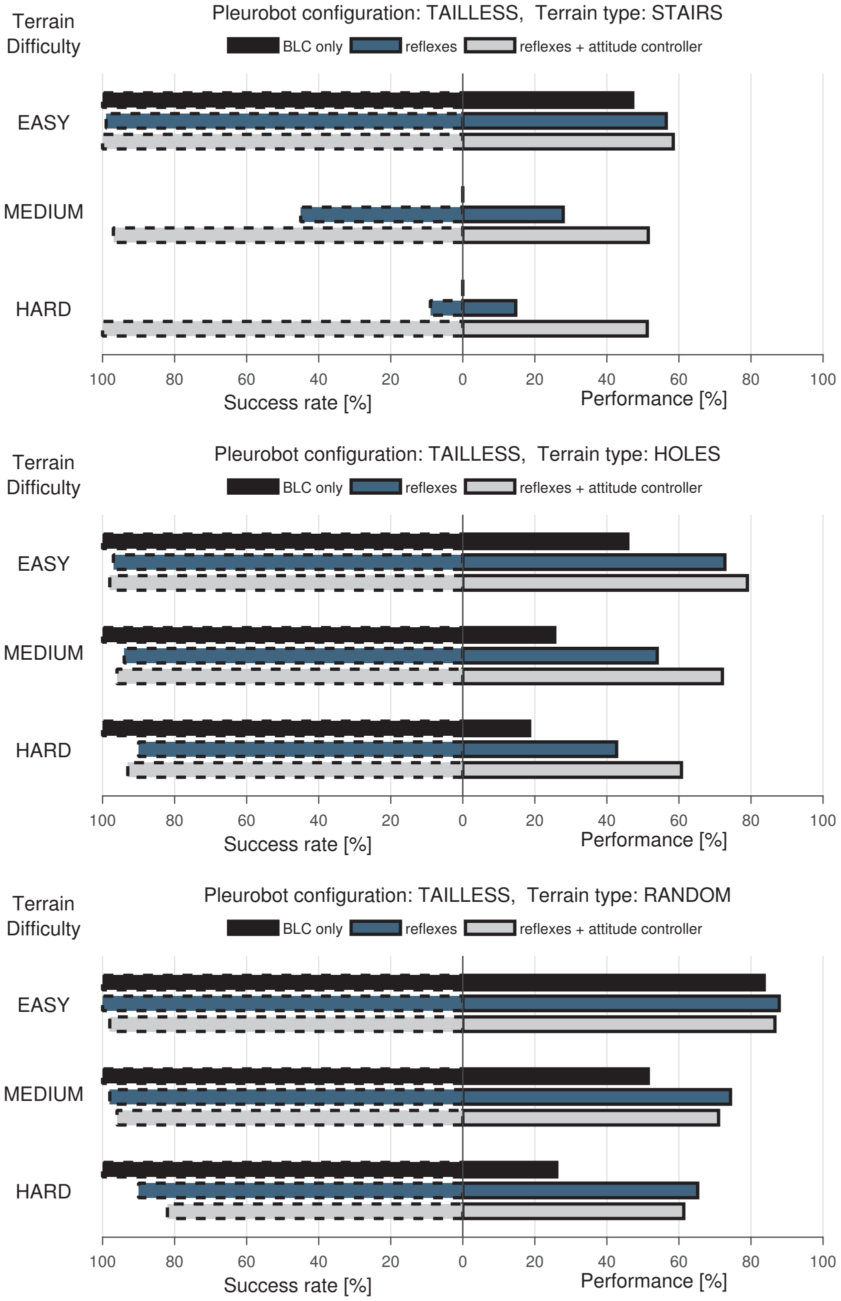


Figure 7.5 – Performance of the *basic locomotion controller (BLC)*, *BLC + reflex* and *BLC + reflex + attitude controller* for the tailless version of Pleurobot over *stairs* (up), *holes* (middle) and *random* terrain (down).

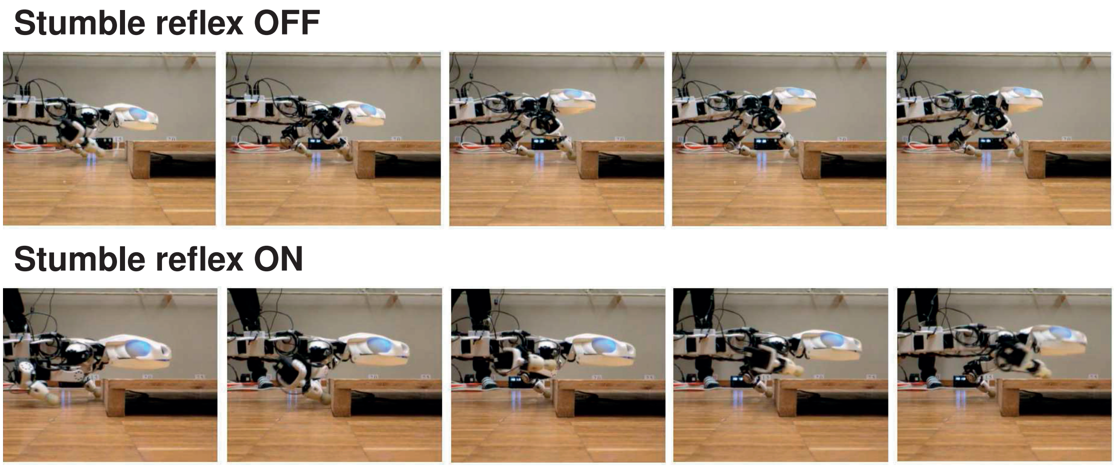


Figure 7.6 – The stumbling reflex experiment with the tailless configuration of Pleurobot. The snapshots show approximately a one cycle of the front leg stepping on the stair. Without the reflex (up) a swing is not high enough and the leg does not reach over the stair and it pushes the robot back. With the reflex (down) the leg trajectory is modified and the leg reaches over the stair.

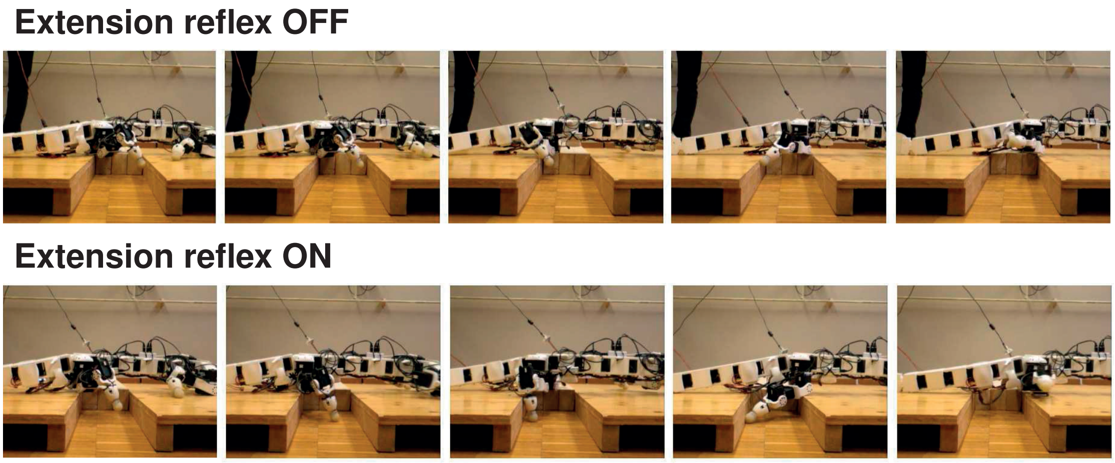


Figure 7.7 – The leg extension reflex experiment with the tailed configuration of Pleurobot. The snapshots show approximately a one cycle of the hind leg walking over the hole. Without the reflex (up) the leg does not reach the bottom of the hole and executes the stance phase in the air. With the reflex (down) the leg extends until it reaches the ground.

7.6 Porting the reflex module to Krock-2 robot

The reflex control module was originally designed for Pleurobot, to improve its rough terrain locomotion performance. In this section we demonstrate[2] it with the Krock-2, which primary design was to be a field-ready robot. Krock-2 is equipped with the same force sensors and the same mounting mechanism as Pleurobot. Fig. 7.8 shows simulation experiments with different combinations of enabled reflexes and gait variations. The height of manually placed cuboid obstacles is 15 cm. Without the stumbling reflex the robot cannot step over the obstacle and bounces against it (Fig. 7.8A). The shoulder joint configuration of Krock-2 enables it to use a greater swing height which helps it to step on the obstacles (Fig. 7.8B). However, without the leg extension reflex, the robot gets stuck once its body is placed on the obstacle because the stance height is significantly lower than the obstacles. Similar behavior is observed when only the stumble reflex is enabled (Fig. 7.8C). When using both reflexes, the robot had the least amount of deviation from the original walking direction. The swing height and other gait parameters across experiments shown in A, C and D parts of Fig. 7.8 are kept the same.

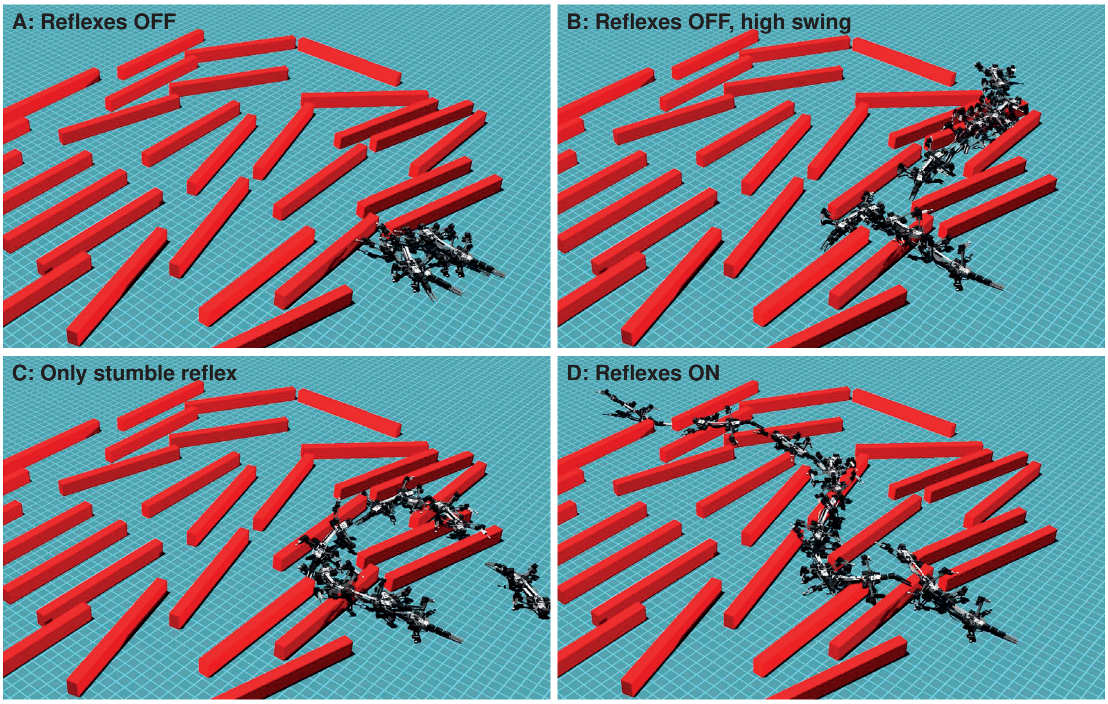


Figure 7.8 – Simulation experiments of Krock-2 walking over obstacles for four different combinations of enabled reflexes and gait variations. The overlaid robot snapshots are taken every 8 s.

---

[2] The video of the experiments can be fount at: https://biorob2.epfl.ch/video/308

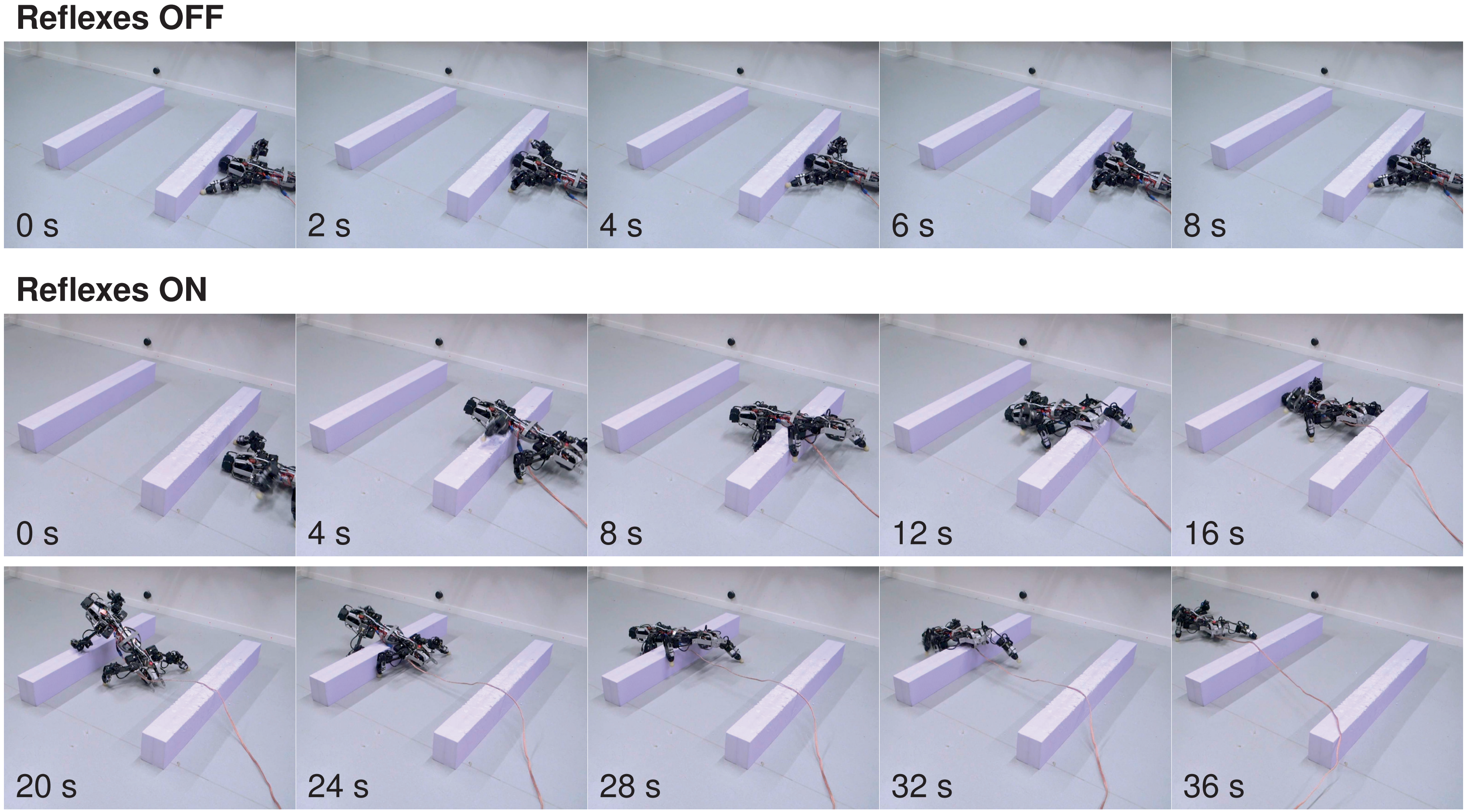


Figure 7.9 – Hardware experiment of Krock-2 walking over obstacles, with and without the reflexes.

A smaller scale experimental setup containing two obstacles, shown in Fig. 7.9, is built for the real robot. The obstacle height is the same as in the simulation - 15 cm. Without the reflexes or an exaggerated swing height, the robot cannot step over the obstacle. With the reflexes the robot successfully completed the course by traversing both obstacles. This experiment showed that the biggest challenge for the robot is to get its hind feet on the obstacle, for which it took several attempts.

7.7 Discussion and conclusion

We presented control modules that serve as augmentations of the basic locomotion controller used by a sprawling posture robot. The reflex module implements a set of reflexes whose activation is based on feet contact information. Their purpose is to rapidly modify feet trajectories. If an unwanted contact (an obstacle) occurs during a leg swing phase, the stumbling reflex adjusts the foot trajectory in order to attempt stepping on or over the obstacle. If the contact is missing during a leg stance phase, the leg extension reflex moves the foot trajectory towards the expected ground plane. The attitude control module uses the measurements from the IMU attached to the robot and modifies foot trajectories in order to decrease the rolling and pitching of the robot's body.

The influence of added modules on Pleurobot's locomotion performance over different variations of rough terrain was systematically assessed in the simulation. The results showed how added modules improve the robot's speed and success rate (not flipping over) in most of the cases. The reflexes were also tested on the real robot, demonstrating the effects of the each individual reflex. Furthermore, the reflex module was ported to Krock-2 robot and tested in a scenario involving obstacles whose height was more than 50% of the robot's leg length. With the reflexes enabled, the robot successfully traversed the obstacles. The use of sprawled posture prevented it from flipping over even though high body pitching and rolling was observed while climbing the obstacles.

These control modules proved to be a simple yet effective way to rapidly adapt robot gait to the terrain, relying on a sensory feedback. The accuracy of force sensors does not need to be high, as they are used only to detect a contact, for which the force thresholds are adjusted manually.

7.7.1 Future work

To tackle even more difficult terrains (higher obstacles) we would need to have a more sophisticated interplay between the reflexes and the attitude controller which are, for now, completely separated. For example, this would involve a state machine which would disable the extension reflex for certain legs if the roll or pitch angles get too high to prevent counteraction between reflexes and the attitude controller. Another possible enhancement is to actively use the tail during the locomotion, which is something that lizards do dynamically [23]. It could be used to give an extra push if the robot gets stuck or it is slipping too much. To allow this, a reliable forward velocity estimation is required. The attitude controller can be upgraded with the posture controller, which would also control the robot center of mass.

8 Crawling in Cluttered Environments and Pipes

The previous chapter showed how to improve robot's locomotion over rough terrain, often found in disaster response missions. However, the environments present in such missions might not be traversable by using a standard walking gait. Here we introduce a crawling gait designed to help Krock-2 move through narrow spaces like pipes. We describe several new control and estimation modules needed for the task.

8.1 Introduction

A big challenge in search and rescue robotics is coping with an environment which is potentially unstructured and cluttered, as a result of a disaster (earthquake, flood, explosion etc.). Such an environment could be partially or completely unmapped, requiring robots to adapt their locomotion to surrounding conditions. As discussed in the Chapter 6, what rescuers need are tools to investigate small cracks and openings that are not accessible for humans. Due to its low profile, Krock-2 is suitable for such search and exploration oriented tasks. Although its flat design allows it to easily enter low but wide openings, anything too narrow for its regular walking gait remains inaccessible for the robot. Other than openings and pockets in rubble, such environments could also be more structured, like pipes or ducts. To access them, Krock-2 has to utilize unconventional crawling gait pushing itself against obstacles or walls.

Most of related research focused on building robotic platforms that are specifically designed for a pipe crawling task. Those robots can generally be grouped into wheeled (or tracked) and legged robots. The wheeled robots rely on maintaining contact between wheels and the pipe's walls [132, 203, 171]. For us, more interesting are legged robots. In [178] researchers describe the design of a spider like quadruped robot for planar locomotion inside tunnels. Its design and controller are based on immobilization theory [163], allowing it to move without relying on friction. However, such approach requires a precise map of the environment. Similar spider-like robot design is considered in [134], relying on simpler reflexive and reactive control scheme. An 8-legged pipe crawling robot that relies on precise contact force control is described in [201]. Due to its segmented, actuated spine, it can advance through pipe junctions and take sharp turns.

Robots that are not specifically engineered for pipe locomotion, but are generally capable of it, are the snake robots. Due to their slim bodies, such robots can easily enter small openings making them fit for exploration of cluttered terrain. Examples of snake robots moving inside pipes can be found in [200] or [167]. The snake robot in the former example is capable of vertical climbing inside a pipe.

Our approach is to use the already existing quadruped robot Krock-2 and adapt its control to make it capable of pipe crawling, thus expanding its multi-modal locomotion capabilities. To the best to our knowledge, Krock-2 is the only legged robot suitable for open ground walking and pipe crawling, without being specifically designed for the latter task.

8.2 Problem statement

The considered scenario is crawling through places which are too narrow to a standard walking gait (Fig. 8.1B). To simplify the problem, the scenario will involve moving through pipes or between two closely placed walls. From the control point of view, both of these two options are equivalent, so from now on we will refer to the problem as a "pipe locomotion problem". We assume that a too limited vertical space in the pipe (height) is not an issue, which holds true for pipes with circular cross-section (i.e. the diameter is not too small). Furthermore, we allow pipes to be placed on a slope and have turns.

There are several ways Krock-2 could potentially locomote inside a narrow pipe, as illustrated in Fig. 8.1:

- Use a high posture gait with feet underneath the body (Fig. 8.1C). Although simple to execute as it only requires a modification of feet trajectories, the balancing of the robot would be poor, due to a high center of mass and a narrow support polygon (especially if the bottom of the pipe is also curved, or the pipe is located on a slope).
- Use a low posture with longitudinally stretched legs (Fig. 8.1D). Like the previous option, it only requires a modification of feet trajectories, without additional changes of the locomotion controller. However, fully stretching the legs would result in reduced manipulability of a foot position, due to singularities in the limb configuration. In such a configuration, the 4 DoF limb can effectively use only two DoFs (one DoF in the shoulder and a second one in the knee) making it difficult to control the end effector trajectory.
- Allow the body to rest on the ground and push against the pipe wall while keeping the feet above the body (Fig. 8.1E). Although it does not follow the KISS principle[1], such approach would allow the robot to take steeper slopes by exploiting the friction between the body and the pipe, as well as between the feet and the pipe. In this case balancing is not an issue since the robot is already lying on the ground.

Going through tight spaces makes undesired contacts between the robot's body and an environment impossible to avoid. Such contacts would hinder the walking gaits described in first two points above. Therefore, we will consider the third, more challenging but also more fitting approach, where the robot fully exploits the constrained environment it is located in. We will henceforth refer to this type of gait as pipe crawling.

The questions to be broken down and answered are as follows:

- What is the sequence of actions the robot has to make in order to move through a pipe? Pushing against walls is a very different action compared to a normal walking gait. It requires a different state machine that is switching between leg's stance phase (a foot is in contact with a wall) and a swing phase (no contact) as well as a different coordination between legs.
- How to estimate and control the contact forces? Since pipe crawling heavily relies on contacts, the robot has to be able to sense and control them. This is not an easy task for a position controlled robot such as Krock-2, that is not equipped with a high precision contact force / joint torque sensors.

---

[1] Keep it simple and straightforward.

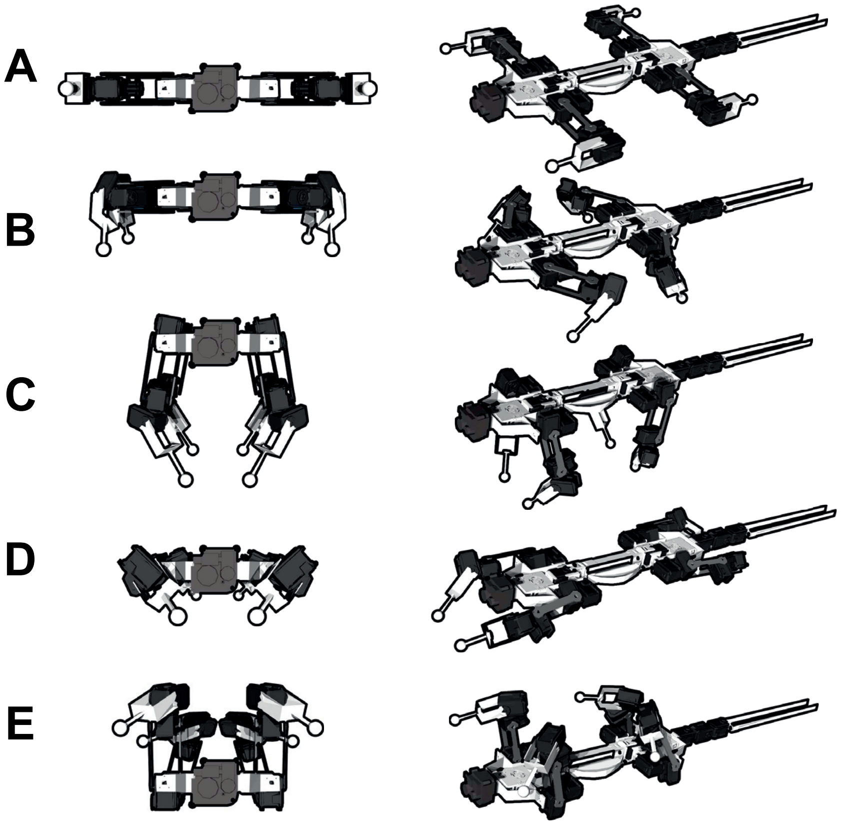


Figure 8.1 – Gait width comparison of possible locomotion strategies. (**A**) Joints at zero position. (**B**) Regular walking. (**C**) High posture with narrow walking gait. (**D**) Low posture with narrow gait. (**E**) Posture adapted for pushing against parallel walls.

- How to distribute the contact forces between the feet? Slipping between the feet and the wall should be avoided, while successfully overcoming friction between the body and the ground as well as gravity (slope). At the same time, the contact force should stay within the limits imposed by the servo motors.

- How to handle a pipe curvature? Lighting conditions inside a pipe are usually not great which means the robot should be able to follow the pipe without relying on cameras or manual steering by an operator within a line of sight.

8.3 Pipe crawling algorithm

Our pipe crawling algorithm is inspired by the work done in [54], where Focci et al. designed a controller that allowed a torque controlled quadruped robot to walk inside an inclined groove. Although we do not have a torque controlled robot, many ideas from their work can be adapted to our pipe crawling problem. In the following sections we describe our solutions to previously stated questions.

8.3.1 Crawling state machine

We can break down the pipe crawling algorithm into several phases that are sequentially repeating (Fig. 8.2). Each sequence starts by the *leg selection* that determines which leg(s) will execute a swing phase next. Before breaking the contact with a wall, the contact force reference for the leg is gradually reduced during the *unload leg* phase. In the *move swing leg* phase, the foot follows a user predefined trajectory, reaching towards the next foothold. The *loading leg* phase is triggered when a contact between swing leg and the wall is detected. In this phase the contact force reference is gradually increased until it reaches the desired value. If there is no detected contact by the end of the swing phase, the leg moves towards the wall in the *wall approach* phase. If the leg cannot reach the wall, the leg will search for the ground (*ground approach* phase). We assume the wall and ground surface vectors are perpendicular to each other and to the robot's line of locomotion. Another assumption is that the ground is always reachable. Therefore *ground approach* phase will always achieve the contact and transition into *load leg* phase. The transitions between phases are governed by a crawling state machine shown in Fig. 8.2. The *leg selection* depends on the user defined gait sequence. When the robot is lying on flat ground, the fastest way to crawl is to execute swing phase for all the legs at once. If the ground is inclined and there is a risk of slipping through the pipe, the robot should at all times support itself against the pipe. In such case, an appropriate gait sequence is to only move a pair of legs (front or hind) at a time, while the other pair maintains wall contact.

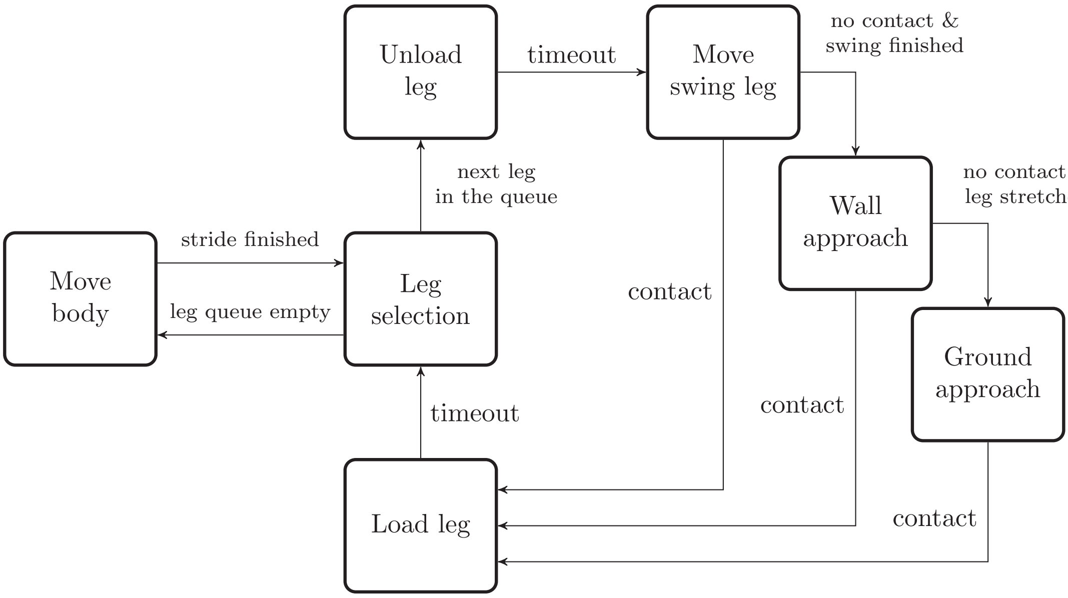


Figure 8.2 – State machine of crawling gait.

8.3.2 Contact force distribution

The pipe crawling method we described heavily relies on maintaining multiple body and leg contacts with an environment. Unlike during walking, these contacts produce forces which greatly counteract each other resulting in internal forces through the robot's body. A force $\boldsymbol{f}_c$ at the foot contact point can be split into two orthogonal components: the normal force $\boldsymbol{f}_\perp = (\boldsymbol{n}_s^\top \boldsymbol{f}_c)\boldsymbol{n}_s$, aligned with the normal vector of the contact surface, and the shear force $\boldsymbol{f}_\parallel = \boldsymbol{f}_c - \boldsymbol{f}_\perp$. If we assume a Coulomb friction model, the no slipping condition holds if the following is true:

$$\|\boldsymbol{f}_\parallel\| \leq \mu_f \|\boldsymbol{f}_\perp\|, \tag{8.1}$$

where μ_f is a static friction coefficient. This expression defines a friction cone and as long as the contact force vector is within it, the slipping will not occur. This is important because the robot relies on shear forces to push itself forward, assuming the pipe walls are aligned with the direction of locomotion. A role of normal force is to assure the contact force at each feet stays within its friction cone. A resulting force vector, in order to produce successful locomotion, has to overcome a friction between robot's body and the ground and compensate the gravity when on a slope. In the following we describe how the forces should be distributed between the contact feet to satisfy such conditions. The approach is based on the work of Focci et al. [54], which we adapt to our scenario.

We start by assuming that the robot's spine motion during crawling can be neglected, therefore the trunk can be approximated by a rigid body to which we will refer as robot's base. Another assumption is that the center of mass (CoM) is fixed to the base, although its location also depends on the legs motion. The balance of forces, in robot's frame of reference, acting on the CoM when the robot has $1 \leq k \leq 4$ legs in the contact with an environment is:

$$m\ddot{\boldsymbol{x}}_{\text{com}} + \boldsymbol{f}_{\text{bex}} = \sum_{i=1}^{k} \boldsymbol{f}_{c,i}, \tag{8.2}$$

where m is the mass of the robot, $\ddot{\boldsymbol{x}}_{\text{com}} \in \mathbb{R}^{3\times 1}$ acceleration of the CoM and $\boldsymbol{f}_{c,i} \in \mathbb{R}^{3\times 1}$ is contact force acting on i^{th} foot. The force $\boldsymbol{f}_{\text{bex}} \in \mathbb{R}^{3\times 1}$ represents external forces acting on the base, other than the ones caused by feet contacts, like body-ground friction and gravity component parallel to the ground. For example, if the robot lays on a slope with an inclination of ϕ_s [rad] and the robot's direction is aligned with the slope (which holds true for our scenario), $\boldsymbol{f}_{\text{bex}}$ equals to:

$$\boldsymbol{f}_{\text{bex}} = \begin{bmatrix} mg\sin(\phi_s) + m\mu_b\cos(\phi_s) & 0 & 0 \end{bmatrix} \tag{8.3}$$

where μ_b is the friction coefficient between a ground and robot's body. If the robot lays flat on the ground and the only torque acting on the base is due to the feet contacts with an environment, we can write:

$$\boldsymbol{I}_{\text{com}}\dot{\boldsymbol{\omega}}_{\text{com}} = \sum_{i=1}^{k} \boldsymbol{p}_{\text{com},i} \times \boldsymbol{f}_{c,i}, \tag{8.4}$$

where $\boldsymbol{I}_{\text{com}} \in \mathbb{R}^{3\times3}$ is rotational inertia, $\dot{\boldsymbol{\omega}}_{\text{com}} \in \mathbb{R}^{3\times1}$ is angular acceleration of the base and $\boldsymbol{p}_{\text{com},i} \in \mathbb{R}^{3\times1}$ is a position vector pointing from CoM to the location of i^{th} foot.

Relations 8.2 and 8.4 can be compactly written in a matrix form:

$$\underbrace{\begin{bmatrix} \boldsymbol{I} & \cdots & \boldsymbol{I} \\ [\boldsymbol{p}_{\text{com},1}\times] & \cdots & [\boldsymbol{p}_{\text{com},k}\times] \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} \boldsymbol{f}_{c,1} \\ \vdots \\ \boldsymbol{f}_{c,k} \end{bmatrix}}_{\boldsymbol{f}_c} = \underbrace{\begin{bmatrix} m\ddot{\boldsymbol{x}}_{\text{com}} + \boldsymbol{f}_{\text{bex}} \\ \boldsymbol{I}_{\text{com}}\dot{\boldsymbol{\omega}}_{\text{com}} \end{bmatrix}}_{\boldsymbol{b}} \tag{8.5}$$

To solve this system of equations for $\boldsymbol{f}_c$, while satisfying additional constraints, we formulate an optimization problem:

$$\min_{\boldsymbol{f}_c} \quad (\boldsymbol{A}\boldsymbol{f}_c - \boldsymbol{b})^\top \boldsymbol{W}(\boldsymbol{A}\boldsymbol{f}_c - \boldsymbol{b}) + \lambda \boldsymbol{f}_c^\top \boldsymbol{f}_c \tag{8.6a}$$

$$\text{s.t.} \quad \boldsymbol{d}_l \le \boldsymbol{C}\boldsymbol{f}_c \le \boldsymbol{d}_u \tag{8.6b}$$

The cost function is aimed to minimize a deviation from the exact solution of 8.5 while keeping the contact forces as low as possible. The weights $\boldsymbol{W} \in \mathbb{R}^{6\times6}$ and $\lambda \in \mathbb{R}$ define a trade-of between these two objectives. The inequality constraint places the solution inside the friction cone and within the limits $f_{c,\min,i}$ and $f_{c,\max,i}$. The matrix $\boldsymbol{C} \in \mathbb{R}^{5k\times3k}$ and vectors $\boldsymbol{d}_l$, $\boldsymbol{d}_u \in \mathbb{R}^{5k\times1}$ are constructed as follows:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{C}_k \end{bmatrix}, \qquad \boldsymbol{d}_l = \begin{bmatrix} \boldsymbol{d}_{l,1} \\ \vdots \\ \boldsymbol{d}_{l,k} \end{bmatrix}, \qquad \boldsymbol{d}_u = \begin{bmatrix} \boldsymbol{d}_{u,1} \\ \vdots \\ \boldsymbol{d}_{u,k}, \end{bmatrix} \tag{8.7}$$

where

$$\boldsymbol{C}_i = \begin{bmatrix} (-\mu_{f,i}\boldsymbol{n}_{s,i} + \boldsymbol{t}_{s1,i})^\top \\ (-\mu_{f,i}\boldsymbol{n}_{s,i} + \boldsymbol{t}_{s2,i})^\top \\ (\mu_{f,i}\boldsymbol{n}_{s,i} + \boldsymbol{t}_{s1,i})^\top \\ (\mu_{f,i}\boldsymbol{n}_{s,i} + \boldsymbol{t}_{s2,i})^\top \\ \boldsymbol{n}_{s,i} \end{bmatrix}, \qquad \boldsymbol{d}_{l,i} = \begin{bmatrix} -\infty \\ -\infty \\ 0 \\ 0 \\ f_{\min,i} \end{bmatrix}, \qquad \boldsymbol{d}_{u,i} = \begin{bmatrix} 0 \\ 0 \\ \infty \\ \infty \\ f_{\max,i} \end{bmatrix} \tag{8.8}$$

The first four rows in matrix C_i define a friction pyramid as an approximation of a cone, where $\boldsymbol{n}_{s,i}$ is a normal vector to the contact surface while $\boldsymbol{t}_{s1,i}$ and $\boldsymbol{t}_{s2,i}$ are two non collinear vectors orthogonal to $\boldsymbol{n}_{s,i}$. The fifth row sets a box constraint on the magnitude of the contact force.

8.3.3 Contact force estimation

Being able to detect contacts and estimate the contact force is a fundamental requirement for the task we are trying to solve. Krock-2 is equipped with 3D force sensors Optoforce. Each leg has the sensor located within an enclosure that is mounted between the foot and the knee, as shown in the Chapter 7 (Fig. 7.1). The sensors are factory calibrated only around the vertical axis when the contact point is exactly in the middle of the sensor's hemisphere. Due to our enclosure mechanism, the contact point on the sensor can significantly vary, making the readings imprecise. To improve the measurements, we have to build an estimator.

In [190], Wagner et al. took a model-based approach to estimate a ground inclination using the same sensors. Although it is possible to model the sensor itself, the difficulties come from the way forces are transfered from the foot to the sensor itself. The cup touching the sensor is not glued to it, meaning the point of contact on the sensor depends on magnitude and direction of the force applied to the foot and on friction between the rod and sensor enclosure. To avoid such difficulties, we take a model-free approach, that relies on an artificial feed-forward neural network. Similar approach was taken by Chuah et al. [26], who used neural networks to relate contact forces with internal states of their pressure based force sensor.

As an input to the neural network we use temperature compensated readings $S_1, ..., S_4$ from the sensor's four photodiodes. Estimated force values provided by the sensor are constructed as a linear combination of those four values ([183], [190]):

$$\begin{aligned} F_x &= \frac{S_1 - S_3}{2}, \\ F_y &= \frac{S_4 - S_2}{2}, \\ F_z &= \frac{S_1 + S_2 + S_3 + S_4}{4}. \end{aligned} \tag{8.9}$$

This relation hold when the contact point is on top of the sensor, which is not necessary our case. By using photodiode instead of force readings, we increase our input space dimension from $\mathbb{R}^3$ to $\mathbb{R}^4$ which will potentially provide more useful data for the neural network.

The data collection procedure was to push the robot's foot by hand against a force plate (type 9260AA3, Kistler, 2011), which is used as a ground truth (targets). At the same time photodiode signals and leg joint angles were recorded. The joint angles are needed to reconstruct leg's forward kinematics and extract the foot orientation with respect to the force plate. If a backlash in the sensor enclosure which is not measurable is ignored, the sensor orientation matches the one of the foot. The procedure is repeated for each of the legs. In the end, around 45 min of data was recorded per leg, sampled at 100 Hz.

For each leg, a two layer neural network with 30 neurons in the hidden layer was trained in Matlab using Neural Network Toolbox. The networks were deployed on the robot in C++ as a class that accepts network parameters provided by Matlab and automatically generates the corresponding network structure.

8.3.4 Contact force control

Robots like HyQ, Anymal and MIT-Cheetah excel in controlling their contact forces due to actuators which are capable of good torque control. On the other hand, Krock-2 is actuated by position controlled servo motors. An overview of robot force control can be found in [202]. A simple method we can apply to our robot is admittance control [195] that relates the force error to the end effector velocity. The basic assumption is that of a compliant environment where end effector can push in. If the environmental stiffness is k_{env}, then the force produced by the robot's end effector pushing into it is:

$$\boldsymbol{f} = k_{\text{env}} \boldsymbol{\delta}, \tag{8.10}$$

where $\boldsymbol{\delta}$ is displacement vector measured from the surface of undisturbed environment. The needed compliance (the inverse of stiffness) is inherent to the design of Krock-2's legs. Deflections in mechanical leg structure, silicone surface of Optoforce sensors as well as a silicon ball-foot provide needed compliance even if the environment is completely stiff.

Let's denote with $\boldsymbol{e}_{f,i}(k)$ the difference (error) between contact force reference $\boldsymbol{f}_{c,i}(k)$ and estimated force $\boldsymbol{f}_{\text{est},i}(k)$ of $i-$th leg at time step k:

$$\boldsymbol{e}_{f,i}(k) = \boldsymbol{f}_{c,i}(k) - \boldsymbol{f}_{\text{est},i}(k). \tag{8.11}$$

The idea of admittance control is to drive the force error to zero by adjusting the end effector position. This can be done by using a PID controller:

$$\boldsymbol{\delta}_{f,i}(k) = k_{p,a} \boldsymbol{e}_{f,i}(k) + k_{i,a} \sum_{j=0}^{k} \boldsymbol{e}_{f,i}(j) + k_{d,a} \frac{\boldsymbol{e}_{f,i}(k) - \boldsymbol{e}_{f,i}(k-1)}{dt}, \tag{8.12}$$

where $\boldsymbol{\delta}_{f,i} \in \mathbb{R}^3$ is a control signal and $k_{p,a}$, $k_{i,a}$ and $k_{d,a}$ are proportional, integral and derivative gains respectively. The gains are scalar in $\mathbb{R}$, however they can be replaced with vectors in $\mathbb{R}^3$ so the user can tune the controller separately for all three components of force.

The signal $\boldsymbol{\delta}_{f,i}$ provides a position adjustment for a foot in x, y and z direction. The adjustment can be split into tangential and normal component with respect to the contact surface. Due to uncertainties in force estimation and friction coefficients (both body-ground and foot-wall) trying to control the tangential component could produce unwanted slipping or motion of legs or of the entire body. Thus we choose to only control the force in the normal direction to the wall by using a modified control signal:

$$\boldsymbol{\delta}_{n,f,i}(k) = \boldsymbol{n}_{s,i}^{\top} \boldsymbol{\delta}_{f,i}(k). \tag{8.13}$$

If $\boldsymbol{p}_{c,i}$ is the position of i-th foot at the moment of wall contact (transition into state *Load leg* in

Fig. 8.2), then the modified foot trajectory reference is calculated as:

$$\hat{\boldsymbol{p}}_{c,i} = \boldsymbol{p}_{c,i} + \boldsymbol{\delta}_{n,f,i}. \tag{8.14}$$

8.3.5 Pipe navigation

Because of the large relative robot's size compared to openings it aims to crawl through, the robot's heading and spine bending have to be controlled to prevent crashing into walls. Furthermore, the robot has to be aware of the local curvature of the path ahead to properly orient its girdles in an attempt to follow it. Crawling through pipes, ducts or cracks in cluttered environments makes it difficult for an operator to keep the robot within a direct line of sight and thus precisely control robot's movements. Using the front mounted camera could potentially help with manually controlling the heading, however the operator would not have an overview of full robot's state within the pipe. To simplify operating the robot, the heading, together with body posture adaptation to follow the pipe midline will be included to the crawling controller as an extra module. Such modules give the robot partial autonomy, requiring the operator to only command the crawling speed. The module will have three main tasks: pipe curvature estimation, path following and spine adaptation.

The robot's only exteroceptive sensor that could be used to estimate the pipe curvature is the camera, but using image processing to estimate the curvature would not be an easy task. Another option would be to use a laser range finder, which would require mounting additional sensors and modifying the hardware. Thus we decided to rely on proprioception and odometry.

The main assumption when using the odometry is that the feet slippage is low. The robot continuously updates its position in the world frame by integrating the displacement during the *Move body* phase. The idea is to track robot's feet contacts with the wall and from them estimate the curvature of the pipe. Another assumption is that both left and right feet achieve contacts at the same height meaning a midline between those points is also the midline of the pipe. To estimate a local curvature, the robot only needs to track its intermediate contacts thus eliminating cumulative odometry error.

Let's denote the position of i-th leg in the world frame at the beginning of the current and previous *Move body* phase with $\boldsymbol{p}_{0c,i}$ and $\boldsymbol{p}^{-}_{0c,i}$ respectively. These variables represent points on the wall and do not include the admittance control modification $\boldsymbol{\delta}_{n,f,i}$. Local curvature can be estimated by fitting a circle with origin $\boldsymbol{O} = \begin{bmatrix} x_o, & y_o \end{bmatrix}^\top$ and radius R through pipe's midline, ideally passing exactly in between contact points made by left and right feet. To find a circle which satisfies such condition, we formulate an unconstrained optimization problem:

$$\min_{\boldsymbol{O}} \quad \left(2R\boldsymbol{I}_{4\times1} - \boldsymbol{R}_L(\boldsymbol{O}) - \boldsymbol{R}_R(\boldsymbol{O})\right)^\top \boldsymbol{W} \left(2R\boldsymbol{I}_{4\times1} - \boldsymbol{R}_L(\boldsymbol{O}) - \boldsymbol{R}_R(\boldsymbol{O})\right) \tag{8.15a}$$

where vectors $\boldsymbol{R}_L(\boldsymbol{O}) \in \mathbb{R}^{4\times1}$ and $\boldsymbol{R}_R(\boldsymbol{O}) \in \mathbb{R}^{4\times1}$ contain the distances of left (legs 1 and 3) and right (legs 2 and 4) feet from the circle's origin $\boldsymbol{O}$:

$$\begin{aligned}\boldsymbol{R}_L(\boldsymbol{O}) &= \left[\left\|\boldsymbol{O}-\boldsymbol{p}_{0c,1}\right\|, \quad \left\|\boldsymbol{O}-\boldsymbol{p}_{0c,3}\right\|, \quad \left\|\boldsymbol{O}-\bar{\boldsymbol{p}}_{0c,1}\right\|, \quad \left\|\boldsymbol{O}-\bar{\boldsymbol{p}}_{0c,3}\right\|\right], \\ \boldsymbol{R}_R(\boldsymbol{O}) &= \left[\left\|\boldsymbol{O}-\boldsymbol{p}_{0c,2}\right\|, \quad \left\|\boldsymbol{O}-\boldsymbol{p}_{0c,4}\right\|, \quad \left\|\boldsymbol{O}-\bar{\boldsymbol{p}}_{0c,2}\right\|, \quad \left\|\boldsymbol{O}-\bar{\boldsymbol{p}}_{0c,4}\right\|\right].\end{aligned} \tag{8.16}$$

The diagonal weight matrix $\boldsymbol{W} \in \mathbb{R}^{4\times4}$ is used to give lower weight to points from a previous step, taking into account possible errors in odometry. The circle radius R is determined by a location of the front girdle $\boldsymbol{p}_{fg}$:

$$R = \left\|\boldsymbol{O}-\boldsymbol{p}_{fg}\right\|. \tag{8.17}$$

The fitted circle represents the path to be followed during the upcoming *Move body* phase.

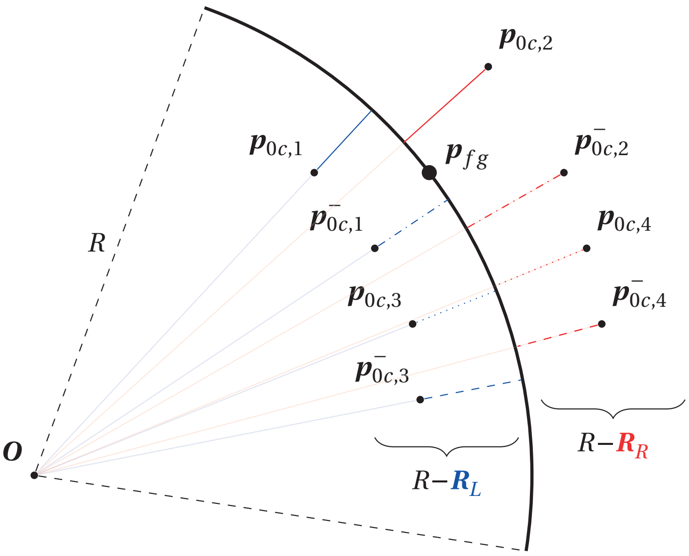


Figure 8.3 – Odometry based local curvature estimation. The algorithm places a circle passing through front girdle $\boldsymbol{p}_{fg}$. The circle's origin $\boldsymbol{O}$ and radius R are computed in a way the difference in distance between feet contact pairs and the circle is minimized. The distances of corresponding pairs are illustrated with the same line type, while the color indicates left (blue) and right (red) feet contact.

Robot's motion during that phase is controlled in the same way as explained in Chapter 3, meaning that only inputs are linear and angular velocity of the front girdle. The linear velocity is a user provided parameter, and angular velocity is controlled by a path follower PD controller with gains $k_{p,pf}$ and $k_{d,pf}$, applied on the error between the robot's heading and the vector connecting the front girdle and a point on the path. The point is defined by a look-ahead distance lh_{pf}.

8.4 Experiments

Evaluation of the proposed framework is done in both simulation and on the real hardware. The experiments are designed to test all the different components of the framework with an emphasis on the force distribution, admittance controller and curvature estimation. The main parameters defining gait, controllers or environment are shown in table 8.1. Their tuning was done in simulation, however a few parameters had to be re-tuned for the hardware experiments[2].

Table 8.1 – Crawling parameters. If different values were used in simulation and hardware experiments, it is indicated with (S) or (H) respectively.

Parameter	Symbol	Value
Foot-wall friction coefficient	μ_f	0.5
Body-ground friction coefficient	μ_b	0.25
Friction tuning parameter	p_μ(%)	0.25
Minimum and maximum contact force	$f_{\text{fmin}}, f_{\text{max}}$ [N]	−60, 60
Admittance controller proportional gain	$k_{p,a}$ [$\mathrm{m\,N^{-1}}$]	0.05 (S), 0.06 (H)
Admittance controller derivative gain	$k_{d,a}$ [$\mathrm{m\,s\,N^{-1}}$]	0.05
Path following controller proportional gain	$k_{p,pf}$ [$\mathrm{m\,N^{-1}}$]	0.05
Path following controller derivative gain	$k_{d,pf}$ [$\mathrm{m\,s\,N^{-1}}$]	0.05
Path following look ahead distance	lh_{pf} [m]	0.05
Leg loading / unloading force rate	F_{rate} [$\mathrm{N\,s^{-1}}$]	25 (S), 30 (H)
Leg approach speed	v_{app} [$\mathrm{m\,s^{-1}}$]	0.05
Contact detection force threshold	— [N]	0.05 (S), 5 (H)
Stride length	— [m]	0.12
Swing sequence - flat	—	(FL, FR, HL, HR)
Swing sequence - slope	—	(FL, FR), (HL, HR)

8.4.1 Simulation

To simulate the robot and an environment, we used Webots$^{\text{TM}}$. The pipes are represented as a 3D mesh surface that is automatically generated from a user specified central line and cross section. The simulated robot is made of rigid links and lacks structural compliance assumed by the admittance controller. As a workaround, we increased the value of *Constraint Force Mixing* parameter used by Webots' internal physics solver ODE, which controls the softness of contacts. This provided needed environmental compliance allowing robot's feet to penetrate into the contact surface, making the admittance control possible. Modeling Optoforce sensors and their mounting mechanism would be difficult. Instead we use ideal force sensors mounted directly on each of the feet, which makes the entire contact force estimator not needed in simulation.

Simulated pipes were created by using Webots' built in *SolidPipe.proto* definition, that generates

---

[2] The video of the experiments can be fount at: https://biorob2.epfl.ch/video/313

a 3D mesh defined by a central curve and a cross-section geometry. Across the experiments, we made pipes with circular cross section with a 0.2 m radius. A practical demonstration of the pipe crawling algorithm is shown in Fig. 8.4. The robot traversed a horizontally laid pipe with a single turn of a radius 0.5 m. The Fig. reports snapshots of the experiment, that were taken approximately 20 s apart. Under each Webots snapshot, we report a corresponding odometry based curvature estimation i.e. what the robot "sees". Although the odometry based path does not correspond to the pipe geometry, the robot successfully finished the course. This shows that estimation of the local curvature is good enough, to steer the robot through the turn. Desired and measured contact forces for the front left leg during the experiment are reported in Fig. 8.5.

To test the algorithm in more complex scenarios, three more simulated pipe environments were created. The first one (Fig. 8.6) contains a series of turns and straight segments. The total length of the pipe is around 14 m for which to finish it took the robot slightly over 9 minutes. The second scenario (Fig. 8.7) is designed to test the robot's ability to crawl up the slope. The midline of pipe is a parabola, meaning the slope constantly increases. It took the robot around 18 min to crawl through 15 m of the pipe. The difference in time it took to finish first and second course, although the pipe length is similar, comes from the use of different gaits. In the first case, the robot lays flat on the ground at all times and can execute swing phase of all the legs simultaneously. In the second scenario, where the robot is on the slope, wall contacts need to always be maintained to prevent sliding down the slope. Therefore, only one pair of legs executes swing phase at the time, making entire crawling slower. Finally, the third scenario (Fig. 8.8) contains a helix shaped pipe which combines the first two scenarios, requiring the robot to constantly turn and climb a slope. The pipe length is 12.9 m, curvature radius is 1 m and the helix is 3 m high. It took the robot around 20 min to finish the course. Finally, to test the controller when the distance between opposite walls is not constant, we created a simulated corridor with the width variation of up to 15 cm. The variation did not interfere with performance of the curvature estimation module. The experiment is shown in Fig. 8.9.

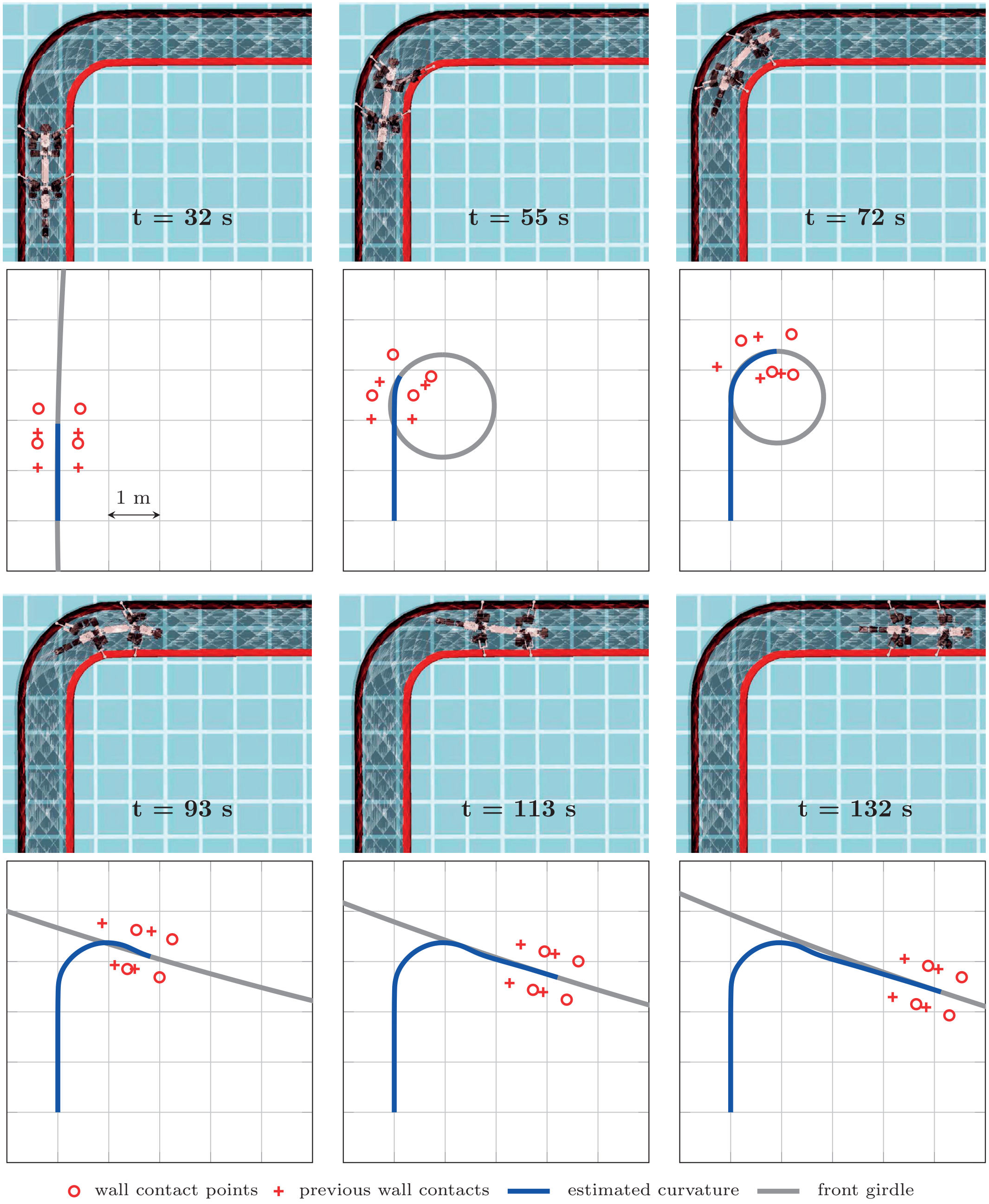


Figure 8.4 – Snapshots of a crawling through a turn. Underneath each simulation snapshot, that are approximately 20 s apart, we show a state of odometry based curvature estimation. The state includes current (red circles) and previous (crosses) wall contacts used to estimate a circle (gray) that defines the local curvature. The path made by the front girdle is shown in blue.

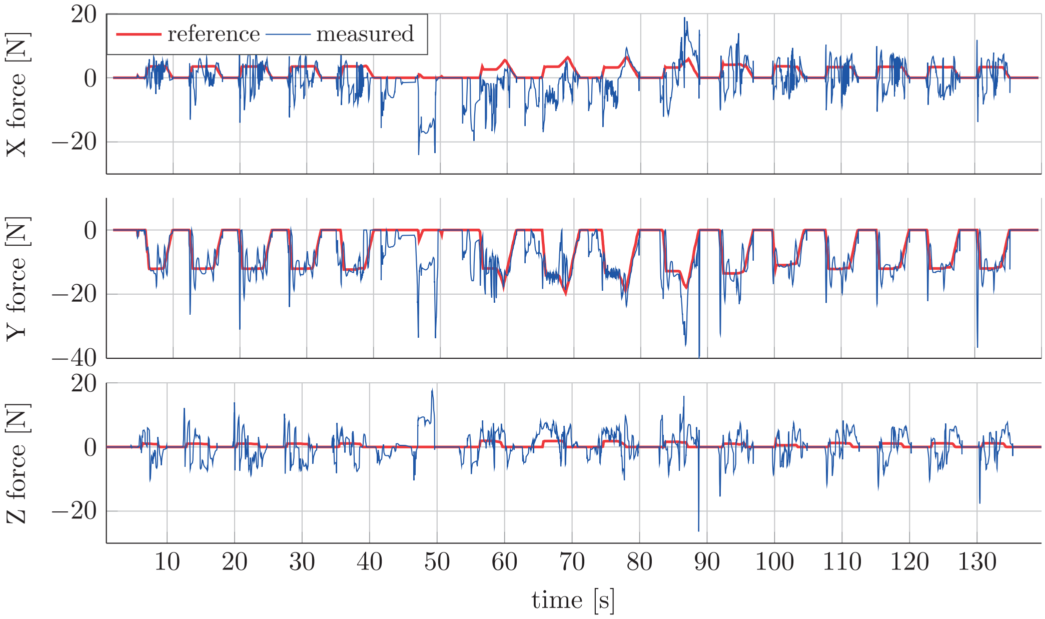


Figure 8.5 – Contact force reference and measured force of the front left leg during the simulation experiment is shown in 8.4. The admittance controller acts only on the normal component of the contact force, which matches Y component of the force.

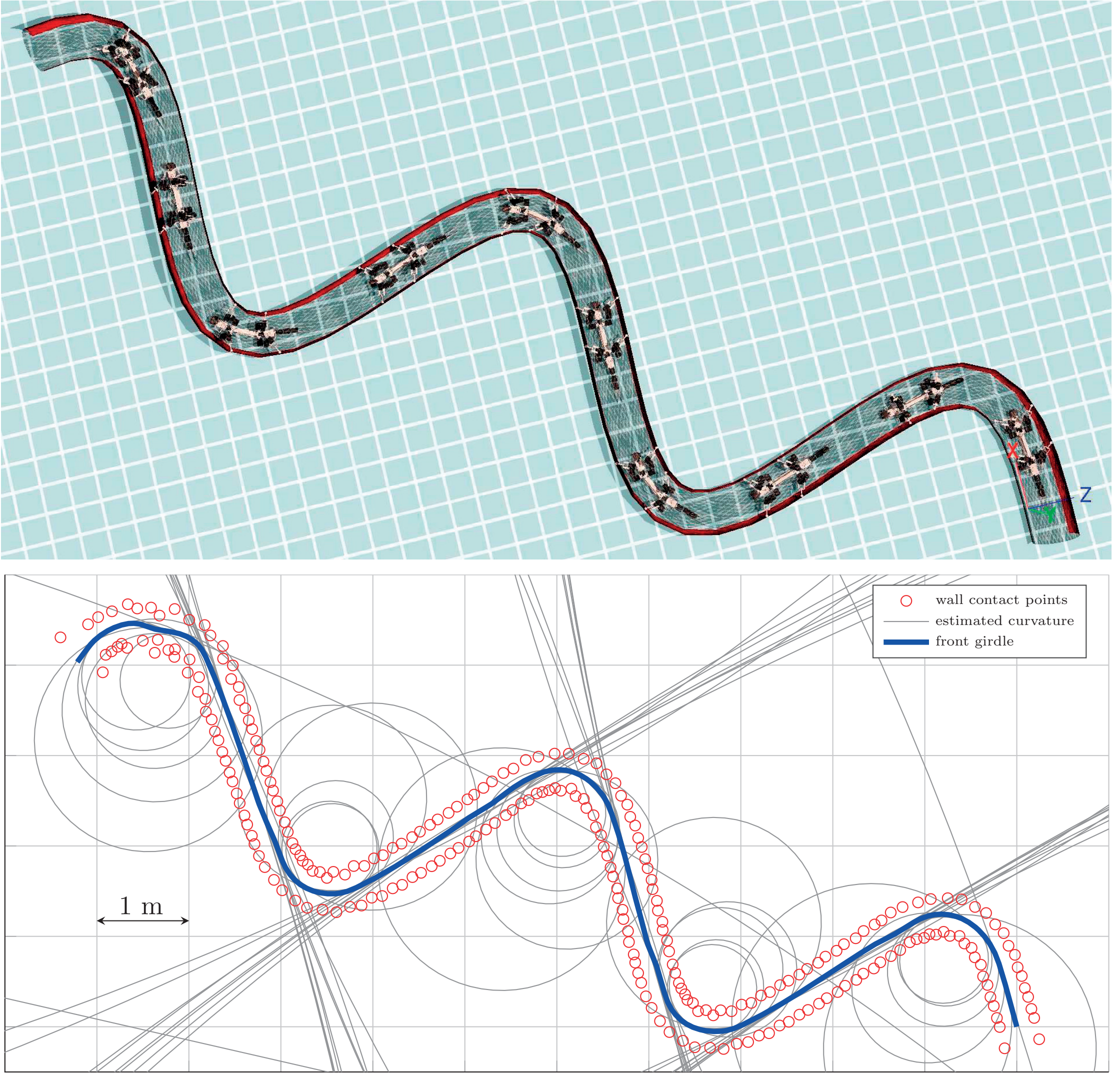


Figure 8.6 – Overlaid snapshots of crawling through a pipe with multiple turns. The simulation snapshots (up) are 120 s apart. The odometry view (down) shows estimated local curvature (gray circles and arcs) for each crawling step during the entire simulation run.

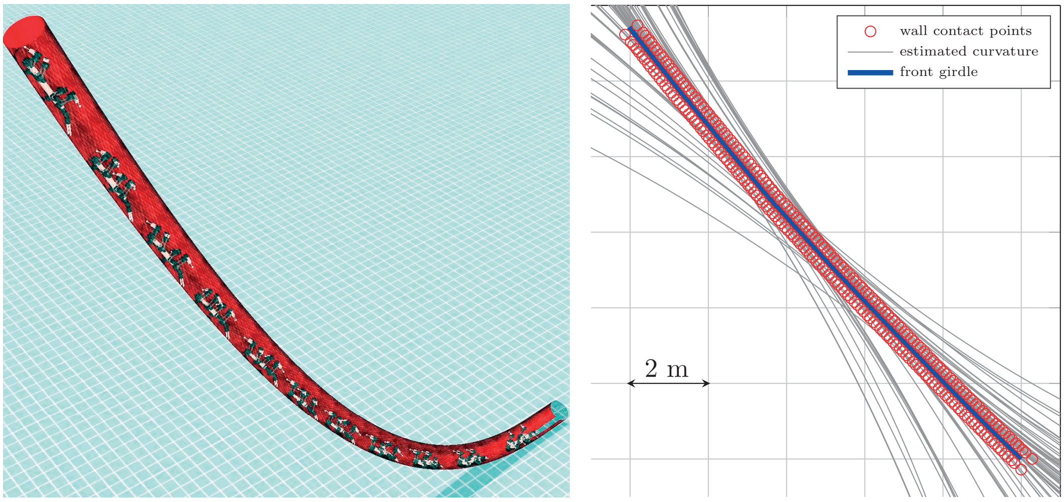


Figure 8.7 – Overlaid snapshots of crawling through a pipe with increasing inclination. The simulation snapshots (left) are 240 s apart. The odometry view (right) shows estimated local curvature (arcs) for each crawling step during entire simulation run. The odometry is computed in a 2D plane (robot's coronal plane) and does not capture the inclination of the pipe.

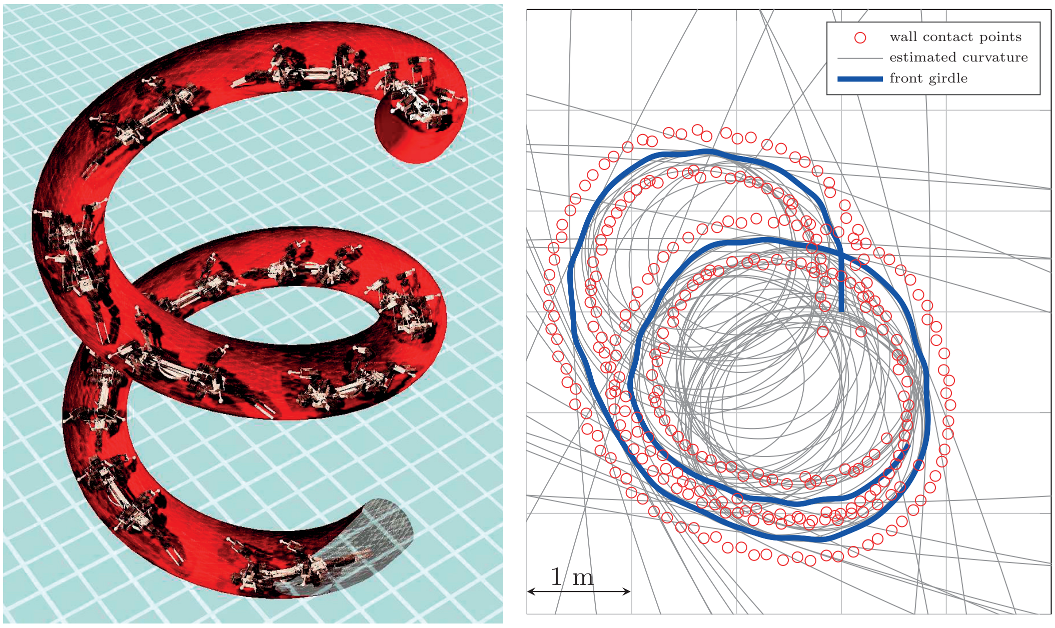


Figure 8.8 – Overlaid snapshots of crawling through a helix pipe. The simulation snapshots (left) are 120 s apart. The odometry view (right) shows estimated local curvature (circles and arcs) for each crawling step during entire simulation run. The odometry is computed in a 2D plane (robot's coronal plane) and does not capture the inclination of the pipe.

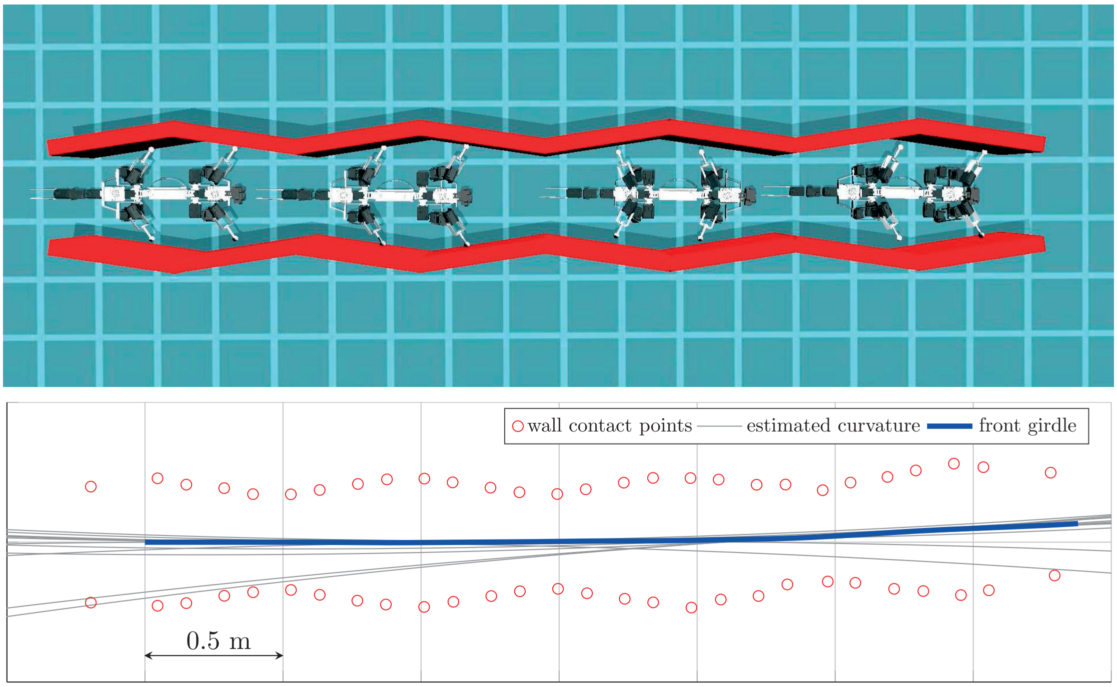


Figure 8.9 – Overlaid snapshots of crawling through a corridor with the variable width from 32 cm to 47 cm. The simulation snapshots (top) are 40 s apart. The odometry view (bottom) shows estimated local curvature (arcs) for each crawling step during entire simulation run.

8.4.2 Hardware experiments

The only difference in the control algorithm between simulation and hardware experiments is in the contact force estimation. While the problem is bypassed in simulation by using the idealized force sensors, the real hardware used artificial neural networks to estimate the force based on Optoforce readings. To evaluate the estimation accuracy, the robot performed a series of steps and posture adjustments on a force plate. The comparison is shown in Fig. 8.10.

The initial crawling test is performed in a straight pipe segment with a length of 2.1 m. Due to construction simplicity, the pipe is represented by two parallel walls (planks) placed 40 cm apart. The snapshots of a single stride and contact forces are shown in Fig. 8.11. The force component perpendicular to the wall (Y direction) is the only one controlled by the admittance controller and generally follows (qualitatively) the reference, however there are significant oscillations present during *Move body* phase. Nevertheless the robot successfully finished the task, without significant slipping, which can be credited to a conservative choice of relation between body and foot friction coefficients. To test curvature estimation, we constructed a half-circular path with walls placed 42 cm apart. Snapshots of the experiment and odometry view are shown in Fig. 8.12. The ground was made of rough concrete with a high body-ground friction. On the other hand, the walls were made of a smooth polyvinyl chloride (PVC) plastic resulting in a relatively low

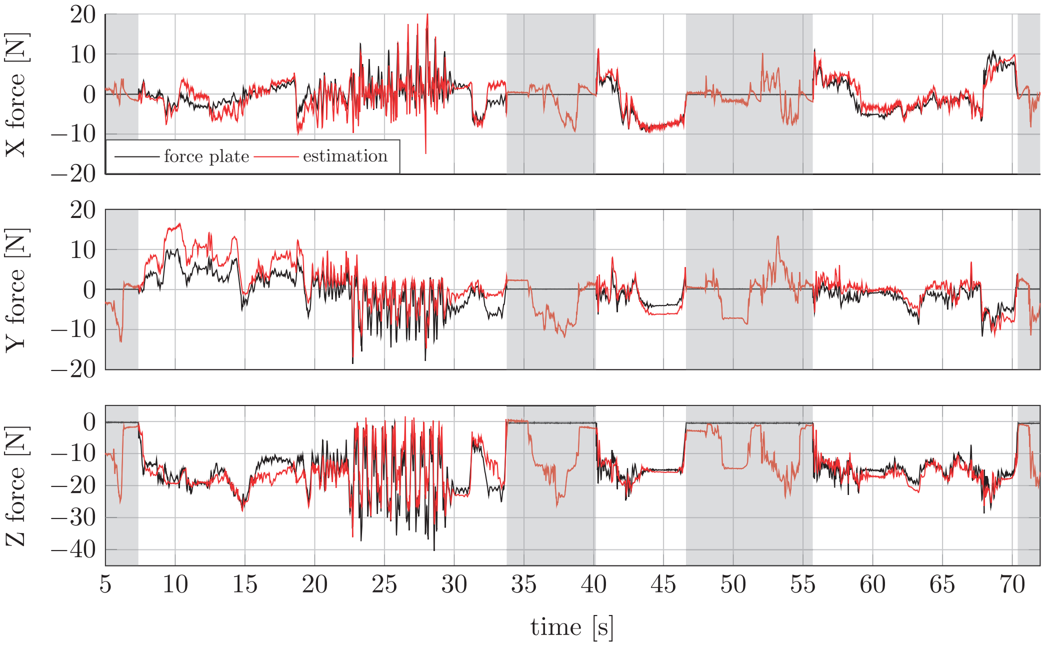


Figure 8.10 – Evaluation of the force estimation by the neural network used across all the experiments. In the evaluation, the robot performed a series of steps with its front right leg over the force plate (FP). During longer contacts, the robot was commanded to move its body without lifting the leg to produce a wider range of contact forces. The shaded segments indicate the leg was not in the contact with the FP (it was either in the air or stepped next to the FP).

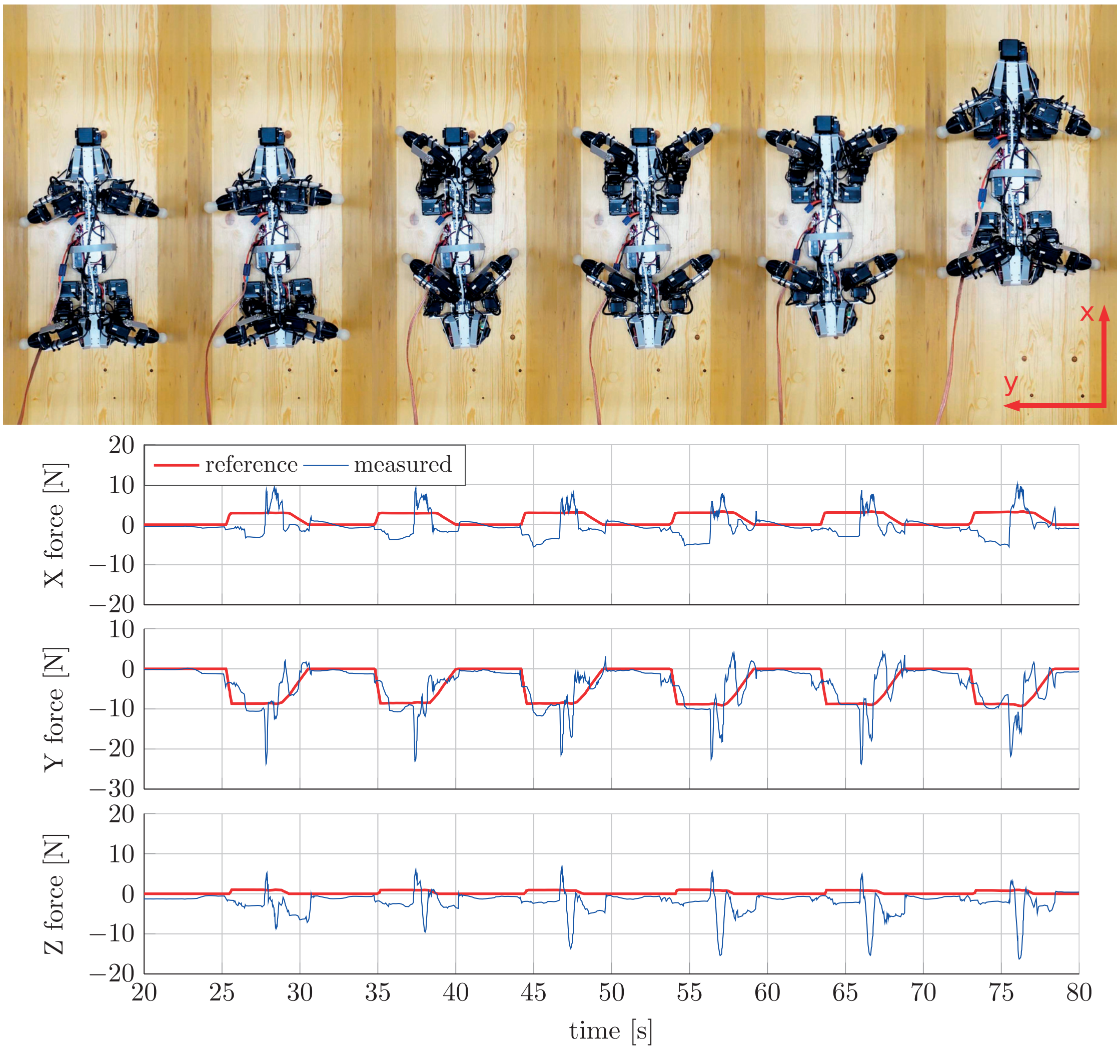


Figure 8.11 – Crawling experiment through a straight flat segment. (Up) Snapshots of a single stride, showing simultaneous swing phase of all the legs and *Move body* phase. The snapshots are 2 s apart. (Down) Desired and estimated contact force of the front left leg. The Y component is controlled by the admittance controller.

foot-wall friction. As a result some feet slipping was present, which is visible in the odometry view. The difference in friction between experiments was adjusted for by changing p_μ parameter, as presented in the table 8.1. Finally, we initiated a slope experiment where we inclined the setup used for the straight segment experiment. At the inclination of 25° the robot was able to push forward during *Move body* phase, however it failed to firmly hold during a swing phase with only a single pair of legs pushing against the wall. Due to a risk of breaking the robot, we decided not to continue with the experiments at the current state of hardware and crawling algorithm. The details about the problem and possible solutions are discussed in the following section.

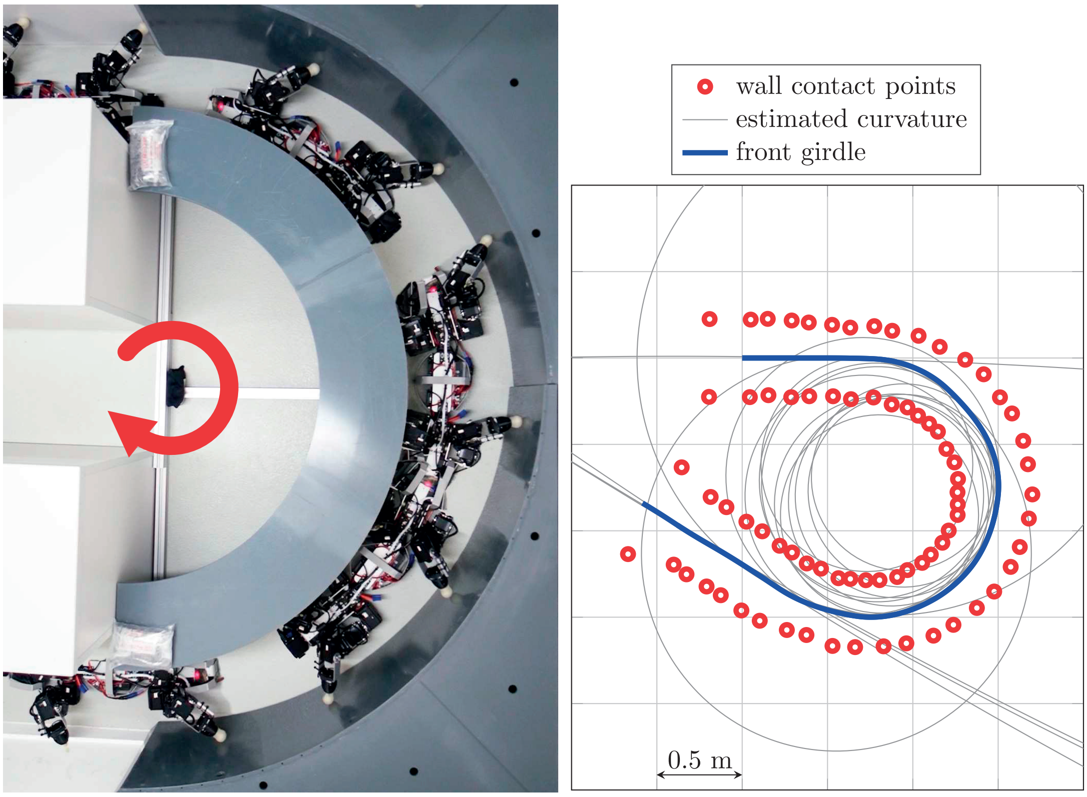


Figure 8.12 – Crawling experiment through a turn. (Left) The experimental setup with snapshots of the robot that are 40 s apart. Width of the path is 42 cm. (Right) The odometry view showing detected wall contacts, estimated curvature and path made by the front girdle.

8.5 Discussion

In this chapter we showed how a specialized type of locomotion like crawling through pipes can be achieved with the robot that is not specifically built for it. The robot is now capable of another mode of locomotion, vastly different from a "normal" walking gait. The new locomotion capabilities are demonstrated on the examples of crawling through circular pipes or between parallel walls.

This method is specifically crafted for sprawling posture robots, however we hope it can bring some value to other quadruped or even other multilegged systems. Sprawling posture robots like Krock-2 are not proficient in using an upright walking posture having feet and legs fully underneath hip joints. Such posture would make a walking gait as narrow as the distance between hips and the crawling gait proposed in this chapter would not help to reduce gait width. However, a possible value of the proposed gait lies in a high slope locomotion, where the friction between ground and feet is not high enough for the robot to climb the slope. In such a case the robot could utilize surrounding walls and obstacles to generate contact forces that would stay within friction cones and avoid slipping.

In order to achieve a crawling task, several different control modules had to be brought together. Although computationally intensive, the contact force distribution module was undemanding to implement due to its mostly hardware agnostic nature and low requirements on the feedback (only feet position and IMU readings). At the same time, its role is less important compared to the one reported in [54]. A slippage caused by lower accuracy of contact forces will not necessarily result in robot's failure, like falling down, since the robot is already on the ground. Contact force estimation and control proved to be more difficult to execute on a position controlled robot with a limited sensing. The limitations became apparent when trying to crawl up a slope. Choosing too conservative friction coefficients caused the robot to exceed needed contact force and overload its servo motors. On the other hand, low margins within friction cones cause too much slipping. Although the robot can recover from some slipping, we observed on a slope that if one side of the robot slips more than the other and the robot gets rotated within the pipe, it is difficult to reliably get a stable grip. This is mostly due to our assumption that wall normals are always parallel to robot's sagittal plane. Furthermore a big drawback of the admittance controller is that it only acts on the force perpendicular to the sagittal plane, ignoring shear forces. The decision to do so was based on the assumption that accuracies of force estimation and foot positioning are not high enough to reliably control all the components of contact forces. The final building block for the crawling through curved paths is odometry curvature estimation. Although it lacks precision to build a precise map of the pipe or environment the robot moved through, it proved to be good enough to estimate a local curvature of surrounding geometry. We observed that in some cases, even if the curvature is wrongly estimated, most of the time the robot manages to recover in subsequent steps. Without the curvature estimation, the robot gets easily stuck in sharper pipe turns since it neither adapts its spine bending nor feet trajectories.

8.5.1 Future work

Some of the flaws that were discovered during experiments can be fixed by improving several aspects of both hardware and control. Possibly the least reliable component of the algorithm is force estimation. Given the current hardware, the estimation can be improved by using a more complex neural network which as an input also takes a full state of the limb i.e. joint angles, joint angle reference and current sensed by servo motors. Another possibility is to use a different, more precise force sensor with a simpler mounting solution.

To control robots position within a pipe, only an already existing steering technique presented in Chapter 3 was used. Although effective for following pipe's curvature, due to its non-holonomic behavior, it lacks ability for adjustments in lateral direction. An additional module which would continuously adjust robots posture relative to the walls, whenever there are feet in contact with an environment, would prevent situations where the robot deviates too much from the pipe's midline. Such module could potentially be based on Virtual Model Control (VMC) [154], by defining a set of virtual springs and dampers strategically connecting the robot's body with pipe walls and the estimated midline.

A big discrepancy between simulation and hardware experiments was observed in the robot's performance of crawling up a slope. Actively correcting positioning within a pipe would help with recovering from unwanted slipping. Another option would be to modify the hardware in a way to prevent the slippage in the first place. Assuming the robot can always face a slope with the same end, either by crawling forward or backward, the body surface that is in the contact with the ground could be modified to provide anisotropic friction that would make sliding up the slope easy, but provide high friction in the opposite direction (e.g. by using dry adhesives [105]).

Finally, locomotion through tight spaces and small openings in a less structured environment, possibly present in search and rescue scenarios, would require a modification of the curvature estimation module. Holes or bumps in the side walls could trick the algorithm into detecting a significant curvature even if the walls are straight. A possible solution could be to perform a series of "foot taps" across the wall during the swing phase to increase the number of points used for the curvature estimation.

9 Conclusion

In this thesis we talked about a very specific type of quadruped robot. Inspired by amphibians, reptiles and even their ancestors, these robots are characterized by sprawled posture limbs and an actuated, segmented spine. In robotics, such unique morphology is not common, which makes it an interesting research topic. Wide support on the ground and low body height makes those robots very stable, while the spine and tail can be used for both anguilliform swimming in water and augmenting walking on ground. This opens possibilities for finding an unique niche for real world applications. Observing locomotion of the animals that inspired the robots' mechanical design helps to design controllers that replicate, to some degree, the observed skills. Once the former is achieved, the robots can in return be used as tools to systematically evaluate, quantify and explain the animals' locomotor behavior.

The starting point of our research was Pleurobot - a bio-mimetic salamander-like robot designed to precisely replicate kinematics of salamander walking and swimming gaits (Chapter 2). While its original purpose was to serve as a research tool for neuroscience and biomechanics, we took a more conventional approach to controlling its locomotion (Chapter 3). Relying on inverse kinematics based controllers, we aimed to replicate principles of animal locomotion, mostly the way the spine is coordinated with limbs. The proper coordination can extend the limbs' reach resulting in higher walking speed. The coordination problem is solved by coupling the geometry of feet motion to the girdle orientation. The girdles' orientation, together with an imposed constraint that the hind girdle should follow a path made by the front, defines the spine motion. Such control scheme greatly improves turning capabilities of the robot as the spine is actively used to reduce the turning radius. At the same time, the switching between turning and walking is seamless, which makes the robot easy to pilot, even for inexperienced operators.

As already stated, the sprawling posture is characterized by a low body height from the ground and a wide support polygon. Falling over due to a loss of balance is therefore not a big concern, especially for Krock-2 which can locomote upside down. However, while walking (e.g. utilizing a walking trot gait) the robot's center of mass is not always within the support polygon. As the result, the hips or swing feet tend to make unwanted ground contacts. To eliminate or at least reduce the problem, we looked into how to actively control the center of mass motion to always stay within the support polygon (Chapter 4). Even though this is a simplification of the real cause for the loss of balance, which involves the zero moment point, we observed an increase of balancing performance across several metrics and different gaits. The CoM control is based on a model predictive control framework that relies on predictions of the future support polygon locations.

The locomotion controller of Pleurobot provides quite some freedom in designing the robot's gait. It gives a full control over shaping the feet trajectories, timings of leg motion, oscillation amplitude of the spine as well as fine tuning of leg posture. This allows us to emulate, with a certain degree of accuracy, gaits present in animals that share a similar morphology with the robot. Such idea is explored by using OroBOT - a robot, similar to Pleurobot, but specially designed to mimic the kinematics and estimated weight distribution of the *Orobates pabsti* fossil (Chapter 5). The Orobates is an extinct stem amniote that shares similar morphological features with modern crocodilians or salamanders. An unique opportunity that allowed this research was the discovery of fossilized Orobates trackways, which can be used to extract certain geometrical features of the gait used by the animal. By porting the Pleurobot's locomotion controller to OroBOT, we designed a set of different gaits that comply with the geometrical constraints imposed by the trackways. The OroBOT was used to evaluate the different gaits across three metrics that emerge from the robot's dynamics: efficiency, balance and precision of stepping over the trackways. The dynamic evaluation was used to augment the kinematic analysis of the fossil (e.g. whether the bones collide while executing a gait). By combining both studies we found a most probable subset of gaits the animal might have used. Comparing those gait features to the extant animals, we found that the Orobates might have had a similar locomotor behavior (higher body lift) as a modern caiman.

While exploring possible field applications of Pleurobot-like robots (e.g. in disaster robotics), we were presented with an unique opportunity to build robots that resemble a crocodile and a monitor lizard, for filming purposes in Africa (Chapter 6). As the result, we designed Krock-1 class of robots and built two units - SpyCroc and SpyLizard. Although they share a similar morphology with Pleurobot, the design has been modified for increased robustness, easier handling, maintenance and repairs. Using the robots in Africa for almost two weeks gave us valuable insights into their field performance and overall reliability of the software and control framework. Off-the-shelf Dynamixel servo motors and consumer grade Linux computer Odroid-XU4, that were part of Krock-1 robots, proved to be a good choice for a field robot, as they fared well in harsh conditions involving dust, water and high temperatures.

Learning from our experience with Krock-1 robots, we built an improved version Krock-2 for application in disaster robotics. Krock-2 is equipped with force sensors to estimate foot contact forces, an IMU and a set of cameras. The major change in robot's kinematics is switching the order or shoulder joints (pitch and yaw) that gave the robot a mirror symmetry in the coronal plane. Combined with an IMU, the robot can immediately adapt to being flipped over. The flat profile makes it able to move through low passages and obstacles. We also created a user interface application for a smartphone, to accompany the robot and visualize its posture, sensor data and the cameras feed in real time. We show that using an available multiplatform visualization tools like Unity game engine, can help researchers to rapidly develop interface applications for their robot prototypes.

Disasters like earthquakes, tsunamis or hurricanes often produce a large scale damage to urban structures leaving behind a severely unstructured environment made of debris. Our robots have good mobility on flat ground, however due to their rigid position controlled joints, they lack adaptability to unwanted contacts with an environment. To improve their locomotion over unstructured, rough terrain, we designed additional control modules that augment the locomotion controller and leverage the force sensors and the IMU. It was demonstrated how a stumble and leg extension reflex, in combination with a simple attitude controller, help our robots to (reliably) overcome rough terrain (Chapter 7).

In addition to rough terrain which might still be traversable by utilizing a standard walking gait, narrow cracks, passages and pipes require a different locomotion strategy. We used Krock-2 to demonstrate a new mode of locomotion where the robot crawls through a pipe by pushing against the pipe walls (Chapter 8). Due to the robot's leg kinematics, such strategy allows the robot to move through narrower pipes than previously possible. The crawling controller consists of several modules working together: contact force distribution, estimation and control, crawling state machine, pipe curvature estimation and pipe centerline follower.

Each of the aforementioned ventures provided answers to the main questions addressed in the thesis. The short discussions of those questions are given below.

How to design and test a locomotion controller for a kinematically redundant quadruped robot that fully utilizes its segmented spine and properly coordinates it with the limbs, resulting in increased maneuverability?
The inverse kinematics approach reduced a high dimensional joint space to a lower dimensional Cartesian space of the robot's key points, such as feet and girdles. From there on, designing and generating the key point trajectories is easy as we can seek the inspiration from model animals, whether of periodic (straight walking) or non-periodic (changing direction) behavior. Such purely kinematic approach is sufficient for flat ground locomotion due to benefits of sprawling posture. At the same time, it provides a good basis for additional control modules providing new functionalities.

How to use a bio-informed robotic platform to study locomotion capabilities of an extinct animal?
Here we had access to the fossil of Orobates, which provides enough information to build a robot with similar kinematic properties and mass distribution. A limited amount of data about possible gaits can be obtained from fossil bone motion analysis and fossilized trackways, that provide constraints on gait geometry. Since only a subset of all the possible gaits can be explored, many gait characteristics have to be methodically constrained by observing modern animals that share

similarities with the fossil. Metrics used to evaluate the gaits have to be carefully constructed to have a biological significance while being applicable to the robot. The most plausible subset of gaits is determined by excluding gaits with low scores across multiple metrics. Such ranking based exclusion reduces the sensitivity to noise in a score within individual metric (e.g. small changes in uncertain gait parameters do not have a big impact on the final result).

Which design modifications and control add-ons can give these robots unique capabilities in a field of disaster robotics? Decreasing a number of actuated degrees of freedom, while keeping a full controllability over the key points (feet, girdles), reduces robot's weight and complexity. Relying on simple and cheap manufacturing and assembly allows fast and easy field repairs and maintenance. Additional sensors and feedback driven control (e.g. contact based reflexes), that adapts and reacts to the environment, enable robots to overcome more complex terrains. Moreover, modified leg kinematics and contact force sensors are used to achieve unconventional modes of locomotion (pipe crawling), previously not explored with a sprawled bio-mimetic robot.

9.1 Future steps

We worked with four different, yet similar robots: Pleurobot, OroBOT, Krock-1 and Krock-2, that were used to tackle a variety of different questions and problems. This gave us a good insight into their capabilities and limitations. All the knowledge and experience acquired throughout the work presented in this thesis, can be used to ask new questions and propose further research directions. Possible future steps that can both deepen and broaden the scope of our research on sprawling posture robots with an actuated spine are given in the following.

All the robots that we have used share the same actuation system based on high-end Dynamixel servo motors. They were used as a black box — robot's control framework sends them an angle position reference, whose following is solved by their internal control algorithms. This is aligned with the inverse kinematics based approach to robot control. However, Dynamixel servo motors also provide limited torque control mode. Although such feature has already been available for models MX-64 and MX-106, a recent firmware update gives users an access to more internal parameters. Exploring and enabling torque control capabilities of these robots would bring new possibilities. Limiting the joint torque and achieving motor backdrivability, would make the robot more compliant and adaptable to the environment. The precise position control of the robot's body could be replaced with more reactive control methods that rely on a sensory feedback. Although the lack of precision in control would be a problem for upright posture robots that could easily lose balance, sprawled robots are ideal candidates for such control approach. The joint compliance would prevent motor overloads, that often happen while in the position control mode, due to unwanted contacts with obstacles or badly estimated contact force.

The motor overloads proved to be a problem in the pipe crawling experiments with Krock-2, where they would often shut down upon errors in the contact force estimation. The torque-controlled limbs would make it easier to control the foot contact force. However, the main principle behind the pipe crawling would still rely on pushing against walls with enough force to overcome the external forces acting on the robot's body (e.g. gravity). This limits a maximum slope the robot could climb. A possible solution would be to equip the feet with dry adhesives used by gecko-inspired climbing robots [105]. Such adhesives provide a high adhesion without a need for high normal forces acting on the climbing surface. To crawl through a vertical pipe, the robot would only need to lift its own weight. As the 4 DoF limbs of Krock-2 provide a limited control of the foot angle, a possible research direction would be to focus on the foot design, e.g. relying on dry adhesives and claws, that would enable the robot to crawl over highly inclined surfaces or unstructured obstacles.

Even though the segmented spine and tail enable our robots to swim, their aquatic capabilities were not properly explored in this thesis. This is mostly because of difficulties in making them waterproof. The latest waterproof suit, built by a professional tailoring workshop, seems promising, but it requires further testing and improvements. There is a constant trade-off between the suit robustness (higher thickness, multiple layers) and flexibility to avoid hindering robot's motion. Ideally, the suit should have a minimal impact on the robot's terrestrial locomotion, while still providing a reliable protection against water. Besides constantly improving the design, future steps could involve equipping the suit with additional water detection sensors that could automatically trigger transitions between walking and swimming. To prepare such multimodal behavior for the real world application, we should solve the problem of transitioning out from water when the water-land boundary (e.g. a shore) is not smooth and gradual.

Although our work on reconstructing a gait of an extinct animal using OroBOT was fairly comprehensive, there are several possible improvements to the study. The evaluation of the kinematic metric (bone motion analysis) was confirmed on a modern animal (caiman) and the same should be done for the dynamic metrics involving the robot. For example, Pleurobot could be used to find the most probable gait of a salamander *Pleurodeles waltl*, which inspired the robot's bio-mimetic design. Since salamanders are still around and we have a possibility to collect all the needed data about the their gaits, this could either validate our approach (in particular, our choice of metrics) or help us with refining it.

9.2 Final words

Addressing the questions in this thesis required constant development and maintenance of robot hardware, software and control algorithms over more than four years. This was possible by relying on widely used, off-the-shelf components. Instead of developing them from scratch, most of the effort could be put into their integration into a functional robot. Moreover, the knowledge and developed framework could be reused while transitioning from one robot to another. Even though these robots were created and used for different purposes (a tool for neuroscience, studying a fossil or in disaster robotics), they all share the same core architecture. As a consequence, the decisions made in the beginning, like the choice of actuators or relying on the inverse kinematics, were kept through our work. This means the paths we have taken to tackle the problems and questions are not the only possible or even the optimal ones. The future steps that are proposed here are just some of the possible further explorations and ways to improve our robots.

The legacy of our work will hopefully surpass the scope of this thesis. Bringing a robot to life and shaping it into a pleasant to work with research tool requires way more than just tossing control algorithms on it. Our entire software framework was built to accelerate deployment of new control algorithms, allow future expandability, and provide a user friendly interface that makes carrying out robot experiments faster and easier. The developed software framework has already been reused on a swimming lamprey-like robot and a cat-like quadruped. The collected experience and learned lessons from our robot deployment in Africa and orientation towards building field-ready robots could encourage future researchers to test their robots more often outside the laboratory environment.

A Supplementary Information: Orobates and OroBOT

A.1 OroBOT leg control

A solution for the leg inverse kinematics problem is a set of leg joint angle trajectories $\boldsymbol{q} \in \mathbb{R}^{4x1}$ that bring the foot attachment point (i.e. leg wrist/ankle) to a desired position defined as Cartesian coordinates $\boldsymbol{p}_r \in \mathbb{R}^{3x1}$. The position $\boldsymbol{p}_r$ does not depend on wrist/ankle joint angle $q_W \in \mathbb{R}$, thus only the first four leg joints are considered to be part of the kinematic chain, excluding the foot orientation. A role of the wrist/ankle joint is to keep the foot aligned with a line of locomotion, meaning it has to compensate for the foot rotation around the vertical axis. If $\boldsymbol{R}_F \in \mathbb{R}^{3x3}$ is a matrix describing the foot orientation in robot's frame of reference, then q_W is calculated as follows:

$$q_W = -\text{atan}\frac{\boldsymbol{R}_{F1,2}}{\boldsymbol{R}_{F2,2}}. \tag{A.1}$$

The leg inverse kinematics, the first four joints in the kinematic chain, is solved as described in Chapter 3.

A.2 OroBOT spine control

Since all the trunk joints revolute about parallel axis, the trunk motion is constrained to a plane (i.e. co-planar kinematic chain). Let's denote the front and hind girdle positions in the plane with $\boldsymbol{p}_F$, $\boldsymbol{p}_H \in \mathbb{R}^{2x1}$, and the orientations around the vertical axis with ϕ_F, $\phi_H \in \mathbb{R}$. The trunk inverse kinematics problem is then defined as follows: find the trunk joint angles $\boldsymbol{q}_T \in \mathbb{R}^{4x1}$ for which positions $\boldsymbol{p}_F$, $\boldsymbol{p}_H$ and orientations ϕ_F, ϕ_H are as close as possible to desired values. The problem is solved as described in Chapter 3.

A.3 OroBOT foot trajectory

A foot trajectory is a closed curve, defined in the robot frame of reference. The overall robot motion is the consequence of all foot executing such trajectories at their respective timing. Since the OroBOT's feet are complex, with multiple points touching the ground simultaneously, the point that is commanded to follow the trajectory is the attachment point of a foot (i.e. wrist/ankle). The leg trajectory is split into a segment where a foot is in ground contact - a stance phase, and a segment when the foot is in the air - a swing phase.

All the main parameters that determine the trajectory are show in Fig. 5.7. The stance segment is defined with starting and ending points $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$. It is parallel to the line of locomotion, meaning the points $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are at the same distance W from the girdle along y-axis. Towards the end of stance, following an exponential law, the trajectory raises for a total amount of h along the z-axis. Such shape is inspired by animals which end their stance by having only fingers touching the ground.

The swing trajectory is a three dimensional bezier curve connecting points $\boldsymbol{T}_2$, $\boldsymbol{T}_3$ and $\boldsymbol{T}_1$. Using a bezier curve allows for defining touch-down and lift-of angles α_1 and α_2. The point $\boldsymbol{T}_3$ can be placed at the different distance along the y-axis from the girdle compared to points $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$. The difference w defines the swing width.

The parameters that do not influence the shape, but relative position of the trajectory with respect to the girdle are: stance width W, body lift B, stance length L and longitudinal offset Δx (Fig. 5.7).

A.4 OroBOT spine trajectory

The front girdle position is considered to be a base of the kinematic chain (trunk), while the hind girdle position is constrained by its placement on the line of locomotion and constant length of spinal segments. Thus, the spine trajectory is only defined with a time evolution of angles ϕ_F and ϕ_H as follows:

$$\begin{aligned} \phi_F &= A_s \sin(2\pi f(t+\theta_S)), \\ \phi_H &= A_s \sin(2\pi f(t+\theta_S)+\pi), \end{aligned} \tag{A.2}$$

where A_s is the amplitude of girdle oscillations, f is the walking frequency and θ_S is the phase offset between the spine and legs.

A.5 Temporal gait parameters of OroBOT

The four legs, repeatedly following foot trajectories, together with a spine bending constitute OroBOT's gait. Other than on the trajectory geometry, the gait depends on several temporal parameters.

A relative duration of stance and swing phase is defined by a duty factor D. Let $\theta_i \in [0, 1 >]$ be a phase of i-th leg, defining the foot's location on the trajectory. For $0 \leq \theta_i < D$, the leg is in the stance, and for $D \leq \theta_i < 1$ the leg is in the swing. The phase is calculated as follows:

$$\theta_i = \mathrm{mod}(f \cdot t + \theta_{off,i}, 1), \tag{A.3}$$

where $\theta_{off,i}$ is i-th leg phase offset. The robot's forward velocity is assumed to be constant, thus there is a linear mapping between the phase and the trajectory during stance. The duty factor and a phase offset between legs determine how many legs are in stance at any given moment.

A.6 Main gait characteristics of OroBOT

The main characteristics used to describe OroBOT's gait are: spine bending, body lift and long-axis rotation (LAR) (Fig. 5.3).

The spine bending is represented as a maximum girdle rotation measured around the vertical axis. It is directly linked to the amplitude of girdle oscillations A_s, which is a parameter used in the spine controller. The same amplitude is used for both front and hind girdle.

The body lift B is a foot trajectory parameter that sets the distance between the initial point of the stance trajectory and the girdle, measured along the vertical axis. The same value of B is used to define the foot trajectory for all the legs.

The long-axis rotation is the only gait characteristic that is not directly represented with a single gait parameter. An amplitude of long-axis rotation over a single stride is not know in advance. It is measured once an entire walking cycle is executed. The parameter that has the highest influence on LAR is the weight matrix M used in the cost function of the leg inverse kinematics optimization problem. The matrix M is a diagonal matrix with positive elements on the diagonal: $M = \mathrm{diag}(m_1, m_2, m_3, m_4)$. Each element m_i represents a relative weight of how well the i-th leg joint tracks the desired joint angle $\Delta q_{r,i}$. A higher value of m_i means the corresponding joint is more likely to stay close to the desired angle, compared to the joints with the lower weight. If the desired angles $\Delta q_{r,i}$ are set to a constant value, the higher weight penalizes movement of the joint. In case of OroBOT, parameter $\boldsymbol{\Delta q}_r$ equals to the zero vector which corresponds to the robot's posture shown in Fig. 5.4D.

The LAR in animals is complemented by a shoulder protraction/retraction (P/R). In the robot, the P/R and LAR are represented as rotations of the 1st and 3rd leg joints respectively. To determine which of those joints is dominantly used during locomotion, a matrix M is parametrized as follows:

$$M = \mathrm{diag}(\sqrt{10^{m_{LAR}}} \cdot m_0, m_0, 1/\sqrt{10^{m_{LAR}}} \cdot m_0, m_0), \tag{A.4}$$

where m_{LAR} is the parameter that sets a relative usage between LAR and P/R. For example, if $m_{LAR} = 1$, the P/R motion is 10 times more penalized compared to LAR. The opposite is achieved for $m_{LAR} = -1$. For $m_{LAR} = 0$, all the leg joints have the same basic weight m_0.

Having all the other parameters fixed, each of the robot gaits is defined by a triplet (m_{LAR}, A_s, B) that directly or indirectly influence the main gait characteristics. The set of all triplets is a cube:

$$S = \{(m_{LAR}, A_s, B) : m_{LAR,min} \leq m_{LAR} \leq m_{LAR,max}, A_{s,min} \leq A_s \leq A_{s,max}, B_{min} \leq B \leq B_{max}\}. \tag{A.5}$$

The boundaries (minimum and maximum parameter values) are chosen to cover entire range of physically possible gaits:

$$\begin{array}{ll} m_{LAR,min} = -1, & m_{LAR,max} = -1, \\ A_{s,min} = 0 \text{ [rad]}, & A_{s,max} = 1 \text{ [rad]}, \\ B_{min} = 9 \text{ [cm] or } 0.21 \text{ IGD}, & B_{max} = 27 \text{ [cm] or } 0.62 \text{ IGD}, \end{array}$$

For example, gaits with body lift smaller than B_{min} would result in feet not touching the ground. Values of m_{LAR} outside the given range are still plausible, but do not produce significantly different gaits compared to the ones within the range. Due to a limited computational resources, robot's performance can only be evaluated on a finite set. Let's denote such set with S_N, which contains points on a regular $N \times N \times N$ mesh grid.

A.7 Constraints to the OroBOT gaits imposed by the trackways

The Orobates trackways (Fig. 5.4) provide geometric constraints for OroBOT feet trajectories during stance phase. The measured values are stride length, stride width and ipsilateral footprint distance. The trackways are not completely periodic, therefore the mean values are used.

Stride length L_{str} is the distance between consecutive steps of the same leg. In periodic gaits, it is equal to the displacement of robot body during a single step. The stance length L is equal to the body displacement during stance phase. Since robot velocity is constant, the stance and stride lengths are linked through the duty factor D:

$$L = L_{str} \cdot D. \tag{A.6}$$

Thus, trackways are used to fix the value of the stance length.

Stride width $W_{str}$is the distance between contralateral footsteps along the line perpendicular to the direction of locomotion. Assuming the gait is laterally symmetrical, the stride width is used to fix the value of trajectory width:

$$W = \frac{W_{str}}{2}. \tag{A.7}$$

The ipsilateral footprint distance Δ_{ips} is the shortest distance between footprints of ipsilateral feet, measured along the line of locomotion. It is a function of robot intergirdle distance IGD, the difference between the trajectory offsets of the front Δx_F and hind leg Δx_H (Fig. 5.7), stride length $L = L_{str}$ and ipsilateral phase difference $\Delta\phi$:

$$\Delta_{ips} = \text{IGD} + (\Delta x_F - \Delta x_H) - L_{str}(1 - \Delta\phi). \tag{A.8}$$

A.8 Defining open gait parameters

Several key parameters that describe robot gait are not constrained by the trackways. For example, the trackways provide no data about possible swing trajectories. The only criteria is for the foot to have enough ground clearance to avoid ground contacts. Thus the trajectory height H is set to 12 [cm] for all the legs. The trajectory width w is set to 6 [cm] to keep the trajectory within leg's operational space.

Unlike swing, the stance trajectory is mostly defined by the trackways. From them, it is visible that feet had full contact with the ground, as opposed to walking on the toes. However, the touch-down angle α_1 and lift-off angle α_2, as well as lift-off height h are not defined. Inspired by a "tear" shaped extant animal trajectories, the angles are set to $\alpha_1 = 135^\circ$ and $\alpha_2 = 45^\circ$. The lift-off height h is set to 30% of the total trajectory height H, resulting in $h = 4$ [cm]. The exponential time constant that determines how fast the stance trajectory approaches its final height is set to 10, meaning 63% of the lift occurs in the last 10% of the stance.

The trackways provide a constraint on the relative difference between trajectory offsets $\Delta X = \Delta x_F - \Delta x_H$, however the absolute values of Δx_F and Δx_H are not fixed. The value of ΔX is negative meaning either the front feet trajectories need to be shifted backward, or the hind should be shifted forward. It is observed that the robot, due to its mass distribution tends to tilt backwards (ZMP outside its support polygon) and shifting the hind trajectory forward would increase the tilting. Therefore the entire relative difference ΔX is contained within Δx_F:

$$\begin{aligned} \Delta x_F &= \Delta X + \Delta x_0, \\ \Delta x_H &= \Delta x_0, \end{aligned} \tag{A.9}$$

where Δx_0 is the common trajectory shift. The procedure fixing its value is explained in the following subsection.

The trackways provide no information on possible temporal parameters. Due to a simplicity, the phase offsets between legs are chosen to keep diagonal leg pairs in phase, while contralateral and ipsilateral pairs are in counterphase: $\theta_{off} = [0, 0.5, 0.5, 0]$. The resulting gait is classified as a walking trot. The duty factor D is set to 0.75, which makes the duration of a swing three times shorter than stance.

A.9 Trajectory shift and feet parameters

As the trajectory shift Δx_0 and feet parameters have a high impact on the robot performance it is needed to carefully pick their values. The passive feet joints are modeled as spring-joint systems with stiffnesses $k_{f,1}$, $k_{f,2}$, $k_{f,3}$ and dampings $d_{f,1}$, $d_{f,2}$, $d_{f,3}$. The motion of each joint in a free space is modeled as a 2nd order dynamical system:

$$\ddot{\gamma} + \frac{d_{f,i}}{I_i}\dot{\gamma} + \frac{k_{f,i}}{I_i}\gamma = 0, \tag{A.10}$$

where γ_i is the angle of i-th foot joint and I_i is its inertia. The natural frequency ω_n and the damping ratio ζ of such system are defined as:

$$\begin{aligned} \omega_n &= \sqrt{\frac{k_{f,i}}{I_i}}, \\ \zeta &= \frac{d_{f,i}}{2\sqrt{k_{f,i} I_i}}. \end{aligned} \tag{A.11}$$

A reasonable decision is to make the passive joints overdamped ($\zeta > 1$)which prevents oscillatory behavior. Thus we chose $\zeta = 2$, which is used to express the joint damping as a function of the stiffness:

$$d_{f,i} = 2\zeta\sqrt{k_{f,i} I_i}. \tag{A.12}$$

Further simplification is made by expressing stiffnesses $k_{f,2}$ and $k_{f,3}$ via $k_{f,1}$:

$$\begin{aligned} k_{f,2} &= \frac{k_{f,1}}{4}, \\ k_{f,3} &= \frac{k_{f,1}}{2}. \end{aligned} \tag{A.13}$$

Finally, let's denote a set of possible Δx_0 and $k_{f,1}$ combinations with

$$P = \{(\Delta x_0, k_{f,1}):\ \Delta x_{0,min} \leq \Delta x_0 \leq \Delta x_{0,max},\ k_{f,1} \leq k_{f,1,min} \leq k_{f,1,max}\}. \tag{A.14}$$

Since there is no clear criteria of how to select their values, we evaluate robot's performance for different values of those parameters and pick the ones which result in the highest performance. The evaluation is done in multiple points of the gait set S, to avoid bias towards a single sub-region. As finite sets we chose S_4 and P_{11}. The combined evaluation set $S_4 \times P_{11}$ contains 7744 points ($4 \times 4 \times 4 \times 11 \times 11$).

The score of each point in P_{11} is calculated as follows: (i) assign each gait a score equal to a

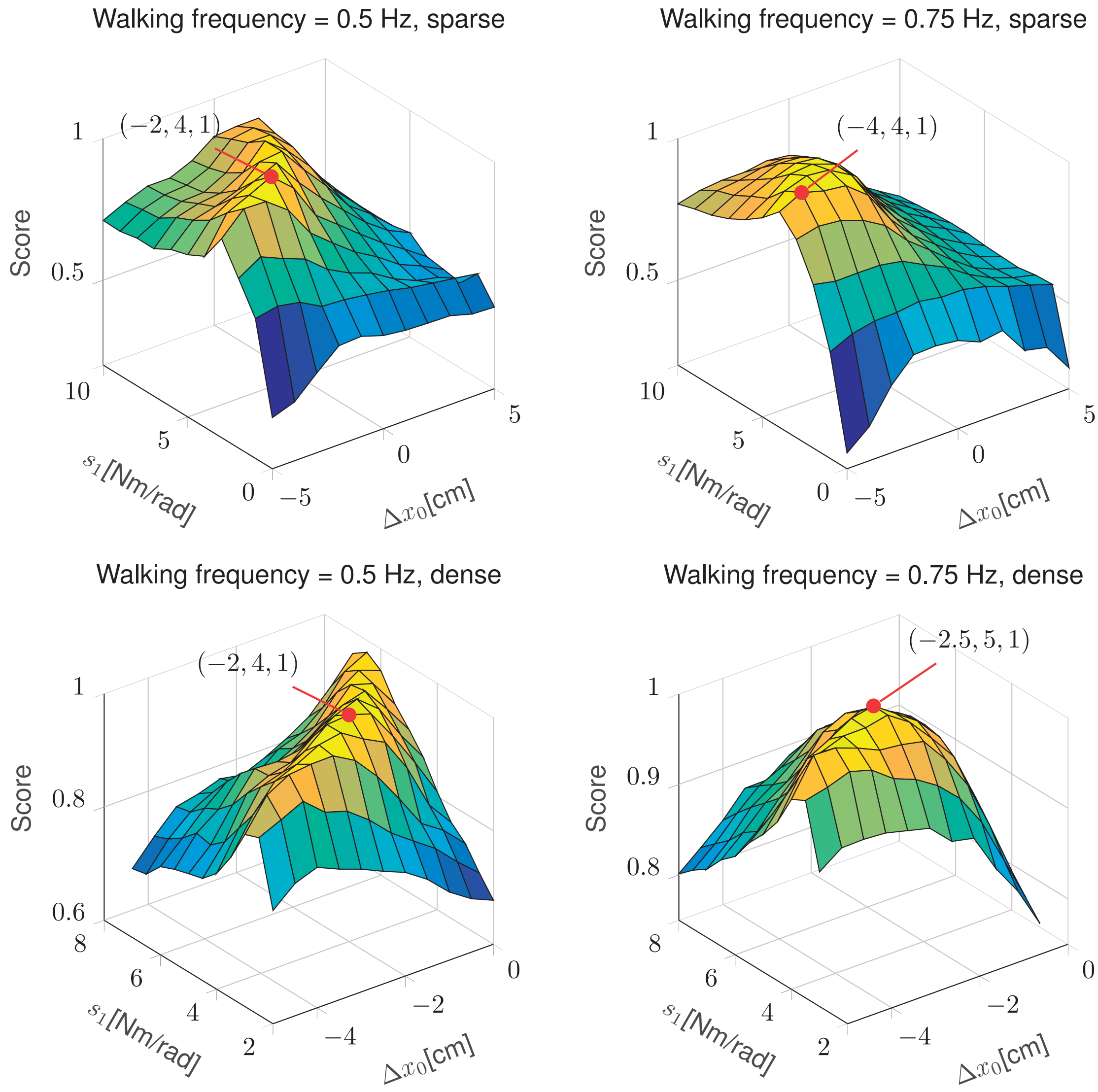


Figure A.1 – Exploration of the optimal foot stiffness and trajectory offset values. The exploration was done in two steps: on the coarse grid of foot parameters to get a region of the optimum (top) and on the dense grid to refine the optimum (down). The process was repeated for two frequencies: 0.5 [Hz] (left) and 0.75 [Hz] (right).

geometric mean of the main metrics scores, (ii) sum the scores across all the gaits in S_4. The procedure is repeated for walking frequencies of 0.5 [Hz] and 0.75 [Hz] (Fig. A.1). Finally, we use the values Δx_0 and $k_{f,1}$ corresponding to the point in P_{11} with the highest score:

$$\Delta x_0 = -2 \text{ [cm]}, \qquad k_{f,1} = 4 \text{ [Nm/rad] @ 0.5, [Hz]}\,,$$
$$\Delta x_0 = -2.5 \text{ [cm]}, \qquad k_{f,1} = 5 \text{ [Nm/rad] @ 0.5, [Hz]}\,.$$

A.10 Metrics definition

The robot performance using a certain gait is described using several criteria (metrics) which are relevant in robotics but also have biological significance.

A.10.1 Efficiency

Mechanical power in i-th rotational joint over a time period $T_0 \leq t \leq T_F$ is given by:

$$P_i = \frac{1}{T_F - T_0} \int_{T_0}^{T_F} \tau_i \omega_i dt, \tag{A.15}$$

where τ_i is the torque and ω_i corresponds to the joint angular velocity. Given that the body can rotate at a certain speed whose direction is contrary to that of the torque applied, the previous definition encapsulates cases of a negative power where an environment does work on the mechanical system.

As OroBOT's joints are actuated by position controlled servos which are not capable of regenerating power (energy) over instances of negative work, the following power description is more accurate:

$$P_{a,i} = \frac{1}{T_F - T_0} \int_{T_0}^{T_F} |\tau_i \omega_i| dt. \tag{A.16}$$

A compromise between the two power definitions, where the first one has a higher biological significance, while the second better fits the robotic system, is to account only for the instances of positive power:

$$P_{p,i} = \frac{1}{T_F - T_0} \int_{T_0}^{T_F} \tau_i \omega_i dt, \quad \forall \tau_i \omega_i > 0, \quad \text{otherwise } 0. \tag{A.17}$$

Animals have to use metabolic energy to produce torque in their joints and also to move bones fast. Even though passive mechanisms are known, these probably did not apply to early tetrapods. Focussing on the positive aspect of torque x velocity thus can be justified.

Finally, the efficiency metrics for the entire robot that has 28 actuated joints is defined as a reciprocal of the power measure:

$$M_E = 1 / \sum_{i=1}^{28} P_{p,i}. \tag{A.18}$$

A.10.2 Balance

OroBOT feet trajectories are constructed in a way the robots body, from a purely kinematic point of view, should always stay parallel to the ground. Taking into account the dynamics, this is not the case. If robot's zero moment point (ZMP) falls outside the support polygon formed by the feet on ground, the resulting torque tilts the robot. Due to a nature of sprawling locomotion, it is difficult for the robot to get completely flipped over by such tilting, that could be characterized as a fall. Even though it does not end with a fall, the amount of tilting indicates how closely the support polygons capture the ZMP, which is often a design goal of advanced robot controllers. Biological significance comes from the desire of animals to minimize compromising effects on vestibular and optical perception. Thus, the following measure is derived:

$$M_B = 1/\left(\frac{1}{T_F - T_0}\int_{T_0}^{T_F} |\dot{u}| + |\dot{v}| dt\right), \tag{A.19}$$

where $\dot{u}$ is the roll rate measured around x-axis, and $\dot{v}$ the pitch rate measured around y-axis of the robot.

A.10.3 Precision

To define how accurate is the foot placement corresponding to the desired foot trackway target, we use the following algorithm:

1. Measure ground projections of robot's feet during mid 70% of stance phase.
2. Align the centerline of projections with idealized trackways (by using a linear regression).
3. Formulate and solve a QP optimization problem that minimizes the distance between the corresponding projected and idealized steps.

The steps are visualized on an example in Fig. A.2. The remaining distance (value of the cost function) after the optimization is used as a precision metric M_P.

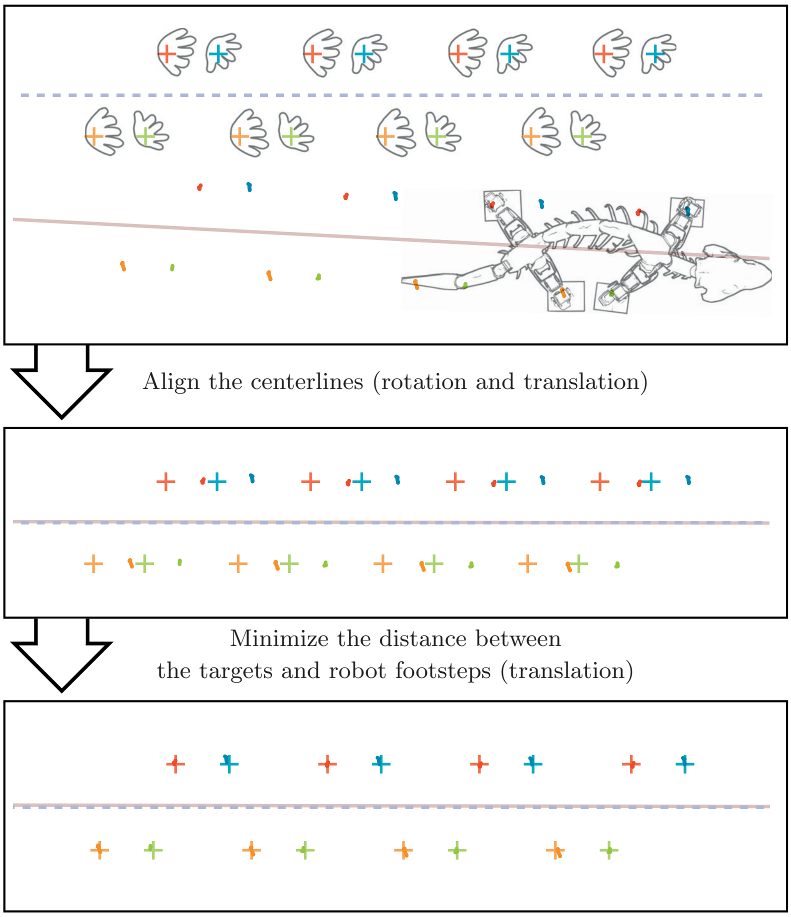

Align the centerlines (rotation and translation)
Minimize the distance between
the targets and robot footsteps (translation)

Figure A.2 – Computation of the precision metrics. (Top) The idealized trackways and the robot footsteps extracted from Webots simulation were not necessarily aligned in the world reference frame, since the robot did not use path-following strategies. (Middle) The trackways and the footsteps were approximately aligned by matching their centerlines via translation and rotation. (Bottom) A precise alignment was done by translation, whose amount was determined through an optimization that minimized distances between the corresponding footsteps. The remaining distances were summed and used as a measure of precision.

A.11 Gait exploration and evaluation

Each main gait parameter was set to eight different values, thus making a parameter set S_8 containing 512 different gaits. The evaluation procedure for k-th gait is:

1. Simulate the robot until it makes 13 steps and record the data required for the metrics calculation.
2. Remove the data from the first 3 steps to ignore effects of transitioning from standing to walking.
3. Calculate metrics $M_{E,k}$, $M_{B,k}$ and $M_{P,k}$.

Resulting metrics scores M_E, M_B and M_P can be visualized in a three dimensional parameter space (Fig. 5.4). Instead of having m_{LAR} as one of the space axis, an amplitude of long axis rotation A_{LAR} is used, thus enabling a direct comparison with the kinematic Maya simulation.

It is important to emphasize that all of the 512 gaits are feasible to execute on the robot. The dynamic simulation includes realistic servo motor torque and angle limits. If the feet trajectories are impossible to follow (due to angle limits) in a certain gait, the gait is nevertheless executed, although with modified trajectories. Any deviation from a commanded trajectory results with a lower score in the absolute precision metrics.

B Supplementary Tables: Orobates and OroBOT

Table B.1 – Analyzed experimental subjects. TL: total length; IGD: inter-girdle distance (gleno-acetabular distance). *For comparison also, data for the holotype specimen of *Orobates pabsti* measured using the digital reconstruction of the holotype specimen are provided (from [139]).

Species	Sex	Mass [g]	TL [mm]	IGD [mm]
Mexican salamander (*Ambystoma mexicanum*) hindlimb to snout vent length ratio: approx.0.3 forelimb to hindlimb length ratio: approx.0.85	Female	58.9	194	82
	Female	72.6	194	81
	Male	66.5	205	88
	Male	67.6	239	92
Blue tongued skink (*Tiliqua scincoides*) hindlimb to snout vent length ratio: approx.0.17 forelimb to hindlimb length ratio: approx.1.0	Female	1,150	560	240
	Male	965	540	210
Green iguana (*Iguana iguana*) hindlimb to snout vent length ratio: approx.0.75 forelimb to hindlimb length ratio: approx.0.67	Female	1,450	1160	190
	Male	2,010	1250	210
Spectacled caiman (*Caiman crocodylus*) hindlimb to snout vent length ratio: approx.0.39 forelimb to hindlimb length ratio: approx.0.75	Female	10,010	1260	360
	Male	9,700	1370	380
Orobates pabsti* (MNG 10181) hindlimb to snout vent length ratio: approx.0.4 forelimb to hindlimb length ratio: approx.0.9	Unknown	3,981	851	273

Table B.2 – Fossil / robot dimensions. Length comparison between different body segments in Orobates and OroBOT. Organized by columns: (i) Orobates dimensions. (ii) Orobates with an initial head scale of 1.4. (iii) Orobates with a body scale of 1.6 (no head additional scaling). (iv) Final desired lengths of Orobates head and body scaled. (v) Segments to body length ratios. (vi) Final OroBOT dimensions. (vii) OroBOT segments to body length ratios. (viii) Mass distribution ratio differences between scaled Orobates and OroBOT. Data for Orobates taken from [139]. Trunk consisted of pectoral girdle, spine segments and pelvic girdle. Girdles and spine segments are depicted in light green. CoM of OroBOT was calculated using the CAD files in Autodesk Inventor® 2017 and confirmed by manual measurements.

	Orobates [cm]	x1.4 Head [cm]	x1.6 Scale [cm]	x1.4 Head + x1.6 Scale [cm]	Orobates [%]	OroBOT [cm]	OroBOT [%]	Differences [%]
Head	11.14	15.6	17.82	24.95	17.41	23.99	16.87	0.54
Neck	4.24	4.24	6.79	6.79	4.73	9.65	6.78	-2.05
Pectoral Girdle	7.96	7.96	12.73	12.73		13.05		-0.29
Spine	19.65	19.65	31.45	31.45		27.08		2.9
Pelvic Girdle	8.1	8.1	12.95	12.95		15.3		-1.72
Trunk	35.71	35.71	57.13	57.13	39.85	55.43	38.98	0.88
Tail	34.05	34.05	54.48	54.48	38	53.15	37.37	0.63
Total length	85.14	89.6	136.22	143.35	100	142.23	100	0
Pectoral G. Width	6.75	6.75	10.8	10.8	10.8			
Front Limb	18.67	18.67	29.87	29.87	31.47			
Pelvic G. Width	6.07	6.07	9.71	9.71	9.71			
Hind Limb	20.65	20.65	33.05	33.05	33.46			
IGD	27.3		43.68			43.68		
SVL	51.09		88.87			89.08		
CoM	12.36		19.77			19.8		

Table B.3 – Fossil / robot mass distribution. We used data for Orobates from [139]. We averaged maximum and minimum plausible values of the mass of different body segments and calculate their ratios against the total body mass. Same procedure was done for OroBOT. Differences between the mass ratios are reported.

	Orobates mass [kg]			*Orobates* [%]	OroBOT mass [kg]	OroBOT [%]	Differences [%]
	Max	Min	Avg				
Head	0.213	0.213	0.213	5.35	0.389	6.29	0.93
Neck	0.260	0.113	0.187	4.68	0.232	3.75	-0.94
Front limb	0.273	0.182	0.228		0.359		0.0
x2	0.546	0.364	0.455	11.43	0.718	11.60	0.17
Trunk	2.749	1.369	2.059	51.72	3.224	52.09	0.37
Hind limb	0.447	0.255	0.351		0.541		0.0
x2	0.894	0.510	0.702	17.63	1.082	17.48	-0.15
Tail	0.549	0.182	0.366	9.18	0.544	8.79	-0.39
Total	5.211	2.751	3.981	100.0	6.189	100.0	0.0

Table B.4 – OroBOT mass breakdown. We used data for Orobates in [139]. We averaged max and min plausible values of the mass of different body segments and calculate their ratios against the total body mass. Same procedure was done for OroBOT. Differences between the mass ratios are calculated. We also present a mass breakdown of the different parts in OroBOT, where the mass segments correspond to a 3D printed structural part with their associated servomotor. This is the case for neck, spine and tail segments. Similarly, the head segment contains also the mass of the robot's computer and other electronic peripherals (i.e regulator, communication, Bluetooth and Wi-fi modules). In the case of the limbs, they are composed by individual servo motors, attached to each other by small 3D printed plastic light parts i.e. Humerus/Femur, aluminum parts i.e. Pes/Crus, and their corresponding plastic feet. The animal has a clear difference between front and hind limb mass. Which is noticeable also by the cross sectional areas of the humerus/femur. This difference was hard to replicate with the materials and dimensions used in the robotic reconstruction (i.e. similar fabrication and mass for front and hind limbs). Consequently, the mass of the first and second shoulder servo motors in the forelimbs were associated with the front girdle mass, and considered as part of the trunk. In the case of the hindlimbs, only the first shoulder servo motor was considered part of the hind girdle, hence part of the trunk as well.

Body part	Mass [g]		
Head		389.0	
Neck		232.0	
Neck MX64		142.0	
Pectoral girdle +2x yaw MX64		736.0	
2x pitch MX64		284.0	
2x roll MX64		284.0	
Humerus	39.0		39.0
Elbow MX64	142.0		142.0
Ulna	48.0		48.0
Wrist MX28	80.0		80.0
Front foot	50.0		50.0
Spine 1		259.0	
Spine 2		250.0	
Spine 3		220.0	
Pelvic girdle +2x yaw MX64		765.0	
2x pitch MX64		284.0	
Femur +roll MX64	201.0		201.0
Knee MX64	142.0		142.0
Crus	48.0		48.0
Ankle MX28	80.0		80.0
Hind foot	70.0		70.0
Tail 1		37.0	
Tail 2		246.0	
Tail 3		231.0	
Tail tip		30.0	

Table B.5 – Gait parameters and calculated Froude Numbers of different sprawling posture species. (a) Spectacled caiman (*Caiman crocodylus*). (b) Green iguana (*Iguana iguana*). (c) Blue-tongued skink (*Tiliqua scincoides*). (d) Mexican salamander (*Ambystoma mexicanum*). (e) Averaged data for salamander (*Pleurodeles waltl*) [98]. (f) Forelimb and (g) hindlimb of Leopard gecko (*Eublepharis macularius*) [93]. (h) Low speed sprawling posture, (i) high speed sprawling posture, (j) low speed high walk posture and (k) high speed high walk posture of alligator (*Alligator mississippiensis*) [159]. Savannah monitor lizards (*Varanus exanthematicus*) [169]. (l) lizard 1, (m) lizard 2, (n) lizard 3, and (o) lizard 4. (p) Forelimb and (q) hindlimb of Leopard gecko (*Eublepharis macularius*) [92]. (r) Male and (s) female Blue-tongued skink (*Tiliqua scincoides*) [35]. The Froude number was calculated as hf^2/g, where h is the stride length, f is the gait frequency and g the gravity.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
Limb phase (LP)	0.44	0.42	0.44	0.36	0.70	-	-	0.53	0.37	0.48	0.57	-	-	-	-	-	-	-	-
Duty factor (D)	0.61	0.63	0.63	0.82	0.79	0.70	0.75	0.74	0.69	0.71	0.70	-	-	-	-	0.70	0.78	-	-
Stride length [cm]	52.9	28.5	13.4	6.0	6.6	8.5	8.9	13.9	17.9	11.6	16.1	31.2	34.0	22.3	30.1	6.6	6.5	12.5	11.0
Stance time [s]	1.65	0.98	0.46	0.86	1.11	0.33	0.43	1.37	0.86	1.09	0.78	-	-	-	-	0.54	0.64	-	-
Swing time [s]	1.06	0.58	0.27	0.19	0.30	0.14	0.14	0.50	0.39	0.44	0.33	-	-	-	-	-	-	-	-
Stride time [s]	2.71	1.56	0.73	1.05	1.41	0.47	0.57	1.89	1.25	1.54	1.11	-	-	-	-	0.77	0.82	-	-
Speed [cm/s]	19.5	18.3	18.4	5.8	4.7	18.1	15.5	7.4	14.6	7.4	14.6	22.0	25.0	25.0	23.5	-	-	8.0	8.0
Frequency [Hz]	0.37	0.64	1.37	0.95	0.71	2.12	1.74	0.53	0.80	0.65	0.90	0.71	0.74	1.12	0.78	1.30	1.22	0.64	0.73
Froude Number	0.74	1.20	2.57	0.56	0.34	3.92	2.76	0.40	1.17	0.50	1.33	1.58	1.88	2.86	1.87	1.13	0.98	0.52	0.59

Bibliography

[1] Ajallooeian, M., Gay, S., Tuleu, A., Sprowitz, A., and Ijspeert, A. J. Modular control of limit cycle locomotion over unperceived rough terrain. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (2013), IEEE, pp. 3390–3397.

[2] Alexander, R. M. *Principles of animal locomotion.* Princeton University Press, 2003.

[3] Alexander, R. M., and Jayes, A. A dynamic similarity hypothesis for the gaits of quadrupedal mammals. *Journal of zoology 201*, 1 (1983), 135–152.

[4] Andrada, E., Nyakatura, J. A., Bergmann, F., and Blickhan, R. Adjustments of global and local hindlimb properties during terrestrial locomotion of the common quail (coturnix coturnix). *Journal of Experimental Biology 216*, 20 (2013), 3906–3916.

[5] Ashley-Ross, M. Hindlimb kinematics during terrestrial locomotion in a salamander (dicamptodon tenebrosus). *Journal of experimental biology 193*, 1 (1994), 255–283.

[6] Autumn, K., Dittmore, A., Santos, D., Spenko, M., and Cutkosky, M. Frictional adhesion: a new angle on gecko attachment. *Journal of Experimental Biology 209*, 18 (2006), 3569–3579.

[7] B Cott, H. Scientific results of an inquiry into the ecology and economic status of the nile crocodile (crocodilus niloticus) in uganda and northern rhodesia. *Journal of Zoology 29*, 4 (1961), 211–356.

[8] Baisch, A. T., Heimlich, C., Karpelson, M., and Wood, R. J. HAMR3: An autonomous 1.7 g ambulatory robot. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (2011), IEEE, pp. 5073–5079.

[9] BBC one - spy in the wild. https://www.bbc.co.uk/programmes/b088t67m. Accessed: 2018-05-03.

[10] Berman, D. S., Henrici, A. C., Kissel, R. A., Sumida, S. S., and Martens, T. A new diadectid (diadectomorpha), orobates pabsti, from the early permian of central germany. *Bulletin of Carnegie Museum of Natural History* (2004), 1–36.

[11] Bicanski, A., Ryczko, D., Cabelguen, J.-M., and Ijspeert, A. J. From lamprey to salamander: an exploratory modeling study on the architecture of the spinal locomotor networks in the salamander. *Biological cybernetics 107*, 5 (2013), 565–587.

[12] Birkmeyer, P., Peterson, K., and Fearing, R. S. DASH: A dynamic 16g hexapedal robot. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on* (2009), IEEE, pp. 2683–2689.

[13] Bledt, G., Wensing, P. M., and Kim, S. Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on* (2017), IEEE, pp. 4102–4109.

[14] Blob, R. W. Evolution of hindlimb posture in nonmammalian therapsids: biomechanical tests of paleontological hypotheses. *Paleobiology 27*, 1 (2001), 14–38.

[15] Blob, R. W., and Biewener, A. A. Mechanics of limb bone loading during terrestrial locomotion in the green iguana (iguana iguana) and american alligator (alligator mississippiensis). *Journal of Experimental Biology 204*, 6 (2001), 1099–1122.

[16] Bosworth, W., Kim, S., and Hogan, N. The mit super mini cheetah: A small, low-cost quadrupedal robot for dynamic locomotion. In *Safety, Security, and Rescue Robotics (SSRR), 2015 IEEE International Symposium on* (2015), IEEE, pp. 1–8.

[17] Brainerd, E. L., Baier, D. B., Gatesy, S. M., Hedrick, T. L., Metzger, K. A., Gilbert, S. L., and Crisco, J. J. X-ray reconstruction of moving morphology (xromm): precision, accuracy and applications in comparative biomechanics research. *Journal of Experimental Zoology Part A: Ecological and Integrative Physiology 313*, 5 (2010), 262–279.

[18] Buchli, J., Kalakrishnan, M., Mistry, M., Pastor, P., and Schaal, S. Compliant quadruped locomotion over rough terrain. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on* (2009), IEEE, pp. 814–820.

[19] Buss, S. R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *http://www.math.ucsd.edu/sbuss/ResearchWeb/ikmethods/iksurvey.pdf* (2004).

[20] Byl, K., Shkolnik, A., Prentice, S., Roy, N., and Tedrake, R. Reliable dynamic motions for a stiff quadruped. In *Experimental robotics* (2009), Springer, pp. 319–328.

[21] Cantonal museum of zoology, lausanne. www.zoologie.vd.ch. Accessed: 2018-05-20.

[22] Casper, J., and Murphy, R. R. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 33*, 3 (2003), 367–385.

[23] Chang-Siu, E., Libby, T., Tomizuka, M., and Full, R. J. A lizard-inspired active tail enables rapid maneuvers and dynamic stabilization in a terrestrial robot. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (2011), IEEE, pp. 1887–1894.

[24] Chentanez, T., Huggins, S. E., and Chentanez, V. Allometric relationships of the siamese crocodile, crocdodylus siamensis. *J. Sci. Soc. Thailand 9* (1983), 5–26.

[25] Christian, A., and Garland Jr, T. Scaling of limb proportions in monitor lizards (squamata: Varanidae). *Journal of Herpetology* (1996), 219–230.

[26] Chuah, M. Y., and Kim, S. Enabling force sensing during ground locomotion: A bio-inspired, multi-axis, composite force sensor using discrete pressure mapping. *IEEE Sensors Journal 14*, 5 (2014), 1693–1703.

[27] Clark, J. E., Cham, J. G., Bailey, S. A., Froehlich, E. M., Nahata, P. K., Full, R. J., and Cutkosky, M. R. Biomimetic design and fabrication of a hexapedal running robot. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* (2001), vol. 4, IEEE, pp. 3643–3649.

[28] Cowen, R. Locomotion and respiration in aquatic air-breathing vertebrates. *Evolutionary paleobiology* (1996), 337–354.

[29] Crespi, A., Badertscher, A., Guignard, A., and Ijspeert, A. J. Amphibot i: an amphibious snake-like robot. *Robotics and Autonomous Systems 50*, 4 (2005), 163–175.

[30] Crespi, A., Badertscher, A., Guignard, A., and Ijspeert, A. J. Swimming and crawling with an amphibious snake robot. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (2005), IEEE, pp. 3024–3028.

[31] Crespi, A., and Ijspeert, A. J. Amphibot ii: An amphibious snake robot that crawls and swims using a central pattern generator. In *Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006)* (2006), no. BIOROB-CONF-2006-001, pp. 19–27.

[32] Crespi, A., and Ijspeert, A. J. Salamandra robotica: a biologically inspired amphibious robot that swims and walks. In *Artificial Life Models in Hardware*. Springer, 2009, pp. 35–64.

[33] Crespi, A., Karakasiliotis, K., Guignard, A., and Ijspeert, A. J. Salamandra robotica ii: an amphibious robot to study salamander-like swimming and walking gaits. *IEEE Transactions on Robotics 29*, 2 (2013), 308–320.

[34] Curet, O. M., Patankar, N. A., Lauder, G. V., and MacIver, M. A. Mechanical properties of a bio-inspired robotic knifefish with an undulatory propulsor. *Bioinspiration & biomimetics 6*, 2 (2011), 026004.

[35] Curth, S., Fischer, M. S., and Nyakatura, J. A. Ichnology of an extant Belly-Dragging Lizard—Analogies to early reptile locomotion? *Ichnos 21*, 1 (2014), 32–43.

[36] Curth, S., Fischer, M. S., and Nyakatura, J. A. Ichnology of an extant belly-dragging lizard—analogies to early reptile locomotion? *Ichnos 21*, 1 (2014), 32–43.

[37] Daley, M. A., and Biewener, A. A. Running over rough terrain reveals limb control for intrinsic stability. *Proceedings of the National Academy of Sciences 103*, 42 (2006), 15681–15686.

[38] de Buffrénil, V., and Hémery, G. Variation in longevity, growth, and morphology in exploited nile monitors (varanus niloticus) from sahelian africa. *Journal of Herpetology 36*, 3 (2002), 419–426.

[39] Delmerico, J., Giusti, A., Mueggler, E., Gambardella, L. M., and Scaramuzza, D. "on-the-spot training" for terrain classification in autonomous air-ground collaborative teams. In *International Symposium on Experimental Robotics* (2016), Springer, pp. 574–585.

[40] Dermitzakis, K., Carbajal, J. P., and Marden, J. H. Scaling laws in robotics. *Procedia Computer Science 7* (2011), 250–252.

[41] Di Luca, M., Mintchev, S., Heitz, G., Noca, F., and Floreano, D. Bioinspired morphing wings for extended flight envelope and roll control of small drones. *Interface focus 7*, 1 (2017), 20160092.

[42] Dodson, P. *The horned dinosaurs: A natural history*. Princeton University Press, 1998.

[43] Dörfler, F., and Bullo, F. Synchronization in complex networks of phase oscillators: A survey. *Automatica 50*, 6 (2014), 1539–1564.

[44] Dunbar, D. C., Macpherson, J. M., Simmons, R. W., and Zarcades, A. Stabilization and mobility of the head, neck and trunk in horses during overground locomotion: comparisons with humans and other primates. *Journal of Experimental Biology 211*, 24 (2008), 3889–3907.

[45] Eckert, P., Spröwitz, A., Witte, H., and Ijspeert, A. J. Comparing the effect of different spine and leg designs for a small bounding quadruped robot. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (2015), IEEE, pp. 3128–3133.

[46] Edwards, J. L. The evolution of terrestrial locomotion. In *Major patterns in vertebrate evolution*. Springer, 1977, pp. 553–577.

[47] Faessler, M., Fontana, F., Forster, C., Mueggler, E., Pizzoli, M., and Scaramuzza, D. Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle. *Journal of Field Robotics 33*, 4 (2016), 431–450.

[48] Faraji, S., Pouya, S., Atkeson, C. G., and Ijspeert, A. J. Versatile and robust 3d walking with a simulated humanoid robot (atlas): A model predictive control approach. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on* (2014), IEEE, pp. 1943–1950.

[49] Ferreau, H. J., Kirches, C., Potschka, A., Bock, H. G., and Diehl, M. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation 6*, 4 (2014), 327–363.

[50] Ferworn, A., Wright, C., Tran, J., Li, C., and Choset, H. Dog and snake marsupial cooperation for urban search and rescue deployment. In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on* (2012), IEEE, pp. 1–5.

[51] Fischer, M. S., Krause, C., and Lilje, K. E. Evolution of chameleon locomotion, or how to become arboreal as a reptile. *Zoology 113*, 2 (2010), 67–74.

[52] Fish, F. E., Bostic, S. A., Nicastro, A. J., and Beneski, J. T. Death roll of the alligator: mechanics of twist feeding in water. *Journal of experimental biology 210*, 16 (2007), 2811–2818.

[53] Focchi, M., Barasuol, V., Havoutis, I., Buchli, J., Semini, C., and Caldwell, D. G. Local reflex generation for obstacle negotiation in quadrupedal locomotion. In *Int. Conf. on climbing and walking robots (CLAWAR)* (2013).

[54] Focchi, M., Del Prete, A., Havoutis, I., Featherstone, R., Caldwell, D. G., and Semini, C. High-slope terrain locomotion for torque-controlled quadruped robots. *Autonomous Robots 41*, 1 (2017), 259–272.

[55] Folkertsma, G. A., Kim, S., and Stramigioli, S. Parallel stiffness in a bounding quadruped with flexible spine. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012), IEEE, pp. 2210–2215.

[56] Forssberg, H. Stumbling corrective reaction: a phase-dependent compensatory reaction during locomotion. *J Neurophysiol 42*, 4 (1979), 936–953.

[57] Full, R. J., and Koditschek, D. E. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *Journal of experimental biology 202*, 23 (1999), 3325–3332.

[58] Fuller, P. O., Higham, T. E., and Clark, A. J. Posture, speed, and habitat structure: three-dimensional hindlimb kinematics of two species of padless geckos. *Zoology 114*, 2 (2011), 104–112.

[59] Galloway, K. C., Haynes, G. C., Ilhan, B. D., Johnson, A. M., Knopf, R., Lynch, G. A., Plotnick, B. N., White, M., and Koditschek, D. E. X-RHex: A highly mobile hexapedal robot for sensorimotor tasks.

[60] Garcia, C. E., Prett, D. M., and Morari, M. Model predictive control: theory and practice—a survey. *Automatica 25*, 3 (1989), 335–348.

[61] Gatesy, S. M., Baier, D. B., Jenkins, F. A., and Dial, K. P. Scientific rotoscoping: a morphology-based method of 3-d motion analysis and visualization. *Journal of Experimental Zoology Part A: Ecological and Integrative Physiology 313*, 5 (2010), 244–261.

[62] Gatesy, S. M., Bäker, M., and Hutchinson, J. R. Constraint-based exclusion of limb poses for reconstructing theropod dinosaur locomotion. *Journal of Vertebrate Paleontology 29*, 2 (2009), 535–544.

[63] Gatesy, S. M., and Pollard, N. S. Apples, oranges, and angles: comparative kinematic analysis of disparate limbs. *Journal of theoretical biology 282*, 1 (2011), 7–13.

[64] Gingerich, P. D. Land-to-sea transition in early whales: evolution of eocene archaeoceti (cetacea) in relation to skeletal proportions and locomotion of living semiaquatic mammals. *Paleobiology 29*, 3 (2003), 429–454.

[65] Gong, C., Travers, M., Astley, H. C., Goldman, D. I., and Choset, H. Limbless locomotors that turn in place. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (May 2015), pp. 3747–3754.

[66] Gravish, N., and Lauder, G. V. Robotics-inspired biology. *Journal of Experimental Biology 221*, 7 (2018), jeb138438.

[67] Grieco, J. C., Prieto, M., Armada, M., and De Santos, P. G. A six-legged climbing robot for high payloads. In *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on* (1998), vol. 1, IEEE, pp. 446–450.

[68] Gromov, B., Gambardella, L. M., and Di Caro, G. A. Wearable multi-modal interface for human multi-robot interaction. In *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on* (2016), IEEE, pp. 240–245.

[69] Ha, P., and Byl, K. Feasibility and optimization of fast quadruped walking with one-versus two-at-a-time swing leg motions for robosimian.

[70] Haldane, D. W., Peterson, K. C., Bermudez, F. L. G., and Fearing, R. S. Animal-inspired design and aerodynamic stabilization of a hexapedal millirobot. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (2013), IEEE, pp. 3279–3286.

[71] Hardarson, F. Stability analysis and synthesis of statically balanced walking for quadruped robots. *Royal Institute of Technology* (2002).

[72] Hatton, R. L., and Choset, H. Generating gaits for snake robots: annealed chain fitting and keyframe wave extraction. *Autonomous Robots 28*, 3 (2010), 271–281.

[73] Hawkes, E. W., Ulmen, J., Esparza, N., and Cutkosky, M. R. Scaling walls: Applying dry adhesives to the real world. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (2011), IEEE, pp. 5100–5106.

[74] Haynes, G. C., Khripin, A., Lynch, G., Amory, J., Saunders, A., Rizzi, A. A., and Koditschek, D. E. Rapid pole climbing with a quadrupedal robot. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on* (2009), IEEE, pp. 2767–2772.

[75] Hebert, P., Bajracharya, M., Ma, J., Hudson, N., Aydemir, A., Reid, J., Bergh, C., Borders, J., Frost, M., Hagman, M., et al. Mobile manipulation and mobility as manipulation—design and algorithms of robosimian. *Journal of Field Robotics 32*, 2 (2015), 255–274.

[76] Herdt, A., Diedam, H., Wieber, P.-B., Dimitrov, D., Mombaur, K., and Diehl, M. Online walking motion generation with automatic footstep placement. *Advanced Robotics 24*, 5-6 (2010), 719–737.

[77] Hibot acm-r5h snake robot. http://www.hibot.co.jp/ecommerce/prod-detail/14. Accessed: 2018-04-20.

[78] Hildebrand, M. The quadrupedal gaits of vertebrates. *BioScience 39*, 11 (1989), 766.

[79] Hirose, S. Biologically inspired robots: Snake-like locomotors and manipulators, 1993.

[80] Hirose, S., and Mori, M. Biologically inspired snake-like robots. In *Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on* (2004), IEEE, pp. 1–7.

[81] Hirose, S., and Yamada, H. Snake-like robots [tutorial]. *IEEE Robotics & Automation Magazine 16*, 1 (2009), 88–98.

[82] Hirose, S., and Yamada, H. Snake-like robots [tutorial]. *Robotics Automation Magazine, IEEE 16*, 1 (March 2009), 88–98.

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

[84] Horvat, T., Karakasiliotis, K., Melo, K., Fleury, L., Thandiackal, R., and Ijspeert, A. J. Inverse kinematics and reflex based controller for body-limb coordination of a salamander-like robot walking on uneven terrain. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on* (2015), IEEE, pp. 195–201.

[85] Horvat, T., Melo, K., and Ijspeert, A. J. Model predictive control based framework for CoM control of a quadruped robot. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on* (2017), IEEE, pp. 3372–3378.

[86] Horvat, T., Melo, K., and Ijspeert, A. J. Spine controller for a sprawling posture robot. *IEEE Robotics and Automation Letters 2*, 2 (2017), 1195–1202.

[87] Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C. D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M., et al. Anymal-a highly mobile and dynamic quadrupedal robot. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on* (2016), IEEE, pp. 38–44.

[88] Hutter, M., Gehring, C., Lauber, A., Gunther, F., Bellicoso, C. D., Tsounis, V., Fankhauser, P., Diethelm, R., Bachmann, S., Blösch, M., et al. Anymal-toward legged robots for harsh environments. *Advanced Robotics 31*, 17 (2017), 918–931.

[89] Ijspeert, A. J. Biorobotics: Using robots to emulate and investigate agile locomotion. *science 346*, 6206 (2014), 196–203.

[90] Ijspeert, A. J., and Crespi, A. Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model. In *Proceedings 2007 IEEE International Conference on Robotics and Automation* (April 2007), pp. 262–268.

[91] Ijspeert, A. J., Crespi, A., Ryczko, D., and Cabelguen, J.-M. From swimming to walking with a salamander robot driven by a spinal cord model. *Science 315*, 5817 (2007), 1416–1420.

[92] Jagnandan, K., and Higham, T. E. Lateral movements of a massive tail influence gecko locomotion: an integrative study comparing tail restriction and autotomy. *Sci. Rep. 7*, 1 (Sept. 2017), 10865.

[93] Jagnandan, K., Russell, A. P., and Higham, T. E. Tail autotomy and subsequent regeneration alter the mechanics of locomotion in lizards. *J. Exp. Biol. 217*, Pt 21 (Nov. 2014), 3891–3897.

[94] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on* (2003), vol. 2, IEEE, pp. 1620–1626.

[95] Kalouche, S., Rollinson, D., and Choset, H. Modularity for maximum mobility and manipulation: Control of a reconfigurable legged robot with series-elastic actuators. In *Safety, Security, and Rescue Robotics (SSRR), 2015 IEEE International Symposium on* (2015), IEEE, pp. 1–8.

[96] Karakasiliotis, K. Legged locomotion with spinal undulations.

[97] Karakasiliotis, K., Schilling, N., Cabelguen, J.-M., and Ijspeert, A. J. Where are we in understanding salamander locomotion: biological and robotic perspectives on kinematics. *Biological cybernetics 107*, 5 (2013), 529–544.

[98] Karakasiliotis, K., Thandiackal, R., Melo, K., Horvat, T., Mahabadi, N., Tsitkov, S., Cabelguen, J., and Ijspeert, A. From cineradiography to biorobots: an approach for designing robots to emulate and study animal locomotion. *Journal of The Royal Society Interface 13*, 119 (2016), 20151089.

[99] Karumanchi, S., Edelberg, K., Baldwin, I., Nash, J., Reid, J., Bergh, C., Leichty, J., Carpenter, K., Shekels, M., Gildner, M., et al. Team robosimian: semi-autonomous mobile manipulation at the 2015 darpa robotics challenge finals. *Journal of Field Robotics 34*, 2 (2017), 305–332.

[100] Katzschmann, R. K., DelPreto, J., MacCurdy, R., and Rus, D. Exploration of underwater life with an acoustically controlled soft robotic fish.

[101] Kawano, S. M., and Blob, R. W. Propulsive forces of mudskipper fins and salamander limbs during terrestrial locomotion: implications for the invasion of land. *Integrative and comparative biology 53*, 2 (2013), 283–294.

[102] Kelasidi, E., Liljeback, P., Pettersen, K. Y., and Gravdahl, J. T. Innovation in underwater robots: biologically inspired swimming snake robots. *IEEE Robotics & Automation Magazine 23*, 1 (2016), 44–62.

[103] Khoramshahi, M., Spröwitz, A., Tuleu, A., Ahmadabadi, M. N., and Ijspeert, A. J. Benefits of an active spine supported bounding locomotion with a small compliant quadruped robot. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (2013), IEEE, pp. 3329–3334.

[104] Kim, S., Clark, J. E., and Cutkosky, M. R. iSprawl: Design and tuning for high-speed autonomous open-loop running. *The International Journal of Robotics Research 25*, 9 (2006), 903–912.

[105] Kim, S., Spenko, M., Trujillo, S., Heyneman, B., Santos, D., and Cutkosky, M. R. Smooth vertical surface climbing with directional adhesion. *IEEE Transactions on robotics 24*, 1 (2008), 65–74.

[106] King, D. Dlib c++ library. *Access on: http://dlib.net* (2015).

[107] Kitano, S., Hirose, S., Endo, G., and Fukushima, E. F. Development of lightweight sprawling-type quadruped robot titan-xiii and its dynamic walking. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on* (2013), IEEE, pp. 6025–6030.

[108] Kitano, S., Hirose, S., Horigome, A., and Endo, G. Titan-xiii: sprawling-type quadruped robot with ability of fast and energy-efficient walking. *ROBOMECH Journal 3*, 1 (2016), 8.

[109] Lauder, G. V., Anderson, E. J., Tangorra, J., and Madden, P. G. Fish biorobotics: kinematics and hydrodynamics of self-propulsion. *Journal of experimental biology 210*, 16 (2007), 2767–2780.

[110] Lauder, G. V., and Thomason, J. On the inference of function from structure. *Functional morphology in vertebrate paleontology* (1995), 1–18.

[111] Laurin, M., and Reisz, R. R. A reevaluation of early amniote phylogeny. *Zoological Journal of the Linnean Society 113*, 2 (1995), 165–223.

[112] Li, C., Zhang, T., and Goldman, D. I. A terradynamics of legged locomotion on granular media. *science 339*, 6126 (2013), 1408–1412.

[113] Libby, T., Moore, T. Y., Chang-Siu, E., Li, D., Cohen, D. J., Jusufi, A., and Full, R. J. Tail-assisted pitch control in lizards, robots and dinosaurs. *Nature 481*, 7380 (2012), 181.

[114] Liljeback, P., Pettersen, K. Y., Stavdahl, Ø., and Gravdahl, J. T. Experimental investigation of obstacle-aided locomotion with a snake robot. *IEEE Transactions on Robotics 27*, 4 (2011), 792–800.

[115] Liljeback, P., Pettersen, K. Y., Stavdahl, Ø., and Gravdahl, J. T. Snake robot locomotion in environments with obstacles. *IEEE/ASME Transactions on Mechatronics 17*, 6 (2012), 1158–1169.

[116] Losos, J. B. Convergence, adaptation, and constraint. *Evolution 65*, 7 (2011), 1827–1840.

[117] Luiselli, L., Akani, G., and Capizzi, D. Is there any interspecific competition between dwarf crocodiles (osteolaemus tetraspis) and nile monitors (varanus niloticus ornatus) in the swamps of central africa? a study from south-eastern nigeria. *Journal of Zoology 247*, 1 (1999), 127–131.

[118] Lydekker, R. *The royal natural history*, vol. 1. F. Warne, 1894.

[119] Ma, K. Y., Chirarattananon, P., Fuller, S. B., and Wood, R. J. Controlled flight of a biologically inspired, insect-scale robot. *Science 340*, 6132 (2013), 603–607.

[120] Marden, J. H. Scaling of maximum net force output by motors used for locomotion. *Journal of Experimental Biology 208*, 9 (2005), 1653–1664.

[121] Marvi, H., Gong, C., Gravish, N., Astley, H., Travers, M., Hatton, R. L., Mendelson, J. R., Choset, H., Hu, D. L., and Goldman, D. I. Sidewinding with minimal slip: Snake and robot ascent of sandy slopes. *Science 346*, 6206 (2014), 224–229.

[122] McInroe, B., Astley, H. C., Gong, C., Kawano, S. M., Schiebel, P. E., Rieser, J. M., Choset, H., Blob, R. W., and Goldman, D. I. Tail use improves performance on soft substrates in models of early vertebrate land locomotors. *Science 353*, 6295 (2016), 154–158.

[123] Melo, K. Modular snake robot velocity for side-winding gaits. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (2015), IEEE, pp. 3716–3722.

[124] Melo, K., Horvat, T., and Ijspeert, A. J. Testing bio-robots in african wilderness prepares them for disaster response missions. *Manuscript in preparation.* (2018).

[125] Melo, K., and Paez, L. Modular snake robot gaits on horizontal pipes. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on* (Oct 2012), pp. 3099–3104.

[126] Melo, K., Paez, L., and Parra, C. Indoor and outdoor parametrized gait execution with modular snake robots. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (2012), IEEE, pp. 3525–3526.

[127] Murphy, R. Disaster robotics. intelligent robotics and autonomous agents series. *The MIT Press* (2014).

[128] Murphy, R. R. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 34*, 2 (2004), 138–153.

[129] Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., and Erkmen, A. M. Search and rescue robotics. In *Springer Handbook of Robotics*. Springer, 2008, pp. 1151–1173.

[130] Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., et al. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics 30*, 1 (2013), 44–63.

[131] Nagatani, K., Kiribayashi, S., Okada, Y., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., and Hada, Y. Redesign of rescue mobile robot quince. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on* (2011), IEEE, pp. 13–18.

[132] Nassiraei, A. A., Kawamura, Y., Ahrary, A., Mikuriya, Y., and Ishii, K. Concept and design of a fully autonomous sewer pipe inspection mobile robot" kantaro". In *Robotics and Automation, 2007 IEEE International Conference on* (2007), IEEE, pp. 136–143.

[133] NCCR robotics. https://www.nccr-robotics.ch/. Accessed: 2018-05-05.

[134] Neubauer, W. A spider-like robot that climbs vertically in ducts or pipes. In *Intelligent Robots and Systems' 94.'Advanced Robotic Systems and the Real World', IROS'94. Proceedings of the IEEE/RSJ/GI International Conference on* (1994), vol. 2, IEEE, pp. 1178–1185.

[135] Neunert, M., Stäuble, M., Giftthaler, M., Bellicoso, C. D., Carius, J., Gehring, C., Hutter, M., and Buchli, J. Whole-body nonlinear model predictive control through contacts for quadrupeds. *arXiv preprint arXiv:1712.02889* (2017).

[136] Ngwenya, A., Patzke, N., Spocter, M. A., Kruger, J.-L., Dell, L.-A., Chawana, R., Mazengenya, P., Billings, B. K., Olaleye, O., Herculano-Houzel, S., et al. The continuously growing central nervous system of the nile crocodile (crocodylus niloticus). *The Anatomical Record 296*, 10 (2013), 1489–1500.

[137] Nicolas, G., Multon, F., Berillon, G., and Marchal, F. From bone to plausible bipedal locomotion using inverse kinematics. *Journal of biomechanics 40*, 5 (2007), 1048–1057.

[138] Nie, C., Pacheco Corcho, X., and Spenko, M. Robots on the move: Versatility and complexity in mobile robot locomotion. *Robotics & Automation Magazine, IEEE 20*, 4 (2013), 72–82.

[139] Nyakatura, J. A., Allen, V. R., Lauströer, J., Andikfar, A., Danczak, M., Ullrich, H.-J., Hufenbach, W., Martens, T., and Fischer, M. S. A three-dimensional skeletal reconstruction of the stem amniote orobates pabsti (diadectidae): analyses of body mass, centre of mass position, and joint mobility. *PloS one 10*, 9 (2015), e0137284.

[140] Nyakatura, J. A., Andrada, E., Curth, S., and Fischer, M. S. Bridging "romer's gap": limb mechanics of an extant belly-dragging lizard inform debate on tetrapod locomotion during the early carboniferous. *Evolutionary Biology 41*, 2 (2014), 175–190.

[141] Nyakatura, J. A., and Fischer, M. S. Functional morphology and three-dimensional kinematics of the thoraco-lumbar region of the spine of the two-toed sloth. *Journal of Experimental Biology 213*, 24 (2010), 4278–4290.

[142] Nyakatura, J. A., and Fischer, M. S. Three-dimensional kinematic analysis of the pectoral girdle during upside-down locomotion of two-toed sloths (choloepus didactylus, linné 1758). *Frontiers in zoology 7*, 1 (2010), 21.

[143] Nyakatura, J. A., Melo, K., Horvat, T., Karakasiliotis, K., Allen, V. R., Andikfar, A., Andrada, E., Arnold, P., Lauströer, J., Hutchinson, J. R., Fisher, M. S., and Ijspeert, A. J. Reverse engineering the locomotion of an extinct stem amniote. *Manuscript submitted to Nature.* (2018).

[144] Owaki, D., Kano, T., Nagasawa, K., Tero, A., and Ishiguro, A. Simple robot suggests physical interlimb communication is essential for quadruped walking. *Journal of The Royal Society Interface 10*, 78 (2013), 20120669.

[145] Paden, B., Čáp, M., Yong, S. Z., Yershov, D., and Frazzoli, E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles 1*, 1 (2016), 33–55.

[146] Park, H.-W., Park, S., and Kim, S. Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3d running of mit cheetah 2. In *Robotics and automation (ICRA), 2015 IEEE international conference on* (2015), IEEE, pp. 5163–5170.

[147] Park, Y.-L., Chen, B.-R., and Wood, R. J. Design and fabrication of soft artificial skin using embedded microchannels and liquid conductors. *IEEE Sensors Journal 12*, 8 (2012), 2711–2718.

[148] Pennycuick, C., and Rezende, M. A. The specific power output of aerobic muscle, related to the power density of mitochondria. *Journal of Experimental Biology 108*, 1 (1984), 377–392.

[149] Pierce, S. E., Clack, J. A., and Hutchinson, J. R. Three-dimensional limb joint mobility in the early tetrapod ichthyostega. *Nature 486*, 7404 (2012), 523.

[150] Pongas, D., Mistry, M., and Schaal, S. A robust quadruped walking gait for traversing rough terrain. In *Robotics and Automation, 2007 IEEE International Conference on* (2007), IEEE, pp. 1474–1479.

[151] Porez, M., Boyer, F., and Ijspeert, A. J. Improved lighthill fish swimming model for bio-inspired robots: Modeling, computational aspects and experimental comparisons. *The International Journal of Robotics Research 33*, 10 (2014), 1322–1341.

[152] Potemkin, E., Astafurov, P., Osipov, A., Malenkov, M., Mishkinyuk, V., and Sologub, P. Remote-controlled robots for repair and recovery in the zones of high radiation levels. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on* (1992), IEEE, pp. 80–82.

[153] Prahacs, C., Saudners, A., Smith, M. K., McMordie, D., and Buehler, M. Towards legged amphibious mobile robotics. *Proceedings of the Canadian Engineering Education Association* (2011).

[154] Pratt, J., Chew, C.-M., Torres, A., Dilworth, P., and Pratt, G. Virtual model control: An intuitive approach for bipedal locomotion. *The International Journal of Robotics Research 20*, 2 (2001), 129–143.

[155] Preuschoft, H., and Klein, N. Torsion and bending in the neck and tail of sauropod dinosaurs and the function of cervical ribs: insights from functional morphology and biomechanics. *PloS one 8*, 10 (2013), e78574.

[156] Raibert, M., Blankespoor, K., Nelson, G., and Playter, R. Bigdog, the rough-terrain quadruped robot. *IFAC Proceedings Volumes 41*, 2 (2008), 10822–10825.

[157] Ramdya, P., Thandiackal, R., Cherney, R., Asselborn, T., Benton, R., Ijspeert, A. J., and Floreano, D. Climbing favours the tripod gait over alternative faster insect gaits. *Nature communications 8* (2017), 14494.

[158] REILLY, S. M., AND ELIAS, J. A. Locomotion in alligator mississippiensis: kinematic effects of speed and posture and their relevance to the sprawling-to-erect paradigm. *Journal of Experimental Biology 201*, 18 (1998), 2559–2574.

[159] REILLY, S. M., AND ELIAS, J. A. Locomotion in alligator mississippiensis: kinematic effects of speed and posture and their relevance to the sprawling-to-erect paradigm. *J. Exp. Biol. 201 (Pt 18)* (Sept. 1998), 2559–2574.

[160] REILLY, S. M., MCELROY, E. J., ODUM, R. A., AND HORNYAK, V. A. Tuataras and salamanders show that walking and running mechanics are ancient features of tetrapod locomotion. *Proceedings of the Royal Society of London B: Biological Sciences 273*, 1593 (2006), 1563–1568.

[161] REILLY, S. M., WILLEY, J. S., BIKNEVICIUS, A. R., AND BLOB, R. W. Hindlimb function in the alligator: integrating movements, motor patterns, ground reaction forces and bone strain of terrestrial locomotion. *Journal of Experimental Biology 208*, 6 (2005), 993–1009.

[162] REISZ, R. R. The origin and early evolutionary history of amniotes. *Trends in ecology & evolution 12*, 6 (1997), 218–222.

[163] RIMON, E., AND BURDICK, J. W. A configuration space analysis of bodies in contact—i. 1st order mobility. *Mechanism and machine theory 30*, 6 (1995), 897–912.

[164] RISKIN, D. K., KENDALL, C. J., AND HERMANSON, J. W. The crouching of the shrew: Mechanical consequences of limb posture in small mammals. *PeerJ 4* (2016), e2131.

[165] ROGNON, C., MINTCHEV, S., DELL'AGNOLA, F., CHERPILLOD, A., ATIENZA, D., AND FLOREANO, D. Flyjacket: an upper-body soft exoskeleton for immersive drone control. *IEEE Robotics and Automation Letters* (2018).

[166] ROLLINSON, D., BUCHAN, A., AND CHOSET, H. Virtual chassis for snake robots: definition and applications. *Advanced Robotics 26*, 17 (2012), 2043–2064.

[167] ROLLINSON, D., AND CHOSET, H. Pipe network locomotion with a snake robot. *Journal of Field Robotics 33*, 3 (2016), 322–336.

[168] ROME, L. C. Energetic cost of running with different muscle temperatures in savannah monitor lizards. *Journal of Experimental Biology 99*, 1 (1982), 269–277.

[169] ROME, L. C. Energetic cost of running with different muscle temperatures in savannah MOnitor lizards. *J. Exp. Biol. 99*, 1 (Aug. 1982), 269–277.

[170] ROSATI, S., KRUŻELECKI, K., HEITZ, G., FLOREANO, D., AND RIMOLDI, B. Dynamic routing for flying ad hoc networks. *IEEE Transactions on Vehicular Technology 65*, 3 (2016), 1690–1700.

[171] Roslin, N. S., Anuar, A., Jalal, M. F. A., and Sahari, K. S. M. A review: hybrid locomotion of in-pipe inspection robot. *Procedia Engineering 41* (2012), 1456–1462.

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

[173] Scaramuzza, D., Achtelik, M. C., Doitsidis, L., Friedrich, F., Kosmatopoulos, E., Martinelli, A., Achtelik, M. W., Chli, M., Chatzichristofis, S., Kneip, L., et al. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments. *IEEE Robotics & Automation Magazine 21*, 3 (2014), 26–40.

[174] Schachner, E. R., Cieri, R. L., Butler, J. P., and Farmer, C. Unidirectional pulmonary airflow patterns in the savannah monitor lizard. *Nature 506*, 7488 (2014), 367.

[175] Sellers, W., Manning, P., Lyson, T., Stevens, K., and Margetts, L. Virtual palaeontology: gait reconstruction of extinct vertebrates using high performance computing. *Palaeontologia Electronica 12*, 3 (2009), 11A.

[176] Semini, C., Tsagarakis, N. G., Guglielmino, E., Focchi, M., Cannella, F., and Caldwell, D. G. Design of HyQ–a hydraulically and electrically actuated quadruped robot. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 225*, 6 (2011), 831–849.

[177] Seok, S., Wang, A., Chuah, M. Y., Otten, D., Lang, J., and Kim, S. Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on* (2013), IEEE, pp. 3307–3312.

[178] Shoval, S., Rimon, E., and Shapiro, A. Design of a spider-like robot for motion with quasi-static force constraints. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on* (1999), vol. 2, IEEE, pp. 1377–1383.

[179] Spenko, M., Haynes, G. C., Saunders, J., Cutkosky, M. R., Rizzi, A. A., Full, R. J., and Koditschek, D. E. Biologically inspired climbing with a hexapedal robot. *Journal of Field Robotics 25*, 4-5 (2008), 223–242.

[180] Stevens, K. A., Ernst, S., and Marty, D. Uncertainty and ambiguity in the interpretation of sauropod trackways. *Dinosaur tracks—The next steps* (2016), 227–243.

[181] Sullivan, C. S. *Function and evolution of the hind limb in Triassic archosaurian reptiles*. Harvard University, 2007.

[182] Sumida, S. S., and Modesto, S. A phylogenetic perspective on locomotory strategies in early amniotes. *American Zoologist 41*, 3 (2001), 586–597.

[183] Tar, A., and Cserey, G. Development of a low cost 3d optical compliant tactile force sensor. In *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on* (2011), IEEE, pp. 236–240.

[184] Tesch, M., Lipkin, K., Brown, I., Hatton, R., Peck, A., Rembisz, J., and Choset, H. Parameterized and scripted gaits for modular snake robots. *Advanced Robotics 23*, 9 (2009), 1131–1158.

[185] Thandiackal, R. *Investigating Sensorimotor Control in Locomotion using Robots and Mathematical Models*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, 2017.

[186] Tuleu, A. Hardware, software and control design considerations towards low-cost compliant quadruped robots.

[187] Vespignani, M., Melo, K., Mutlu, M., and Ijspeert, A. J. Compliant snake robot locomotion on horizontal pipes. In *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (2015), IEEE, pp. 1–8.

[188] Voigt, S., Berman, D. S., and Henrici, A. C. First well-established track-trackmaker association of paleozoic tetrapods based on ichniotherium trackways and diadectid skeletons from the lower permian of germany. *Journal of Vertebrate Paleontology 27*, 3 (2007), 553–570.

[189] Vukobratović, M., and Borovac, B. Zero-moment point—thirty five years of its life. *International Journal of Humanoid Robotics 1*, 01 (2004), 157–173.

[190] Wagner, L., Fankhauser, P., Bloesch, M., and Hutter, M. Foot contact estimation for legged robots in rough terrain. In *Advances in Cooperative Robotics*. World Scientific, 2017, pp. 395–403.

[191] Wang, Y., and Boyd, S. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology 18*, 2 (2010), 267–278.

[192] Watt, A., and Watt, M. Forward vs inverse kinematics in computer animation. *Advanced animation and rendering techniques* (1992), 371–384.

[193] Weinmeister, K., Eckert, P., Witte, H., and Ijspeert, A.-J. Cheetah-cub-s: Steering of a quadruped robot using trunk motion. In *Safety, Security, and Rescue Robotics (SSRR), 2015 IEEE International Symposium on* (2015), IEEE, pp. 1–6.

[194] Wensing, P. M., Wang, A., Seok, S., Otten, D., Lang, J., and Kim, S. Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots. *IEEE Transactions on Robotics 33*, 3 (2017), 509–522.

[195] Whitney, D. E. Force feedback control of manipulator fine motions. *Journal of Dynamic Systems, Measurement, and Control 99*, 2 (1977), 91–97.

[196] WITMER, L. M., ET AL. The extant phylogenetic bracket and the importance of reconstructing soft tissues in fossils. *Functional morphology in vertebrate paleontology 1* (1995), 19–33.

[197] WITTE, H., BILTZINGER, J., HACKERT, R., SCHILLING, N., SCHMIDT, M., REICH, C., AND FISCHER, M. S. Torque patterns of the limbs of small therian mammals during locomotion on flat ground. *Journal of Experimental Biology 205*, 9 (2002), 1339–1353.

[198] WRIGHT, C., JOHNSON, A., PECK, A., MCCORD, Z., NAAKTGEBOREN, A., GIANFORTONI, P., GONZALEZ-RIVERO, M., HATTON, R., AND CHOSET, H. Design of a modular snake robot. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on* (2007), IEEE, pp. 2609–2614.

[199] YAMADA, H., AND HIROSE, S. Study on the 3d shape of active cord mechanism. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* (2006), IEEE, pp. 2890–2895.

[200] YAMADA, H., TAKAOKA, S., AND HIROSE, S. A snake-like robot for real-world inspection applications (the design and control of a practical active cord mechanism). *Advanced Robotics 27*, 1 (2013), 47–60.

[201] ZAGLER, A., AND PFEIFFER, F. "MORITZ" a pipe crawler for tube junctions. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on* (2003), vol. 3, IEEE, pp. 2954–2959.

[202] ZENG, G., AND HEMAMI, A. An overview of robot force control. *Robotica 15*, 5 (1997), 473–482.

[203] ZHANG, Y., AND YAN, G. In-pipe inspection robot with active pipe-diameter adaptability and automatic tractive force adjusting. *Mechanism and Machine Theory 42*, 12 (2007), 1618–1631.

[204] ZHAO, Q., NAKAJIMA, K., SUMIOKA, H., HAUSER, H., AND PFEIFER, R. Spine dynamics as a computational resource in spine-driven quadruped locomotion. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2013), IEEE, pp. 1445–1451.

[205] ZHAO, Q., SUMIOKA, H., NAKAJIMA, K., YU, X., AND PFEIFER, R. Spine as an engine: effect of spine morphology on spine-driven quadruped locomotion. *Advanced Robotics 28*, 6 (2014), 367–378.

TOMISLAV HORVAT

Ch. du Stand 14, Apt. 21
1024 Ecublens, Vaud, Switzerland
Phone: +41 79 136 37 75
Skype: tomislav.horvat27
Email: tomislav.horvat@epfl.ch
thorvat27@gmail.com

January 1990
Single
Croatian
B residence permit
B driving licence

PROFESSIONAL EXPERIENCE

Research assistant **Sep. 2013 – Aug. 2018 (est)**
EPFL, Biorobotics laboratory Lausanne, Switzerland

- Developing and maintaining software and control framework for multiple robots
- Developing and using various control strategies and algorithms for advanced robot locomotion
- Improving robots' fitness to field applications by engineering new human-machine interfaces and adding communication capabilities
- Using a robot, which mimics kinematics of a fossil, to investigating locomotion capabilities of an extinct stem amniote

Teaching assistant **Jun. 2015 – Jan. 2018**
EPFL, Biorobotics laboratory Lausanne, Switzerland

- Main teaching assistant for practical sessions of the Computational Motor Control course - designing and preparing programming and simulation exercises
- Supervising multiple semester projects and internships

Control engineer - intern **Jul. – Aug. 2012**
CERN, Beams Department, Radio-Frequency Group Meyrin, Switzerland

- Created a LHC Radiofrequency cavity model in Matlab / Simulink
- Simulated the model with included VHDL firmware of a new Adaptive Setpoint Algorithm
- Debugged, validated the firmware and tuned critical parameters

EDUCATION

PhD in Robotics **Sep. 2013 – Aug. 2018 (est)**
EPFL, Biorobotics laboratory Lausanne, Switzerland

- Doctoral Thesis: "Control of Bio-Inspired Sprawling Posture Quadruped Robots with an Actuated Spine"

MSc in Electrical Engineering and Information Technology **Sep. 2011 – Jul. 2013**
University of Zagreb, Faculty of Electrical Engineering and Computing Zagreb, Croatia

- Focus: Control, Robotics
- MSc Thesis: "Continuous-time model predictive control"

BSc in Electrical Engineering and Information Technology **Sep. 2008 – Jul. 2011**
University of Zagreb, Faculty of Electrical Engineering and Computing Zagreb, Croatia

- BSc Thesis: "Quasi-stationary optimal control of a wind farm"

TECHNICAL SKILLS

- **Programming**: C, C++, C#, Matlab, Mathematica, Python, VHDL, Javascript
- **Tools**: Simulink, LabView, Git, Inventor, Step 7, Webots robot simulator, Unity
- **Multimedia**: Premiere Pro, Lightroom, Photoshop, Inkscape
- **OS**: Linux, Windows
- **Robots**: Pleurobot, Krock-1, Krock-2, OroBOT, AgnathaX, Serval

LANGUAGES

English	fluent
German	basic
Croatian	native

ACHIEVEMENTS

Handling robots on the Nile's shores as a part of a BBC filming crew for the documentary series "Spy in the Wild"	**2016 (Uganda)**
Controlling a robot at TED Global event during Prof. Ijspeert's talk	**2015 (Geneva)**
Presenter at Bay Area Science Festival to promote Swiss robotics	**2015 (San Francisco)**
Received EPFL Doctoral School fellowship	**2013 (Lausanne)**
Graduated from University of Zagreb with Summa Cum Laude	**2013 (Zagreb)**
Annual Dean's recognitions (Josip Loncar Award) for being in top 1% in the generation	**2009/10/11/12 (Zagreb)**

HOBBIES AND INTERESTS

Technology and gadgets
Photography
Traveling
DIY projects
Board and video games

SELECTED PUBLICATIONS

Journal Articles

- **Horvat, T.**, Melo, K., Ijspeert, A. J., (2017b). “Spine Controller for a Sprawling Posture Robot”. In: *IEEE Robotics and Automation Letters* 2.2, pp. 1195–1202.
- Karakasiliotis, K, Thandiackal, R, Melo, K, **Horvat, T**, Mahabadi, N., Tsitkov, S, Cabelguen, J., Ijspeert, A., (2016). “From cineradiography to biorobots: an approach for designing robots to emulate and study animal locomotion”. In: *Journal of The Royal Society Interface* 13.119, p. 20151089.

Conference Proceedings

- **Horvat, T.**, Melo, K., Ijspeert, A. J., (2017a). “Model predictive control based framework for CoM control of a quadruped robot”. In: *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on.* IEEE, pp. 3372–3378.
- Parsapour, M., Melo, K., **Horvat, T.**, Ijspeert, A. J., (2017). “Challenges in visual and inertial information gathering for a sprawling posture robot”. In: *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on.* IEEE, pp. 2691–2697.
- Vasconcelos, R., Hauser, S., Dzeladini, F., Mutlu, M., **Horvat, T.**, Melo, K., Oliveira, P., Ijspeert, A., (2017). “Active stabilization of a stiff quadruped robot using local feedback”. In: *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on.* IEEE, pp. 4903–4910.
- **Horvat, T.**, Karakasiliotis, K., Melo, K., Fleury, L., Thandiackal, R., Ijspeert, A. J., (2015). “Inverse kinematics and reflex based controller for body-limb coordination of a salamander-like robot walking on uneven terrain”. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on.* IEEE, pp. 195–201.
- **Horvat, T.**, Spudić, V., Baotić, M., (2012). “Quasi-stationary optimal control for wind farm with closely spaced turbines”. In: *MIPRO, 2012 Proceedings of the 35th International Convention.* IEEE, pp. 829–834.